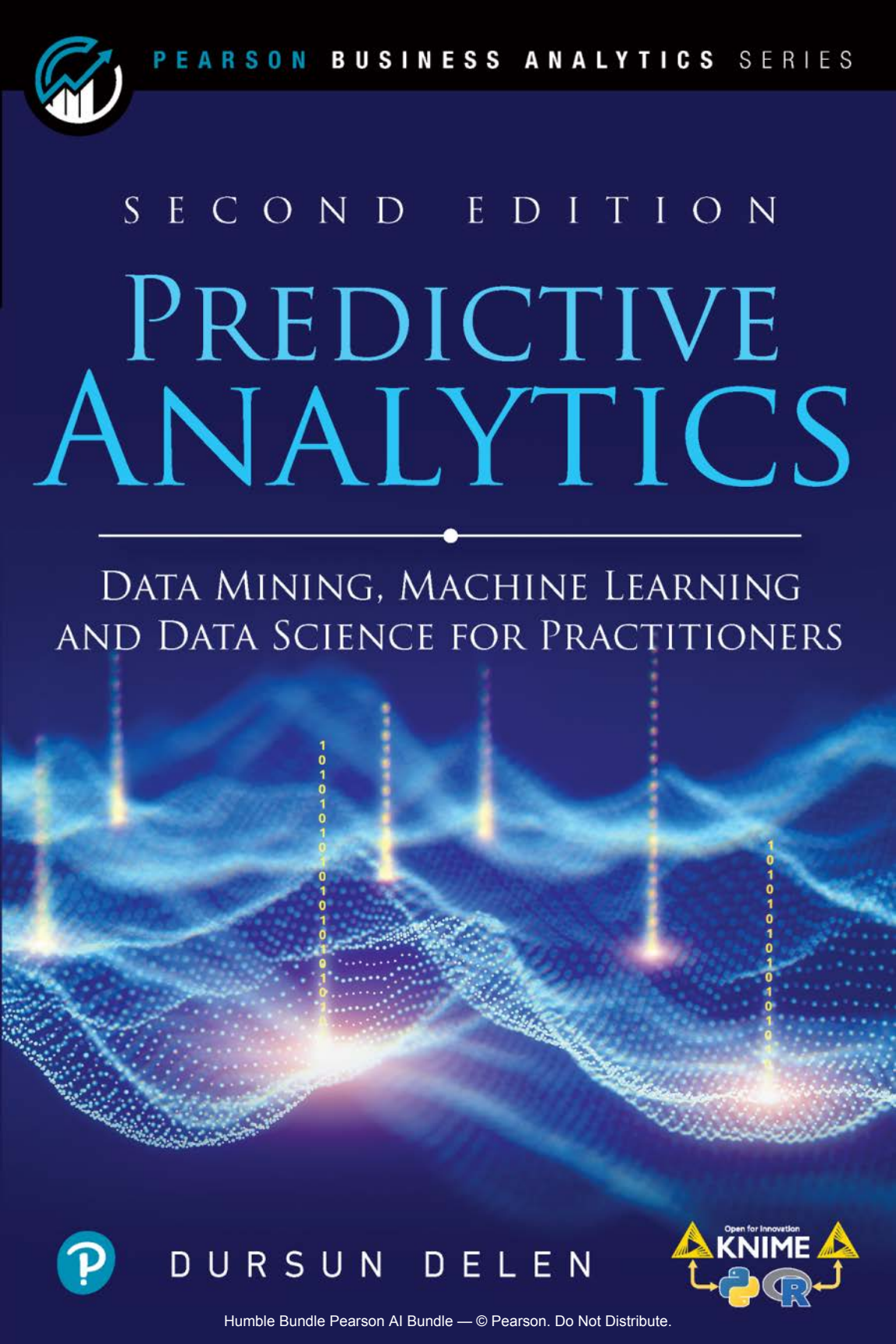


Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Predictive Analytics, Second Edition

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

This page intentionally left blank

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Predictive Analytics, Second Edition

Data Mining, Machine Learning and Data Science for Practitioners

Dursun Delen



Pearson

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

© 2021 by Pearson Education, Inc.
Old Tappan, New Jersey

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Company and product names mentioned herein are the trademarks or registered trademarks of their respective owners.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearson.com/permissions/.

ScoutAutomatedPrintCode

ISBN-10: 0-13-673851-6

ISBN-13: 978-0-13-673851-0

Pearson Education LTD.

Pearson Education Australia PTY, Limited

Pearson Education Singapore, Pte. Ltd.

Pearson Education Asia, Ltd.

Pearson Education Canada, Ltd.

Pearson Educación de Mexico, S.A. de C.V.

Pearson Education—Japan

Pearson Education Malaysia, Pte. Ltd.

Library of Congress Control Number: 2020947805

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

I dedicate this book
to my parents, Ayse and Suleyman,
to my wife, Handan,
to our son, Altug, and to our daughter, Serra,
and to my teachers, who inspired me to be a teacher.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

This page intentionally left blank

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Contents at a Glance

Foreword xi

Chapter 1: Introduction to Analytics 1

Chapter 2: Introduction to Predictive Analytics and Data Mining 33

Chapter 3: Standardized Processes for Predictive Analytics 69

Chapter 4: Data and Methods for Predictive Analytics 95

Chapter 5: Algorithms for Predictive Analytics 147

Chapter 6: Advanced Topics in Predictive Modeling 189

Chapter 7: Text Analytics, Topic Modeling, and Sentiment Analysis 235

Chapter 8: Big Data for Predictive Analytics 289

Chapter 9: Deep Learning and Cognitive Computing 325

Appendix A: KNIME and the Landscape of Tools for Business Analytics and Data Science 391

Index 409

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Contents

Foreword ..... **xi**

Chapter 1 Introduction to Analytics **1**

What's in a Name? 3
Why the Sudden Popularity of Analytics and Data Science? 7
The Application Areas of Analytics 9
The Main Challenges of Analytics 10
A Longitudinal View of Analytics 12
A Simple Taxonomy for Analytics 17
The Cutting Edge of Analytics: IBM Watson 22
Summary 31
References 31

Chapter 2 Introduction to Predictive Analytics and Data Mining **33**

What Is Data Mining? 37
What Data Mining Is Not 39
The Most Common Data Mining Applications 41
What Kinds of Patterns Can Data Mining Discover? 46
Popular Data Mining Tools 52
The Dark Side of Data Mining: Privacy Concerns 58
Summary 66
References 67

Chapter 3 Standardized Processes for Predictive Analytics **69**

The Knowledge Discovery in Databases (KDD) Process 69
Cross-Industry Standard Process for Data Mining (CRISP-DM) 71
SEMMA 80
SEMMA Versus CRISP-DM 84
Six Sigma for Data Mining 84
Which Methodology Is Best? 88
Summary 93
References 94

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Chapter 4 Data and Methods for Predictive Analytics 95

The Nature of Data in Data Analytics 95
Preprocessing of Data for Analytics 99
Data Mining Methods 105
Prediction 106
Classification 108
Decision Trees 119
Cluster Analysis for Data Mining 121
k-Means Clustering Algorithm 127
Association 128
Apriori Algorithm 132
Data Mining and Predictive Analytics Misconceptions and Realities 133
Summary 144
References 145

Chapter 5 Algorithms for Predictive Analytics 147

Naive Bayes 148
Nearest Neighbor 152
Similarity Measure: The Distance Metric 154
Artificial Neural Networks 158
Support Vector Machines 163
Linear Regression 169
Logistic Regression 177
Time-Series Forecasting 179
Summary 185
References 186

Chapter 6 Advanced Topics in Predictive Modeling 189

Model Ensembles 189
Bias–Variance Trade-off in Predictive Analytics 209
Imbalanced Data Problems in Predictive Analytics 215
Explainability of Machine Learning Models for Predictive Analytics 221
Summary 232
References 233

Chapter 7 Text Analytics, Topic Modeling, and Sentiment Analysis 235

Natural Language Processing 241
Text Mining Applications 247

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The Text Mining Process 252
Text Mining Tools 266
Topic Modeling 268
Sentiment Analysis 272
Summary 285
References 286

Chapter 8 Big Data for Predictive Analytics 289

Where Does Big Data Come From? 290
The *Vs* That Define Big Data 292
Fundamental Concepts of Big Data 296
The Business Problems That Big Data Analytics Addresses 302
Big Data Technologies 302
Data Scientists 312
Big Data and Stream Analytics 315
Data Stream Mining 318
Summary 322
References 322

Chapter 9 Deep Learning and Cognitive Computing 325

Introduction to Deep Learning 326
Basics of "Shallow" Neural Networks 331
Elements of an Artificial Neural Network 335
Deep Neural Networks 346
Convolutional Neural Networks 351
Recurrent Networks and Long Short-Term Memory Networks 361
Computer Frameworks for Implementation of Deep Learning 368
Cognitive Computing 373
Summary 386
References 387

Appendix A KNIME and the Landscape of Tools for Business Analytics and Data Science 391

Index 409

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Foreword

Business analytics in general, and predictive analytics specifically, is all about foreseeing the future and making smarter and faster business decisions. The traditional notion of learning from mistakes no longer applies; the reality nowadays is more like "One strike and you are out." Surviving the current turbulent business environment requires absolute precision in the way a manager recognizes problems and opportunities and calculates and implements the most optimal decisions in a timely manner. The organizations that use business analytics not only can survive but often thrive in this type of condition. Business analytics, once considered an advanced/optional capability, is now a commodity—a must-have organizational capability. With business analytics, firms can utilize data to understand what happened, foresee what will happen, and implement what should happen. It is the current gold standard for business management.

Analytics (or perhaps more appropriately, data analytics) can simply be defined as the discovery of meaningful patterns—new and novel information and knowledge—in data. Because we are living in an era of big data, the analytics definitions are mostly focused on the value proposition of data, which is being created in large volumes and varieties with a high velocity. Business analytics is a special application of analytics that leverages the underlying tools, techniques, and principles to develop solutions to ever-so-complex business problems. Firms commonly apply analytics to business data to describe, predict, and optimize their business opportunities and maximize their performance. Business analytics is the most pronounced catchphrase of business circles nowadays. No matter what business journal or magazine you look at, it is very likely that you will see articles about analytics and how it is changing the way managerial decisions are being made. It has become a new label for evidence-based management (i.e., evidence/data-driven decision making). The question is, why has analytics become so popular, and why now? The main reasons (or

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

forces) behind this popularity can be grouped into three categories: need, availability/affordability, and culture change.

Business analytics is often characterized by three levels/echelons representing the hierarchical nature of the term—descriptive, predictive, and prescriptive. Organizations usually start with descriptive analytics, then move into predictive analytics, and finally reach prescriptive analytics. Even though these three echelons of analytics are hierarchical (in terms of level of complexity and sophistication), moving from a lower level to a higher level is not clearly separated. That is, a business can be in the descriptive analytics level while at the same time using predictive and even prescriptive analytics capabilities in a somewhat piecemeal fashion. Predictive analytics, which is the main topic of this book, comes after descriptive analytics and before prescriptive analytics. Organizations that have matured in descriptive analytics move into this critical level where they look beyond what happened and try to answer the question of "What will happen?" This book covers the predictive capabilities of these analytics techniques in depth.

Perhaps because of its rapidly increasing popularity as a buzzword, analytics is being used to replace several previously popular terms, such as intelligence, mining, and discovery. For example, the term *business intelligence* is now *business analytics*, *customer intelligence* is *customer analytics*, *web mining* is *web analytics*, and *knowledge discovery* is *data analytics*. Even the name *business analytics* is being contested by other buzzwords like *data science*, *machine learning*, and *cognitive computing*. But no matter the terminology used, the goal is always the same: creating actionable insight from large and feature-rich data. Because modern analytics can require extensive computation (because of the volume, variety, and velocity at which the data is created—i.e., the big data), the tools, techniques, and algorithms used for analytics projects leverage the most current, state-of-the-art methods developed in a wide variety of fields that include management science, computer science, statistics, machine learning, data science, and mathematics.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Acknowledgments

This book is the manifestation of my over three decades of analytics and data science experiences, created through numerous applied research projects, consultancy engagements, academic and professional teaching activities, corporate and educational mentoring initiatives, and undergraduate, graduate, and executive student advising. Henceforth, I want to acknowledge my colleagues, clients, and students for their direct and indirect contributions to the formation of my analytics knowledge base that laid the foundation for this book.

I feel obligated to also acknowledge the people from whom I have received emotional and psychological support, inspiration, and encouragement throughout the process of writing this book. I have been blessed to have way too many people who qualify for this acknowledgment; therefore, as opposed to trying to list all of them here, I want to mention just a few names: my long-time college friends, Erhan Sayin, Ahmet Murat Fis, and Ali Mutlu, and my beloved brothers-in-law, Dincer Hamamci and Yilmaz Tomak.

I also would like to thank the highly knowledgeable staff at Pearson for their professionalism and dedication to making this book a reality. In particular, I want to thank Ms. Kim Spenceley and her team for their continuous help and unwavering attention to details.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

About the Author

Dr. Dursun Delen is an internationally renowned expert in business analytics, data science, and machine learning. He is often invited to national and international conferences to deliver keynote presentations on topics related to data/text mining, business intelligence, decision support systems, business analytics, data science, and knowledge management. Prior to his appointment as a professor at Oklahoma State University in 2001, Dr. Delen worked for industry for more than 10 years, developing and delivering business analytics solutions to companies. His most recent industrial work was at a privately owned applied research and consulting company, Knowledge Based Systems, Inc. (KBSI), in College Station, Texas, as a research scientist. During his five years at KBSI, Dr. Delen led a number of projects related to decision support systems, enterprise engineering, information systems development, and advanced business analytics that were funded by private industry and federal agencies, including several branches of the Department of Defense, NASA, National Science Foundation, National Institute for Standards and Technology, and the Department of Energy. Today, in addition to his academic endeavors, Dr. Delen provides professional education and consulting services to businesses in assessing their analytics, data science, and information system needs and helping them develop state-of-the-art computerized decision support systems.

In his current academic position, Dr. Delen holds the William S. Spears Endowed Chair in Business Administration and the Patterson Family Endowed Chair in Business Analytics, and he is the director of research for the Center for Health Systems Innovation and regents' professor of management science and information systems in the Spears School of Business at Oklahoma State University. He has published more than 150 peer-reviewed research articles that have appeared in major journals, including *Journal of Business Research*,

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Journal of Business Analytics, *Decision Sciences Journal*, *Decision Support Systems*, *Communications of the ACM*, *Computers & Operations Research*, *Annals of Operations Research*, *Computers in Industry*, *Journal of Production Operations Management*, *Artificial Intelligence in Medicine*, *Journal of the American Medical Informatics Association*, *Expert Systems with Applications*, *Renewable and Sustainable Energy Reviews*, *Energy*, and *Renewable Energy*, among others. He has also authored and coauthored 11 books and textbooks in the broad area of business analytics, data science, and business intelligence.

Dr. Delen regularly chairs tracks and minitracks at various business analytics and information systems conferences. Currently, he is the editor-in-chief for the *Journal of Business Analytics* and AI in Business (in *Frontiers in Artificial Intelligence*), senior editor for the *Journal of Decision Support Systems*, *Decision Sciences*, and *Journal of Business Research*, associate editor for *Decision Analytics*, *International Journal of Information and Knowledge Management*, and *International Journal of RF Technologies*, and is on the editorial boards of several other academic journals. He has been the recipient of several research and teaching awards, including the prestigious Fulbright scholar, regents' distinguished teacher and researcher, president's outstanding researcher, and Big Data mentor awards.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Credits

Table 4.5: Delen, D., D. Cogdell, & N. Kasap, "A Comparative Analysis of Data Mining Methods in Predicting NCAA Bowl Outcomes," International Journal of Forecasting, 28 (2012): 543–552.

Table 4.6: Delen, D., D. Cogdell, & N. Kasap, "A Comparative Analysis of Data Mining Methods in Predicting NCAA Bowl Outcomes," International Journal of Forecasting, 28 (2012): 543–552.

Figure 7.6: © 2020 Microsoft

Figure 8.2: Watson, H. (2012). "The Requirements for Being an Analytics-Based Organization," Business Intelligence Journal, 17(2): 42–44

Figure 9.17: Screenshot of two examples of using the Google Lens, © 2020 Google LLC

Figure 9.22: Courtesy of Microsoft Corporation

Figure A.1: Screenshot of KNIME Analytics Platform © 2020 - KNIME AG

Figure A.2: Screenshot of resulting list of workflows from searching for "deep learning" on the KNIME Hub © 2020 - KNIME AG

Figure A.3: Screenshot of the Mining category in the Node Repository holding the machine learning algorithms © 2020 - KNIME AG

Figure A.4: Screenshot of the composite view from a component containing a scatter plot, a bar chart, and a parallel coordinate plot © 2020 - KNIME AG

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

1

Introduction to Analytics

Predictive analytics, the focus of this book, is a vital part of the business analytics continuum. The chief goal of predictive analytics, as the term *predictive* indicates, is to foresee future events and situations and, thereby, help decision makers to take advantage of the upcoming opportunities in a timely manner or to prevent (or at least mitigate the impact of) the problems that are about to surface. As we will discuss in more detail later in this chapter, within the business analytics continuum, predictive analytics is strategically positioned between descriptive analytics (which is primarily focused on what happened) and prescriptive analytics (which is focused on what decision should be made). Predictive analytics leverages/consumes what descriptive analytics generates (i.e., historical snapshots of information/patterns) in order to produce what is most likely to happen in the future as its output (i.e., predictions/forecasts) and feeds the output into prescriptive analytics as one of its key enablers/inputs to create insight for optimal decisions.

Compared to business intelligence, *business analytics* is a newer term that is very rapidly gaining popularity in the business world. In general terms, *analytics* is the art and science of discovering insight by using sophisticated mathematical models along with a variety of data and expert knowledge to support accurate and timely decision making. In a sense, analytics is all about decision making and problem solving. Today, *analytics* can be defined simply as "the discovery of information/knowledge/insight in data." In this era of abundant

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

data, which some call a *data deluge*, analytics tends to be used to sift through rather large volumes and varieties of data.

Although analytics tends to be data focused, many applications of analytics involve very little or no data; instead, those analytics projects use mathematical and symbolic models that rely on process description and expert knowledge (e.g., optimization and simulation models, expert systems, case-based decision analysis). Therefore, in order to make sure that an analytics application is data driven, a newer and more specific term—*data analytics*—has been coined.

Business analytics is the application of the tools, techniques, and principles of analytics to complex business problems. Organizations commonly apply analytics to business problems (often in data-rich domains) to describe, predict, and optimize business performance. Firms have used analytics in many ways, including the following:

- To improve their relationships with their customers (encompassing all phases of customer relationship management, including acquisition, retention, and enrichment), suppliers, employees, and other stakeholders
- To identify fraudulent transactions and undesirable behaviors/outcomes and, by doing so, save money and improve outcomes
- To enhance product and service features and their pricing, which leads to better customer satisfaction, loyalty-driven marketplace position, and better profitability
- To optimize marketing and advertising campaigns so that the firm can reach more customers with the right kind of message and promotions at the least expense
- To minimize operational costs by optimally managing inventories and allocating resources wherever and whenever they are needed by using optimization and simulation modeling
- To empower employees with the information and insight they need to make faster and better decisions while they are working with customers or customer-related issues

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The term *analytics*, perhaps because of its rapidly increasing popularity as a buzzword, is being used in place of several previously popular terms, such as *intelligence*, *mining*, and *discovery*. For example, the term *business intelligence* has now become *business analytics*; *customer intelligence* has become *customer analytics*, *web mining* has become *web analytics*, and *knowledge discovery* has become *data analytics*. While the term *analytics* has been gaining popularity, several newer terms—including *data science*, *computational intelligence*, *big data analytics*, and *applied machine learning*, to name a few—have emerged to crowd and further complicate the name space. As I describe in this book, the rapid change in the names provides yet more evidence for the expeditious developments and amplified interests in the value proposition of analytics.

What's in a Name?

The relentless creation of buzzwords has led to difficulty understanding the differences and similarities among the various terms. Among the terms that are most often confused with each other—and characterized at the highest level of conceptualization—are business intelligence, business analytics, and data science. In addition to these top terms, terms such as big data, machine learning, Internet of Things (IoT), and natural language processing have become more common. As an attempt to impose some order on these complex terms, we created the rather simple Venn diagram shown in Figure 1.1. The top portion of the figure shows the interrelationships among the most popular terms. Business intelligence is portrayed as a wholly consumed subcomponent of business analytics (i.e., the descriptive analytics phase in business analytics), and business analytics is portrayed as a partially consumed subcomponent of data science. The reason for partial consumption between business analytics and data science is that business analytics also relies on solutions that do not require data but are instead driven by the representation and consumption of business processes and expert knowledge.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

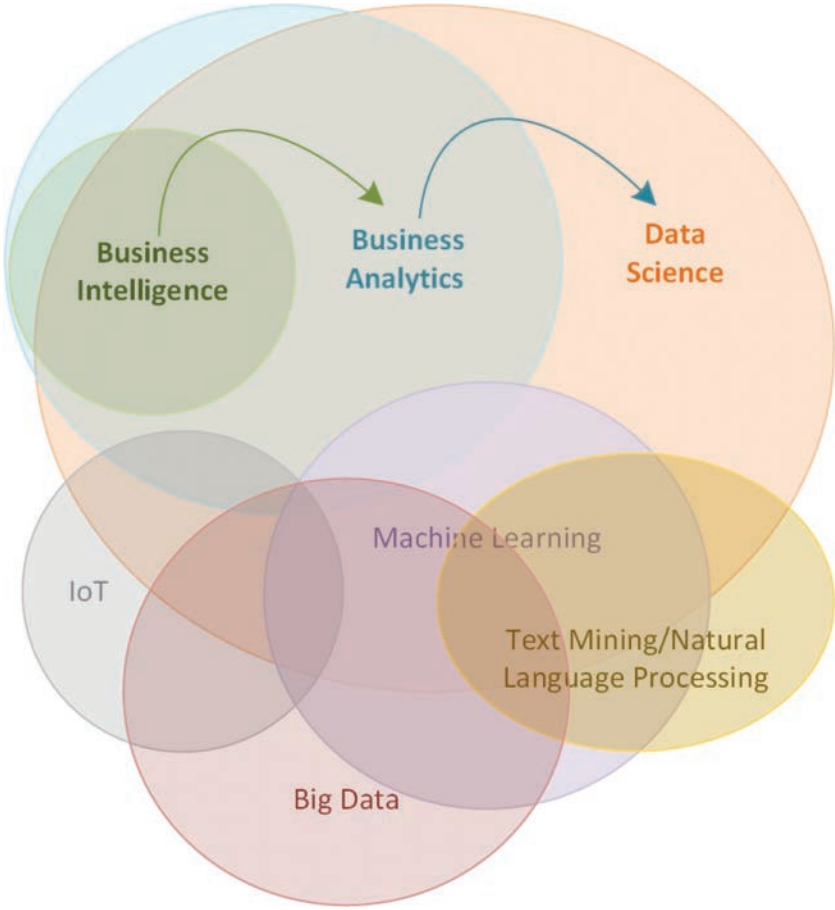


Figure 1.1 Overlaps Among the Popular Concepts and Terms in Analytics and Data Science

The terms shown in the bottom portion of Figure 1.1—big data, machine learning, natural language processing, and Internet of Things—refer to the enablers and/or mechanisms for the three key terms shown at the top of the figure. Although the term *big data* is sometimes used to refer to *big data analytics*, its rightful characterization stems from the volume, variety, and velocity of data that enhances the capabilities of business analytics and data science to a new level of rightness and comprehensiveness. Whereas business intelligence deals with structured data (stored in databases and Excel sheets in the form of rows and columns), business analytics and data science deal

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

with both structured and unstructured data. Big data can be defined by not only structured data but also unstructured data—which is often composed of textual, multimedia, and IoT data—and brings value (and some challenges) to the world of analytics and data science.

Two of these terms are often used synonymously and interchangeably: *business analytics* and *data science*. Are they the same? If so, why do we have two different names for the same concept? Business analytics and data science have the same purpose: to convert data into actionable insight through an algorithm-based discovery process. The two terms are different, however, in scope. Much like the scope enlargement from business intelligence to business analytics, there is a scope enhancement from business analytics to data science. Whereas business analytics adds predictive and prescriptive modeling on top of business intelligence, data science adds big data enablers, low-level programming (e.g., Python, R, SQL/NoSQL, JavaScript, Perl), and advanced machine learning (i.e., deep learning and its variants) on top of business analytics. Another difference between the two terms is in their application domain and the tools employed; while business analytics deals with business problems using workflow-type tools and readily available algorithms, data science deals with a broader range of problems using low-level programming, enhanced algorithms, and custom solution development processes.

Another difference between business analytics and data science can be observed in higher education. While business schools usually offer certificate and graduate/undergraduate degree programs in business analytics, computer science, decision sciences, and industrial engineering schools offer similar programs under the data science umbrella. A quick look at these curriculums reveals that while business analytics programs are more applied, problem-solving, and high-level methodology focused, data science programs are more programming, algorithm development, and low-level method focused. While these observations are generally true, however, they are not universally applicable, standardized, or practiced.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Modern-day analytics and data science projects often require extensive computations because of the volume, variety, and velocity of data and the need to produce results as quickly as possible. Therefore, the tools, techniques, and algorithms used for analytics and data science must leverage advanced state-of-the-art software and hardware alongside the latest methods developed in a variety of fields, including management science, computer science, statistics, and mathematics.

Is There a Difference Between Analytics and Analysis?

Even though the two terms *analytics* and *analysis* are often used interchangeably, they are not the same.

Basically, *analysis* refers to the process of separating a whole problem into its parts so that the parts can be critically examined at a granular level. It is often used when the investigation of a complete system is not feasible or practical, and the system needs to be simplified by being decomposed into more basic components. Once the improvements at the granular level are realized and the examination of the parts is complete, the whole system (either a conceptual or physical system) can be put together using a process called *synthesis*.

Analytics, on the other hand, encompasses a variety of methods, technologies, and associated tools for creating new knowledge/insight to solve complex problems and make better decisions more quickly. In essence, analytics is a multifaceted and multidisciplinary approach to addressing complex situations. Analytics takes advantage of data and mathematical models to make sense of the complicated world we are living in. Even though analytics includes analysis at different stages of the discovery process, it is not just analysis but it also includes synthesis and other complementary tasks and processes.

Where Does Data Mining Fit In?

Data mining is the process of discovering new knowledge in the forms of patterns and relationships in large data sets. The goal

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

of analytics is to convert data/facts into actionable insight, and data mining is the key enabler of that goal. Data mining, which has been around much longer than analytics, dates to 1980s. As *analytics* became an overarching term for all decision support and problem-solving methods and technologies, data mining found itself a rather broad space within that arc, ranging from descriptive exploration to identify relationships and affinities among variables (e.g., market-basket analysis) to development of models to estimate future values of interesting variables. As we will see later in this chapter, within the taxonomy of analytics, data mining plays a key role at every level, from the most simple to the most sophisticated.

Why the Sudden Popularity of Analytics and Data Science?

Analytics is a buzzword in business circles today. No matter what business journal or magazine you look at, it is very likely that you will see articles about analytics and how analytics is changing the way managerial decisions are being made. It has become a new label for evidence-based management (or evidence-/data-driven decision making). But why has analytics become so popular? And why now? The reasons (or forces) behind this popularity can be grouped into three categories: need, availability and affordability, and culture change.

Need

As we all know, business is anything but "as usual" today. Competition has been characterized progressively as local, then regional, then national, but it is now global. Large to medium to small businesses face the pressures of global competition. The tariff and transportation cost barriers that sheltered companies in their respective geographic locations are no longer as protective as they once were. In addition to (and perhaps because of) the global competition, customers have

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

become more demanding. They want the highest-quality products and/or services with the lowest prices in the shortest possible time. Success or mere survival depends on businesses being agile and their managers making the best possible decisions in a timely manner to respond to market-driven forces (i.e., rapidly identifying and addressing problems and taking advantage of opportunities). Therefore, the need for fact-based, better, and faster decisions is more critical now than ever before. In the midst of these unforgiving market conditions, analytics is promising to provide managers the insights they need to make better and faster decisions, which helps improve their competitive posture in the marketplace. Analytics today is widely perceived as aiding business managers as they navigate the complexities of global business practices.

Availability and Affordability

Thanks to recent technological advances and the affordability of software and hardware, organizations are collecting tremendous amounts of data. Automated data collections systems—based on a variety of radio frequency identification (RFID) and other sensors—have significantly increased the quantity and quality of organizational data. Coupled with the content-rich data collected from Internet-based technologies such as social media, businesses now tend to have more data than they can handle. As the saying goes, "They are drowning in data but starving for knowledge."

Along with data collection technologies, data processing technologies have also improved significantly. Today's machines have numerous processors and very large memory capacities, and they are able to process very large and complex data in a reasonable time frame—often in real time. The advances in both hardware and software technology are also reflected in the pricing, continuously reducing the cost of ownership for such systems. In addition to the ownership model, we now have the software- (or hardware-) as-a-service business model,

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

which allows businesses (especially small to medium-size businesses with limited financial power) to rent analytics capabilities and pay only for what they use.

Culture Change

At the organizational level, there has been a shift from old-fashioned intuition-driven decision making to new-age fact-/evidence-based decision making. Most successful organizations have made a conscious effort to shift to data-/evidence-driven business practices. Because of the availability of data and supporting IT infrastructure, this paradigm shift is taking place faster than many thought it would. As the new generation of quantitatively savvy managers replaces the baby boomers, this evidence-based managerial paradigm shift will intensify.

The Application Areas of Analytics

Even though the business analytics wave is somewhat new, there are numerous applications of analytics, covering almost every aspect of business practice. For instance, in customer relationship management, a wealth of success stories tell of sophisticated models developed to identify new customers, look for up-sell/cross-sell opportunities, and find customers with a high propensity toward attrition. Using social media analytics and sentiment analysis, businesses are trying to stay on top of what people are saying about their products/services and brands. Fraud detection, risk mitigation, product pricing, marketing campaign optimization, financial planning, employee retention, talent recruiting, and actuarial estimation are among the many business applications of analytics. It would be very hard to find a business issue where a number of analytics application could not be found. From business reporting to data warehousing, from data mining to optimization analytics, techniques are used widely in almost every facet of business.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The Main Challenges of Analytics

Even though the advantages of and reasons for analytics are evident, many businesses are still hesitant to jump on the analytics bandwagon. These are the main roadblocks to adoption of analytics:

- **Analytics talent.** Data scientists, the quantitative geniuses who can convert data into actionable insight, are scarce in the market; the really good ones are very hard to find. Because analytics is relatively new, the talent for analytics is still being developed. Many colleges have started undergraduate and graduate programs to address the analytics talent gap. As the popularity of analytics increases, so will the need for people who have the knowledge and skills to convert big data into information and knowledge that managers and other decision makers need to tackle real-world complexities.
- **Culture.** As the saying goes, "Old habits die hard." Changing from a traditional management style (often characterized by intuition as the basis of making decisions) to a contemporary management style (based on data and scientific models for managerial decisions and collective organizational knowledge) is not an easy process to undertake for any organization. People do not like to change. Change means losing what you have learned or mastered in the past and learning how to do what you do all over again. It suggests that the knowledge (which is also characterized as power) you've accumulated over the years will disappear or be partially lost. The culture shift may be the most difficult part of adopting analytics as the new management paradigm.
- **Return on investment.** Another barrier to adoption of analytics is the difficulty in clearly justifying its return on investment (ROI). Analytics projects are complex and costly endeavors, and their return is not immediately clear; many executives have

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

a hard time investing in analytics, especially on large scales. Will the value gained from analytics outweigh the investment? If so, when? It is very hard to convert the value of analytics into justifiable numbers. Most of the value gained from analytics is somewhat intangible and holistic. If done properly, analytics could transform an organization, putting it on a new and improved level. A combination of tangible and intangible factors needs to be brought to bear to numerically rationalize investment and movement toward analytics and analytically savvy management practice.

- **Data.** The media is talking about big data in a very positive way, characterizing it as an invaluable asset for better business practices. This is mostly true, especially for businesses that understand and know what to do with it. For others, big data is a big challenge. Big data is not just big; it is unstructured, and it is arriving at a speed that prohibits traditional collection and processing means. And it is usually messy and dirty. For an organization to succeed in analytics, it needs to have a well-thought-out strategy for handling big data so that it can be converted to actionable insight.
- **Technology.** Even though technology is capable, available, and, to a large extent, affordable, technology adoption poses another challenge for traditionally less technical businesses. Although establishing an analytics infrastructure is affordable, it still costs a significant amount of money. Without financial means and/or a clear return on investment, management in some businesses may not be willing to invest in needed technology. For some businesses, an analytics-as-a-service model (which includes both software and the infrastructure/hardware needed to implement analytics) may be less costly and easier to implement.
- **Security and privacy.** One of the most common criticisms of data and data analytics relates to security. We often hear about

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

data breaches of sensitive information, and indeed, the only completely secured data infrastructure is isolated and disconnected from all other networks (which goes against the very reason for having data and analytics). The importance of data security has made information assurance one of the most popular concentration areas in information systems departments around the world. At the same time that increasingly sophisticated techniques are being used to protect the information infrastructure, increasingly sophisticated attacks are becoming common. There are also concerns about personal privacy. Use of personal data about customers (existing or prospective), even if it is within legal boundaries, should be avoided or carefully scrutinized to protect an organization against bad publicity and public outcry.

Despite the hurdles, analytics adoption is growing, and use of analytics is inevitable for enterprises of all sizes and in all industry segments. As the complexity in conducting business increases, enterprises will strive to find order in the midst of the chaotic behaviors. The ones that succeed will be the ones fully leveraging the capabilities of analytics.

A Longitudinal View of Analytics

Although the buzz about it is relatively recent, analytics isn't new. It's possible to find references to corporate analytics as far back as the 1940s, during the World War II era, when more effective methods were needed to maximize output with limited resources. Many optimization and simulation techniques were developed then. Analytical techniques have been used in business for a very long time. One example is the time and motion studies initiated by Frederick Winslow Taylor in the late 19th century. Then Henry Ford measured pacing of assembly lines, which led to mass-production initiatives.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Analytics began to command more attention in the late 1960s, when computers were used in decision support systems. Since then, analytics has evolved with the development of enterprise resource planning (ERP) systems, data warehouses, and a wide variety of other hardware and software tools and applications.

The timeline in Figure 1.2 shows the terminology that has been used to describe analytics since the 1970s. During the early days of analytics, prior to the 1970s, data was often obtained from the domain experts using manual processes (i.e., interviews and surveys) to build mathematical or knowledge-based models to solve constraint optimization problems. The idea was to do the best with limited resources. Such decision support models were typically called *operations research* (*OR*). The problems that were too complex to solve optimally (using linear or nonlinear mathematical programming techniques) were tackled using heuristic methods such as simulation models.



Figure 1.2 A Longitudinal View of the Evolution of Analytics

In the 1970s, in addition to the mature OR models that were being used in many industries and government systems, a new and exciting line of models had emerged: rule-based expert systems (ESs). These systems promised to capture experts' knowledge in a format that computers could process (via a collection of if–then rules) so that they could be used for consultation much the same way that one would use domain experts to identify a structured problem and to prescribe the most probable solution. ESs allowed scarce expertise to be made available where and when needed, using an "intelligent" decision

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

support system. During the 1970s, businesses also began to create routine reports to inform decision makers (managers) about what had happened in the previous period (e.g., day, week, month, quarter). Although it was useful to know what had happened in the past, managers needed more than this: They needed a variety of reports at different levels of granularity to better understand and address the changing needs and challenges of the business.

The 1980s saw a significant change in the way organizations captured business-related data. The old practice had been to have multiple disjointed information systems tailored to capture transactional data of different organizational units or functions (e.g., accounting, marketing and sales, finance, manufacturing). In the 1980s, these systems were integrated as enterprise-level information systems that we now commonly call *ERP systems*. The old mostly sequential and nonstandardized data representation schemas were replaced by relational database management (RDBM) systems. These systems made it possible to improve the capture and storage of data, as well as the relationships between organizational data fields while significantly reducing the replication of information. The need for RDBM and ERP system emerged when data integrity and consistency became an issue, significantly hindering the effectiveness of business practices. With ERP, all the data from every corner of the enterprise is collected and integrated into a consistent schema so that every part of the organization has access to the single version of the truth when and where needed. In addition to the emergence of ERP systems—or perhaps because of these systems—business reporting became an on-demand, as-needed business practice. Decision makers could decide when they needed to or wanted to create specialized reports to investigate organizational problems and opportunities.

In the 1990s, the need for more versatile reporting led to the development of executive information systems (decision support systems designed and developed specifically for executives and their

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

decision-making needs). These systems were designed as graphical dashboards and scorecards so that they could serve as visually appealing displays focused on the most important factors for decision makers to keep track of: the key performance indicators. In order to make this highly versatile reporting possible while keeping the transactional integrity of the business information systems intact, it was necessary to create a middle data tier—known as a *data warehouse* (*DW*)—as a repository to specifically support business reporting and decision making. In a very short time, most large to medium-size businesses adopted data warehousing as their platform for enterprise-wide decision making. The dashboards and scorecards got their data from a data warehouse, and by doing so, they were not hindering the efficiency of the business transaction systems—mostly referred to as ERP systems.

In the 2000s the DW-driven decision support systems began to be called *business intelligence systems*. As the amount of longitudinal data accumulated in the DWs increased, so did the capabilities of hardware and software to keep up with the rapidly changing and evolving needs of the decision makers. Because of the globalized competitive marketplace, decision makers needed current information in a very digestible format to address business problems and to take advantage of market opportunities in a timely manner. Because the data in a DW is updated periodically, it does not reflect the latest information. In order to elevate this information latency problem, DW vendors developed a system to update the data more frequently, which led to the terms *real-time data warehousing* and, more realistically, *right-time data warehousing*, which differs from the former by adopting a data refreshing policy based on the needed freshness of the data items (i.e., not all data items need to be refreshed in real time). Data warehouses are very large and feature rich, and it became necessary to "mine" the corporate data to "discover" new and useful knowledge nuggets to improve business processes and practices—hence the terms *data mining* and *text mining*. With the increasing volumes

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

and varieties of data, the need for more storage and more processing power emerged. While large corporations had the means to tackle this problem, small to medium-size companies needed financially more manageable business models. This need led to service-oriented architecture and software and infrastructure-as-a-service analytics business models. Smaller companies therefore gained access to analytics capabilities on an as-needed basis and paid only for what they used, as opposed to investing in extremely expensive hardware and software resources.

The 2010s and 2020s are seeing yet another paradigm shift in the way that data is captured and used. Largely because of the widespread use of the Internet, new data-generation mediums have emerged. Of all the new data sources (e.g., RFID tags, digital energy meters, clickstream web logs, smart home devices, wearable health monitoring equipment), perhaps the most interesting and challenging is social networking/social media. This unstructured data is rich in information content, but analysis of such data sources poses significant challenges to computational systems, from both software and hardware perspectives. Recently, the term *big data* has been coined to highlight the challenges that these new data streams have brought upon us. Many advancements in both hardware (e.g., massively parallel processing with very large computational memory and highly parallel multiprocessor computing systems) and software/algorithms (e.g., Hadoop with MapReduce and NoSQL) have been developed to address the challenges of big data.

It's hard to predict what the next decade will bring and what the new analytics-related terms will be. The time between new paradigm shifts in information systems and particularly in analytics has been shrinking, and this trend will continue for the foreseeable future. Even though analytics is not new, the explosion in its popularity is very new. Thanks to the recent explosion in big data, ways to collect and store this data, and intuitive software tools, data and data-driven

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

insight are more accessible to business professionals than ever before. Therefore, in the midst of global competition, there is a huge opportunity to make better managerial decisions by using data and analytics to increase revenue while decreasing costs by building better products, improving customer experience, catching fraud before it happens, and improving customer engagement through targeting and customization—all with the power of analytics and data. More and more companies are now preparing their employees with business analytics know-how to drive effectiveness and efficiency in their day-to-day decision-making processes.

A Simple Taxonomy for Analytics

Because of the multitude of factors related to both the need to make better and faster decisions and the availability and affordability of hardware and software technologies, analytics is gaining popularity faster than any other trends we have seen in recent history. Will this upward exponential trend continue? Many industry experts think it will, at least for the foreseeable future. Industry experts and leading consulting companies are projecting that analytics will grow faster than any other business segments in upcoming years; they have also named analytics (and data science) as one of the top business trends of this decade. As interest in and adoption of analytics have grown rapidly, a need to characterize analytics into a simple taxonomy has emerged. The top consulting companies (e.g., Accenture, Gartner, Forrester, IDT) and several technologically oriented academic institutions have embarked on a mission to create a simple taxonomy for analytics. Such a taxonomy, if developed properly and adopted universally, could create a contextual description of analytics, thereby facilitating a common understanding of what analytics is, including what is included in analytics and how analytics-related terms (e.g., business intelligence, predictive modeling, data mining) relate to

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

each other. One of the academic institutions involved in this challenge is INFORMS (Institute for Operations Research and Management Science). In order to reach a wide audience, INFORMS hired Capgemini, a strategic management consulting firm, to carry out a study and characterize analytics.

The Capgemini study produced a concise definition of analytics: "Analytics facilitates realization of business objectives through reporting of data to analyze trends, creating predictive models for forecasting and optimizing business processes for enhanced performance." As this definition implies, one of the key findings from the study is that executives see analytics as a core function of businesses that use it. It spans many departments and functions within organizations, and in mature organizations, it spans the entire business. The study identified three hierarchical but sometimes overlapping groupings for analytics categories: descriptive, predictive, and prescriptive analytics. These three groups are hierarchical in terms of the level of analytics maturity of the organization. Most organizations start with descriptive analytics, then move into predictive analytics, and finally reach prescriptive analytics, the top level in the analytics hierarchy. Even though these three groupings of analytics are hierarchical in complexity and sophistication, moving from a lower level to a higher level is not clearly separable. That is, a business can be in the descriptive analytics level while at the same time using predictive and even prescriptive analytics capabilities, in a somewhat piecemeal fashion. Therefore, moving from one level to the next essentially means that the maturity at one level is completed and the next level is being widely exploited. Figure 1.3 provides a graphical depiction of the simple taxonomy developed by INFORMS and widely adopted by most industry leaders as well as academic institutions.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

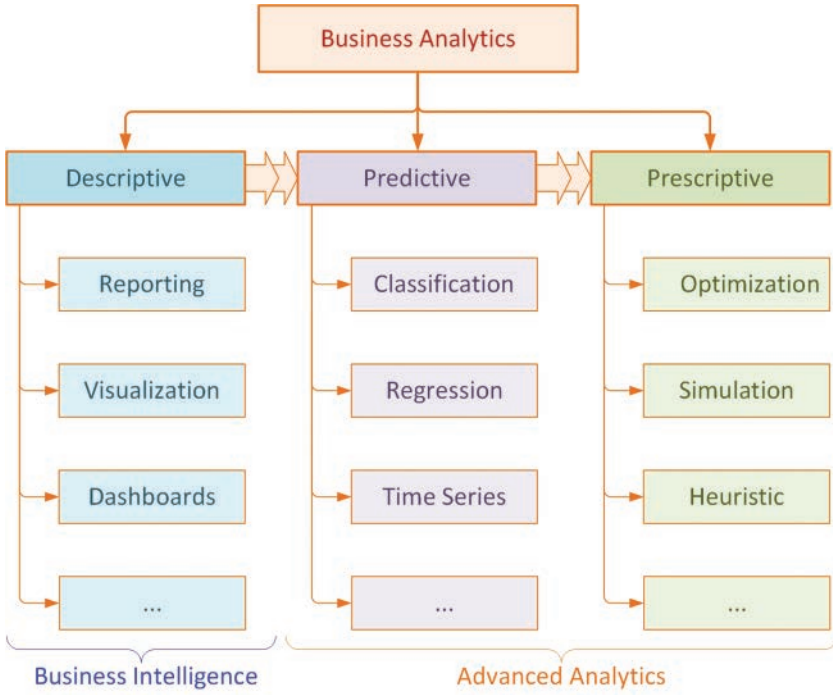


Figure 1.3 A Simple Taxonomy for Analytics

Descriptive analytics is the entry level in analytics taxonomy. It is often called *business reporting* because of the fact that most of the analytics activities at this level deal with creating reports to summarize business activities in order to answer questions such as "What happened?" and "What is happening?" The spectrum of these reports includes *static* snapshots of business transactions delivered to knowledge workers (i.e., decision makers) on a fixed schedule (e.g., daily, weekly, quarterly); *dynamic* views of business performance indicators delivered to managers and executives in a easily digestible form—often in a dashboard-type graphical interface—and in a continuous manner; and ad hoc reporting, where the decision maker is given the capability of creating his or her own specific report (using an intuitive drag-and-drop graphical user interface) to address a specific or unique decision situation.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Descriptive analytics is also called *business intelligence* (*BI*), and predictive and prescriptive analytics are collectively called *advanced analytics*. The logic here is that moving from descriptive to predictive and/or prescriptive analytics is a significant shift in the level of sophistication and therefore warrants the label *advanced*. BI has been one of the most popular technology trends for information systems designed to support managerial decision making since the start of the century. It was popular (to some extent, it still is in some business circles) until the arrival of the analytics wave. BI is the gateway to the world of analytics, setting the stage and paving the way toward more sophisticated decision analysis. Descriptive analytics systems usually work off a data warehouse, which is a large database specifically designed and developed to support BI functions and tools.

Predictive analytics comes right after descriptive analytics in the three-level analytics hierarchy. Organizations that are mature in descriptive analytics move to this level, where they look beyond what happened and try to answer the question "What will happen?" The following chapters cover the predictive capabilities of these analytics techniques in depth as part of data mining; this chapter provides only a very short description of the main predictive analytics classes. Prediction essentially is the process of making intelligent/scientific estimates about the future values of some variables, such as customer demand, interest rates, and stock market movements. If what is being predicted is a categorical variable, the act of prediction is called *classification*; otherwise, it is called *regression*. If the predicted variable is time dependent, the prediction process is often called *time-series forecasting*.

Prescriptive analytics is the highest echelon in analytics hierarchy. It is where the best alternative among many courses of action—that are usually created/identified by predictive and/or descriptive analytics—is determined using sophisticated mathematical models. Therefore, in a sense, this type of analytics tries to answer the question

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

"What should I do?" Prescriptive analytics uses optimization-, simulation-, and heuristics-based decision-modeling techniques. Even though prescriptive analytics is at the top of the analytics hierarchy, the methods behind it are not new. Most of the optimization and simulation models that constitute prescriptive analytics were developed during and right after World War II, when there was a dire need for a lot with limited resources. Since then, some businesses have used these models for some very specific problem types, including yield/revenue management, transportation modeling, and scheduling. The new taxonomy of analytics has made them popular again, opening their use to a wide array of business problems and situations.

Figure 1.4 provides a tabular representation of the three hierarchical levels of analytics, along with the questions answered and techniques used at each level. It shows that data mining is the key enabler of predictive analytics.

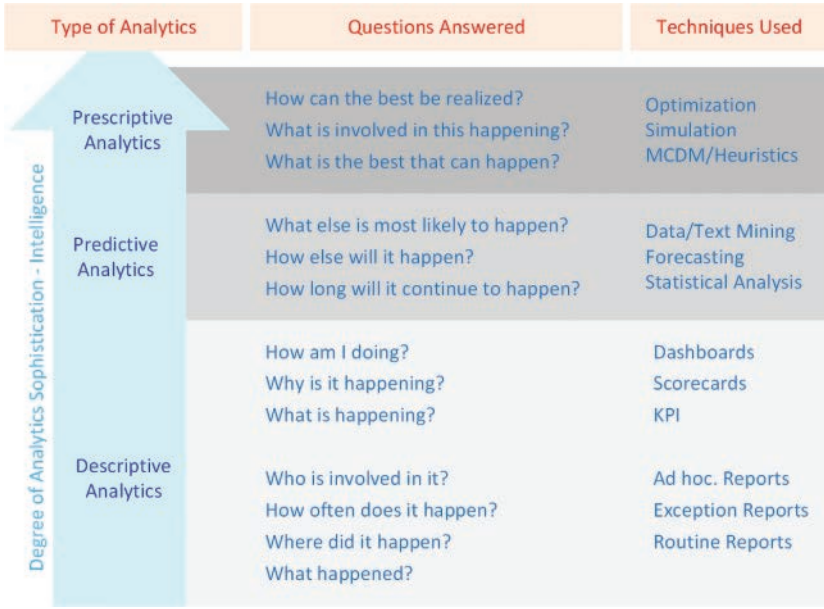


Figure 1.4 Three Levels of Analytics and Their Enabling Techniques

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Business analytics is gaining popularity because it promises to provide decision makers with information and knowledge that they need to succeed. Effectiveness of business analytics systems, no matter the level in the analytics hierarchy, depends largely on the quality and quantity of the data (volume and representational richness); the accuracy, integrity, and timeliness of the data management system; and the capabilities and sophistication of the analytical tools and procedures used in the process. Understanding the analytics taxonomy helps organizations to be smart about selecting and implementing analytics capabilities to efficiently navigate the maturity continuum.

The Cutting Edge of Analytics: IBM Watson

IBM Watson and its successor, Summit, are perhaps the smartest computer systems built to date. Since the emergence of computers and, subsequently, artificial intelligence in the late 1940s, scientists have compared the performance of these "smart" machines with human minds. Accordingly, in the mid- to late 1990s, IBM researchers built a smart machine and used the game of chess (generally credited as the game of smart humans) to test their ability against the best of human players. On May 11, 1997, an IBM computer called Deep Blue beat the world chess grandmaster after a six-game match series: two wins for Deep Blue, one for the champion, and three draws. The match lasted several days and received massive media coverage around the world. It was the classic plot line of human versus machine. Beyond the chess contest, the intention of developing this kind of computer intelligence was to make computers able to handle the kinds of complex calculations needed to help discover new medical drugs, do the broad financial modeling needed to identify trends and do risk analysis, handle large database searches, and perform massive calculations needed in advanced fields of science.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

After a couple of decades, IBM researchers came up with another idea that was perhaps more challenging: a machine that could not only play the American TV quiz show *Jeopardy!* but beat the best of the best. *Jeopardy!* is much more challenging than chess. While chess is well structured and has very simple rules, and therefore is a very good match for computer processing, *Jeopardy!* is neither simple nor structured. *Jeopardy!* is a game designed for human intelligence and creativity, and therefore a computer designed to play it needed to be a cognitive computing system that could work and think like a human. Making sense of imprecision inherent in human language was the key to success.

In 2010 an IBM research team developed Watson, an extraordinary computer system—a novel combination of advanced hardware and software—designed to answer questions posed in natural human language. The team built Watson as part of the DeepQA project and named it after IBM's first president, Thomas J. Watson. The team that built Watson was looking for a major research challenge: one that could rival the scientific and popular interest of Deep Blue and would also have clear relevance to IBM's business interests. The goal was to advance computational science by exploring new ways for computer technology to affect science, business, and society at large. Accordingly, IBM Research undertook a challenge to build Watson as a computer system that could compete at the human champion level in real time on *Jeopardy!* The team wanted to create a real-time automatic contestant on the show, capable of listening, understanding, and responding—not merely a laboratory exercise.

Competing Against the Best at Jeopardy!

In 2011, as a test of its abilities, Watson competed on the quiz show *Jeopardy!* in the first-ever human-versus-machine matchup for the show. In a two-game, combined-point match (broadcast in three

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Jeopardy! episodes during February 14–16), Watson beat Brad Rutter, the biggest all-time money winner on *Jeopardy!*, and Ken Jennings, the record holder for the longest championship streak (75 days). In these episodes, Watson consistently outperformed its human opponents on the game's signaling device, but it had trouble responding to a few categories, notably those having short clues containing only a few words. Watson had access to 200 million pages of structured and unstructured content, consuming 4 terabytes of disk storage. During the game, Watson was not connected to the Internet.

Meeting the *Jeopardy!* challenge required advancing and incorporating a variety of text mining and natural language processing technologies, including parsing, question classification, question decomposition, automatic source acquisition and evaluation, entity and relationship detection, logical form generation, and knowledge representation and reasoning. Winning at *Jeopardy!* requires accurately computing confidence in answers. The questions and content are ambiguous and noisy, and none of the individual algorithms are perfect. Therefore, each component must produce a confidence in its output, and individual component confidences must be combined to compute the overall confidence of the final answer. The final confidence is used to determine whether the computer system should risk choosing to answer at all. In *Jeopardy!* parlance, this confidence is used to determine whether the computer will "ring in" or "buzz in" for a question. The confidence must be computed during the time the question is read and before the opportunity to buzz in. This is roughly between one and six seconds, with an average around three seconds.

How Does Watson Do It?

The system behind Watson, which is called DeepQA, is a massively parallel, text mining–focused, probabilistic evidence-based

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

computational architecture. For the *Jeopardy!* challenge, Watson used more than 100 different techniques for analyzing natural language, identifying sources, finding and generating hypotheses, finding and scoring evidence, and merging and ranking hypotheses. What is far more important than any particular technique the IBM team used is how it combined them in DeepQA such that overlapping approaches could bring their strengths to bear and contribute to improvements in accuracy, confidence, and speed.

DeepQA is an architecture with an accompanying methodology that is not specific to the *Jeopardy!* challenge. These are the overarching principles in DeepQA:

- **Massive parallelism.** Watson needed to exploit massive parallelism in the consideration of multiple interpretations and hypotheses.
- **Many experts.** Watson needed to be able to integrate, apply, and contextually evaluate a wide range of loosely coupled probabilistic question and content analytics.
- **Pervasive confidence estimation.** No component of Watson commits to an answer; all components produce features and associated confidences, scoring different question and content interpretations. An underlying confidence-processing substrate learns how to stack and combine the scores.
- **Integration of shallow and deep knowledge.** Watson needed to balance the use of strict semantics and shallow semantics, leveraging many loosely formed ontologies.

Figure 1.5 illustrates the DeepQA architecture at a very high level. More technical details about the various architectural components and their specific roles and capabilities can be found in Ferrucci et al. (2010).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

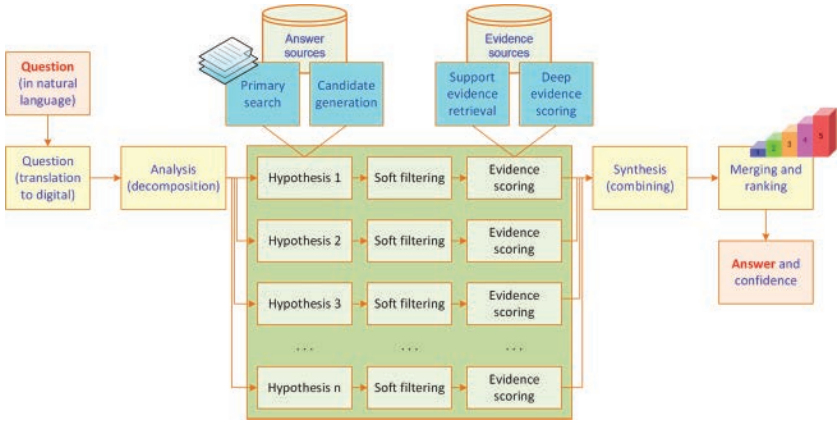


Figure 1.5 A High-Level Depiction of DeepQA Architecture

What Is the Future for Watson?

The *Jeopardy!* challenge helped IBM address requirements that led to the design of the DeepQA architecture and the implementation of Watson. After three years of intense research and development by a core team of about 20 researchers, as well as a significant R&D budget, Watson managed to perform at human expert levels in terms of precision, confidence, and speed at the *Jeopardy!* quiz show.

After the show, the big question was "So what now?" Was developing Watson all for a quiz show? Absolutely not! Showing the rest of the world what Watson (and the cognitive system behind it) could do became an inspiration for the next generation of intelligent information systems. For IBM, it was a demonstration of what is possible with cutting-edge analytics and computational sciences. The message is clear: If a smart machine can beat the best of the best in humans at what they are the best at, think about what it can do for your organizational problems. The first industry that utilized Watson was health care, followed by security, finance, retail, education, public services, and research. The following sections provide short descriptions of what Watson can do (and, in many cases, is doing) for these industries.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Health Care

The challenges that health care is facing today are rather big and multifaceted. With the aging U.S. population, which may be partially attributed to better living conditions and advanced medical discoveries fueled by a variety of technological innovations, demand for health care services is increasing faster than the supply of resources. As we all know, when there is an imbalance between demand and supply, the prices go up and the quality suffers. Therefore, we need cognitive systems like Watson to help decision makers optimize the use of their resources in both clinical and managerial settings.

According to health care experts, only 20% of the knowledge physicians use to diagnose and treat patients is evidence based. Considering that the amount of medical information available is doubling every five years and that much of this data is unstructured, physicians simply don't have time to read every journal that can help them keep up-to-date with the latest advances. Given the growing demand for services and the complexity of medical decision making, how can health care providers address these problems? The answer could be to use Watson, or some other cognitive system like Watson that has the ability to help physicians in diagnosing and treating patients by analyzing large amounts of data—both structured data coming from electronic medical record databases and unstructured text coming from physician notes and published literature—to provide evidence for faster and better decision making. First, the physician and the patient can describe symptoms and other related factors to the system in natural language. Watson can then identify the key pieces of information and mine the patient's data to find relevant facts about family history, current medications, and other existing conditions. It can then combine that information with current findings from tests, and then it can form and test hypotheses for potential diagnoses by examining a variety of data sources—treatment guidelines, electronic medical record data and doctors' and nurses' notes, and peer-reviewed research and clinical studies. Next, Watson can suggest potential diagnostics and treatment options, with a confidence rating for each suggestion.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Watson also has the potential to transform health care by intelligently synthesizing fragmented research findings published in a variety of outlets. It can dramatically change the way medical students learn. It can help healthcare managers to be proactive about the upcoming demand patterns, optimally allocate resources, and improve processing of payments. Early examples of leading health care providers using Watson-like cognitive systems include MD Anderson, Cleveland Clinic, and Memorial Sloan Kettering.

Security

As the Internet expands into every facet of our lives—ecommerce, ebusiness, smart grids for energy, smart homes for remote control of residential gadgets and appliances—to make things easier to manage, it also opens up the potential for ill-intended people to intrude in our lives. We need smart systems like Watson that are capable of constantly monitoring for abnormal behavior and, when it is identified, preventing people from accessing our lives and harming us. This could be at the corporate or even national security system level; it could also be at the personal level. Such a smart system could learn who we are and become a digital guardian that could make inferences about activities related to our life and alert us whenever abnormal things happen.

Finance

The financial services industry faces complex challenges. Regulatory measures, as well as social and governmental pressures for financial institutions to be more inclusive, have increased. And the customers the industry serves are more empowered, demanding, and sophisticated than ever before. With so much financial information generated each day, it is difficult to properly harness the right information to act on. Perhaps the solution is to create smarter client engagement by better understanding risk profiles and the operating

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

environment. Major financial institutions are already working with Watson to infuse intelligence into their business processes. Watson is tackling data-intensive challenges across the financial services sector, including banking, financial planning, and investing.

Retail

Retail industry is rapidly changing with the changing needs and wants of customers. Customers, empowered by mobile devices and social networks that give them easier access to more information faster than ever before, have high expectations for products and services. While retailers are using analytics to keep up with those expectations, their bigger challenge is efficiently and effectively analyzing the growing mountain of real-time insights that could give them a competitive advantage. Watson's cognitive computing capabilities related to analyzing massive amounts of unstructured data can help retailers reinvent their decision-making processes around pricing, purchasing, distribution, and staffing. Because of Watson's ability to understand and answer questions in natural language, it is an effective and scalable solution for analyzing and responding to social sentiment based on data obtained from social interactions, blogs, and customer reviews.

Education

With the rapidly changing characteristics of students—more and more visually oriented/stimulated, constantly connected to social media and social networks, increasingly shorter attention spans—what should the future of education and the classroom look like? The next generation of educational system should be tailored to fit the needs of the new generation, with customized learning plans, personalized textbooks (digital ones with integrated multimedia—audio, video, animated graphs/charts, etc.), dynamically adjusted curriculum, and perhaps smart digital tutors and 24/7 personal advisors. Watson seems

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

to have what it takes to make all this happen. With its natural language processing capability, students can converse with it just as they do with their teachers, advisors, and friends. A smart assistant can answer students' questions, satisfy their curiosity, and help them keep up with the endeavors of the educational journey.

Government

For local, regional, and national governments, the exponential rise of big data presents an enormous dilemma. Today's citizens are more informed and empowered than ever before, and this means they have high expectations for the value of the public sector serving them. And government organizations can now gather enormous volumes of unstructured, unverified data that could serve their citizens—but only if that data can be analyzed efficiently and effectively. IBM Watson's cognitive computing may help make sense of this data deluge, speeding governments' decision-making processes and helping public employees focus on innovation and discovery.

Research

Every year, hundreds of billions of dollars are spent on research and development, most of it documented in patents and publications, creating enormous amounts of unstructured data. To contribute to the extant knowledge base, it is necessary to sift through these data sources to find the outer boundaries of research in a particular field. This is very difficult, if not impossible work, if it is done with traditional means, but Watson can act as a research assistant to help collect and synthesize information to keep people updated on recent findings and insights. For instance, the New York Genome Center is using the IBM Watson cognitive computing system to analyze the genomic data of patients diagnosed with a highly aggressive and malignant brain cancer and to more rapidly deliver personalized, life-saving treatment to patients with this disease (Royyuru, 2014).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Summary

Several popular terms—including business analytics, data science, data mining, business intelligence, and data analytics—are floating around business and scientific circles today. Although they differ from each other slightly in terms of their coverage and application domains, they all have the same purpose, which is to convert data into actionable insight. They aim to solve seemingly unsolvable problems using feature-rich data. A synergistic use of traditional statistics and modern-day machine learning makes this process of knowledge discovery a reality in many industries and scientific fields.

The increased popularity of business analytics and its derivative terms and concepts can be attributed to four main factors: need (to make faster and better decisions in today's highly competitive business world), availability (of data, highly capable software, and increasingly efficient algorithms), affordability (continuously decreasing the cost of ownership via analytics-as-a-service business models and cloud computing), and cultural change (focused on enterprise-wide data-/evidence-based decision-making practices).

This chapter provides an overview of business analytics using a simple three-echelon taxonomy and describes each of these echelons along with their interrelationships with one another. The rest of this book describes the enabling methods and methodologies of predictive analytics, data mining, text mining, and machine learning using a simple, motivational, nontechnical, and example-focused explanation approach.

References

Bi, R. (2014). *When Watson Meets Machine Learning*. www.kdnuggets.com/2014/07/watson-meets-machine-learning.html (accessed June 2014).

DeepQA. (2011). *DeepQA Project: FAQ, IBM Corporation*. www.research.ibm.com/deepqa/faq.shtml (accessed September 2020).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Feldman, S., J. Hanover, C. Burghard, & D. Schubmehl. (2012). *Unlocking the Power of Unstructured Data*. ftp://public.dhe.ibm.com/software/data/sw-library/ecm-programs/IDC_UnlockingThePower.pdf (accessed September 2020).

Ferrucci, D., et al. (2010). "Building Watson: An Overview of the DeepQA Project," *AI Magazine*, 31(3): 59–79.

IBM. (2020). *Build with Watson: The AI for business*. https://www.ibm.com/watson/developer (accessed October 2020).

Liberatore, M., & W. Luo. (2011). "INFORMS and the Analytics Movement: The View of the Membership," *Interfaces*, 41(6): 578–589.

Robinson, A., J. Levis, & G. Bennett. (2010, October). "Informs to Officially Join Analytics Movement," *OR/MS Today*.

Royyuru, A. (2014). *IBM's Watson Takes on Brain Cancer: Analyzing Genomes to Accelerate and Help Clinicians Personalize Treatments*. www.medicaldesignandoutsourcing.com/ibms-watson-takes-on-brain-cancer-2/ (accessed September 2020).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

2

Introduction to Predictive Analytics and Data Mining

Data mining is one of the most important enablers of data analytics in general and predictive analytics in specific. Although its roots date back to the late 1980s and early 1990s, the most powerful data mining applications have been developed since the turn of the 21st century. Many people believe that the recent popularity of analytics can largely be credited to the increasing use of data mining, which is the process of extracting and providing much-needed insight and knowledge to decision makers at all levels of the managerial hierarchy. The term *data mining* was originally used to describe the process through which previously unknown patterns in data were discovered. Software vendors and consulting companies have expanded this definition to include most forms of data analysis in order to increase its reach and capability and thereby increase sales of software tools and services related to data mining. With the emergence of *analytics* as an overarching term for all data analyses, data mining is put back into its proper place—on the analytics continuum where the new discovery of knowledge happens.

In an article in *Harvard Business Review*, Thomas Davenport (2006), a well-known and respected expert in the field of business analytics, argued that the latest strategic weapon for today's businesses is analytical decision making that is based on discovery of new knowledge through data mining. He provided examples of companies

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

such as Amazon.com, Capital One, Marriott International, and others that have used (and are still using) analytics to better understand their customers and optimize their extended supply chains in order to maximize their returns on investment while providing the best possible customer service. This level of success can happen only if the company pursues all avenues, including analytics at all three levels—descriptive, predictive, and prescriptive—to intimately understand its customers and their needs and wants, along with their vendors, business processes, and the extended supply chain.

Data mining is the process of converting data into information and then into knowledge. In the context of knowledge management, data mining is the phase in which new knowledge is created. Knowledge is very distinct from data and information (see Figure 2.1). Whereas data is facts, measurements, and statistics, information is organized or processed data that is timely (i.e., inferences from the data are drawn within a particular time frame) and understandable (i.e., with regard to the original data). Knowledge is information that is contextual, relevant, and actionable. For example, a map that gives detailed driving directions from one location to another could be considered data. An up-to-the-minute traffic bulletin along the freeway that indicates a traffic slowdown due to construction several miles ahead could be considered information. Awareness of an alternative, back-roads route could be considered knowledge. In this case, the map is considered data because it does not contain current relevant information that affects the driving time and conditions from one location to the other. However, having the current conditions as information is useful only if you have knowledge that enables you to avoid the construction zone. The implication is that knowledge has strong experiential and reflective elements that distinguish it from information in a given context.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

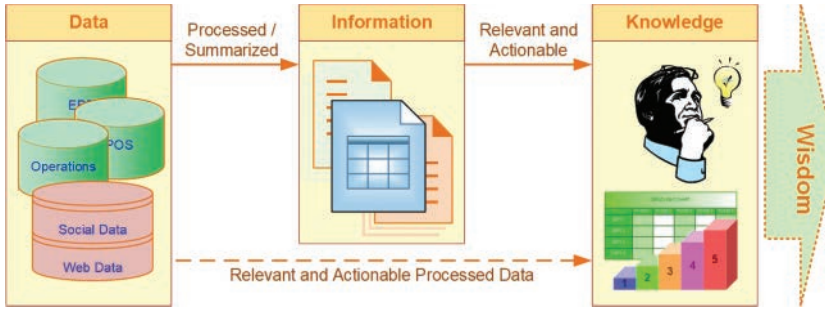


Figure 2.1 The Continuum of Data to Information to Knowledge to Wisdom

Having knowledge implies that it can be exercised to solve a problem, whereas having information does not carry the same connotation. An ability to act is an integral part of being knowledgeable. For example, two people in the same context with the same information may not have the same ability to use the information with the same degree of success. Hence, there is a difference in the human capability to add value. Such differences in ability may be due to different experiences, different training, different perspectives, and other factors. Whereas data, information, and knowledge may all be viewed as assets of an organization, knowledge provides a higher level of meaning about data and information. Because knowledge conveys meaning, it tends to be much more valuable—yet more ephemeral.

Although the term *data mining* is relatively new to many people, the ideas behind it are not. Many of the techniques used in data mining have roots in traditional statistical analysis and artificial intelligence work done since the early part of 1950s. Why, then, has data mining suddenly gained the attention of the business world? Following are some of the most common reasons:

- **More intense competition at the global scale.** There are more suppliers than there is demand to satisfy everybody.
- **Constantly changing needs and wants of the customers.** Because of the increasing number of suppliers and their

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

offerings (e.g., higher quality, lower cost, faster service), customer demand is changing in a dramatic fashion.

- **Recognition of the value of data.** Businesses are now aware of the untapped value hidden in large data sources.
- **Changing culture of management.** Data-driven, evidence-based decision making is becoming a common practice, significantly changing the way managers work.
- **Improved data capture and storage techniques.** Collection and integration of data from a variety of sources into standardized data structures enable businesses to have quality data about customers, vendors, and business transactions.
- **Emergence of data warehousing.** Databases and other data repositories are being consolidated into a single location in the form of a data warehouse to support analytics and managerial decision making.
- **Technological advancements in hardware and software.** Processing and storage capabilities of computing devices are increasing exponentially.
- **Cost of ownership.** While capabilities are increasing, the costs of hardware and software for data storage and processing are rapidly decreasing.
- **Availability of data.** Living in the age of the Internet brings new opportunities to analytically savvy businesses to identify and tap into very large and information-rich data sources (e.g., social networks, social media) to gain better understandings.

Data is everywhere. For instance, data generated by Internet-based activities is increasing rapidly, reaching volumes that we did not even have specific names for in the very recent past. Large amounts of genomic data and related information (in the form of publications and research findings published in journal articles and other media outlets) are being generated and accumulated all over the world. Disciplines

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

such as astronomy and nuclear physics create huge quantities of data on a regular basis. Medical and pharmaceutical researchers constantly generate and store data that can then be used in data mining applications to identify better ways to diagnose and treat illnesses and to discover new and improved drugs. On the commercial side, perhaps the most common use of data and data mining has been in the finance, retail, and health care sectors. Data mining is used to detect and reduce fraudulent activities, especially in insurance claims and credit card use; to identify customer buying patterns; to reclaim profitable customers; to identify trading rules from historical data; and to aid in increased profitability using market-basket analysis. Data mining is already widely used to better target clients, and with the widespread development of ecommerce, this can only become more imperative with time.

What Is Data Mining?

At the most basic level, *data mining* can be defined as the process of discovering (i.e., mining) knowledge (i.e., actionable information) from large amounts of data. When you really think about it, you realize that the term *data mining* is not a proper description of what's really happening; that is, mining of gold from within rocks and dirt is referred to as *gold* mining rather than *rock* or *dirt* mining. Therefore, data mining perhaps should have been named *knowledge mining* or *knowledge discovery*. Despite the mismatch between the term and its meaning, data mining has become the term of choice in the community at large. Several other names—including *knowledge discovery in databases*, *information extraction*, *pattern analysis*, *information harvesting*, and *pattern searching*, among others—have been suggested to replace data mining, but none has gained any significant traction.

Data mining is a process that involves using statistical, mathematical, and artificial intelligence techniques and algorithms to extract and identify useful information and subsequent knowledge (or patterns)

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

from large sets of data. These patterns can be in the form of business rules, affinities, correlations, trends, or predictions. Fayyad et al. (1996) defined data mining as "the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data stored in structured databases," where the data is organized in records structured by categorical, ordinal, and continuous variables. The key terms in this definition have the following meanings:

- *Process* implies that data mining comprises many iterative steps.
- *Nontrivial* means that some experimentation-type search or inference is involved; that is, it is not as straightforward as a computation of predefined quantities.
- *Valid* means that the discovered patterns should hold true on new data with a sufficient degree of certainty.
- *Novel* means that the patterns were not previously known to the user in the context of the system being analyzed.
- *Potentially useful* means that the discovered patterns should lead to some benefit to the user or task.
- *Ultimately understandable* means that the pattern should make business sense that leads to users saying "This makes sense. Why didn't I think of that?"—if not immediately at least after some processing.

Data mining is not a new discipline but rather a new approach in the intersection of several other scientific disciplines. To some extent, data mining is a new philosophy that suggests the use of data and mathematical models to create/discover new knowledge. Data mining leverages capabilities of other disciplines, including statistics, artificial intelligence, machine learning, management science, information systems, and databases, in a systematic and synergistic way (see Figure 2.2). Using collective capabilities of these scientific disciplines, data mining aims to make progress in extracting useful information and knowledge from large data repositories. It is an emerging field that has attracted

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

much attention in a very short time and has fueled the emergence and popularity of the analytics movement.



Figure 2.2 Data Mining: A Multidisciplinary Approach to Knowledge Discovery

What Data Mining Is Not

Because of its appeal, many have used the term *data mining* to refer to any analysis that has something to do with data. For instance, people may refer to casual Internet searching as data mining. It's true that Internet searching is a way to mine large and diverse data/information sources to find (i.e., discover) evidence for a specific

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

question/query—and because of that, it may seem like data mining. However, data mining is the process of discovering reusable patterns through the application of statistical and/or machine learning techniques. Therefore, data mining is much more rigorous and scientific than simply querying the Internet.

Another concept that data mining is often confused with is OLAP (online analytical processing). As the core enabler of the business intelligence movement, OLAP is a collection of database-querying methods for searching very large databases (or data warehouses) through the use of data cubes. Cubes are multidimensional representations of the data stored in data warehouses. With the use of cubes, OLAP helps decision makers to slice and dice organizational data for the purpose of finding answers to "What happened?" "Where did it happen?" and "When did it happen?" types of questions. As sophisticated as it may sound—and perhaps from the standpoint of efficiency, it actually is that sophisticated—OLAP is not data mining. It may be a precursor to data mining. In a sense, they complement each other in that they convert data into information and knowledge for better and faster decision making. Whereas OLAP is part of descriptive analytics, data mining is an essential part of predictive analytics.

There has been much discussion about statistics and data mining; some people suggest that data mining is a part of statistics and others propose that statistics is a part of data mining. Yet others suggest that data mining and statistics are the same thing. Even though we cannot get to the bottom of that discussion here, we can at least shed some light on it by mentioning a few critical points. Data mining and statistics have a lot in common. They both look for relationships in data. The main difference between the two is that statistics starts with a well-defined proposition and hypothesis, whereas data mining starts with a loosely defined discovery statement. Statistics collects a sample of data (i.e., primary data) to test the hypothesis, whereas data mining and analytics use existing data (i.e., often observational, secondary data) to discover novel patterns and relationships. Another difference

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

is in the size of data that they use. Data mining looks for data sets that are as "big" as possible, while statistics looks for right size of data and, if the data is larger than what is needed or required for the statistical analysis, it uses a sample of the data. Statistics and data mining have different definitions of what constitutes "large data": Whereas a few hundred to a thousand data points is large enough for a statistician, several million to a few billion data points is considered large for data mining studies.

In summary, data mining is not a simple Internet search or routine application of OLAP, and it's not the same as statistics. Even though it uses capabilities of these descriptive techniques, data mining is the next level in the analytics hierarchy, where interesting patterns (relationships and future trends) are discovered with the use of data and models.

The Most Common Data Mining Applications

Data mining has become a popular tool in addressing many complex business problems and opportunities. It has proven to be very successful and helpful in many areas, some of which are listed and briefly discussed in the following sections. It is difficult to find an industry or a problem area where a significant number of data mining applications have not already been covered in the literature. The goal of many of these data mining applications is to solve complex problems or to explore emerging opportunities in order to create a sustainable competitive advantage.

Marketing and Customer Relationship Management

Customer relationship management (CRM) is an extension of traditional marketing. The goal of CRM is to create one-on-one relationships with customers by developing an intimate understanding of their

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

needs and wants. As businesses build relationships with their customers over time through a variety of interactions (e.g., product inquiries, sales, service requests, warranty calls, product reviews, social media connections), they accumulate tremendous amounts of data. When combined with demographic and socioeconomic attributes, this information-rich data can be used to identify most likely responders/buyers of new products and services (i.e., customer profiling); understand the root causes of customer attrition in order to improve customer retention (i.e., churn analysis); discover time-variant associations between products and services to maximize sales and customer value; and find the most profitable customers and their preferential needs to strengthen relationships and maximize sales.

Banking and Finance

Data mining can help banks and other financial institutions address a variety of problems and opportunities. Data mining can be used to streamline and automate the processing of loan applications by accurately predicting and identifying the most probable defaulters; to detect fraudulent credit card and online-banking transactions; to find new ways to maximize customer value by selling products and services that customers are most likely to buy; and to optimize cash return by accurately forecasting the cash flow on banking entities (e.g., ATMs, bank branches).

Retailing and Logistics

In retailing, data mining can be used to predict accurate sales volumes at specific retail locations in order to determine correct inventory levels; to identify sales relationships between different products (with market-basket analysis) to improve store layout and optimize sales promotions; to forecast consumption levels for different product types (based on seasonal and environmental conditions); to optimize logistics and hence maximize sales; and to

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

discover interesting patterns in the movement of products (especially for products that have a limited shelf life because they are prone to expiration, perishability, and contamination) in a supply chain by analyzing sensory and RFID data.

Manufacturing

Manufacturers can use data mining to predict machinery failures before they occur through the use of sensory data (enabling *condition-based maintenance*); to identify anomalies and commonalities in production systems to optimize manufacturing capacity; and to discover novel patterns to identify and improve product quality.

Brokerages and Securities Trading

Brokers and traders use data mining to predict when and how much certain stock and/or bond prices will change; to forecast the range and direction of market fluctuations; to assess the effect of particular issues and events on overall market movements; and to identify and prevent fraudulent activities in securities trading.

Insurance

The insurance industry uses data mining techniques to forecast claim amounts for property and medical coverage costs for better business planning; to determine optimal rate plans based on the analysis of claims and customer data; to predict which customers are most likely to buy new policies with special features; and to identify and prevent incorrect claims payments and fraudulent activities.

Computer Hardware and Software

Data mining can be used to predict disk drive failures well before they actually occur, to identify and filter unwanted web content and

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

email messages, to detect and prevent computer network security bridges, and to identify potentially unsecure software products.

Government and Defense

Data mining has a number of government and military applications. It can be used to forecast the cost of moving military personnel and equipment; to predict an adversary's moves and hence develop more successful strategies for military engagements; to predict resource consumption for better planning and budgeting; and to identify classes of unique experiences, strategies, and lessons learned from military operations for better knowledge sharing throughout the organization.

Travel and Lodging

Data mining has a variety of uses in the travel industry. It can be used to predict sales of different services (e.g., seat types in airplanes, room types in hotels/resorts, car types in rental car companies) in order to optimally price services to maximize revenues as a function of time-varying transactions (commonly referred to as *yield management*); to forecast demand at different locations to better allocate limited organizational resources; to identify the most profitable customers and provide them with personalized services to maintain their repeat business; and to retain valuable employees by identifying and acting on the root causes for attrition.

Health and Health Care

Data mining has a number of health care applications. It can be used to help individuals and groups pursue healthier lifestyles (by analyzing data collected with wearable health-monitoring devices); to identify people without health insurance and the factors underlying this undesired phenomenon; to identify novel cost–benefit

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

relationships between different treatments to develop more effective strategies; to forecast the level and the time of demand at different service locations to optimally allocate organizational resources; and to understand the underlying reasons for customer and employee attrition.

Medicine

The use of data mining in medicine is an invaluable complement to traditional medical research, which is mainly clinical and biological in nature. Data mining analyses can be used to identify novel patterns to improve the survivability of patients with cancer; to predict the success rates of organ transplantation patients to develop better donor–organ matching policies; to identify the functions of different genes in the human chromosome (known as *genomics*); and to discover the relationships between symptoms and illnesses (as well as illnesses and successful treatments) to help medical professionals make informed decisions in a timely manner.

Entertainment

Data mining is successfully used in the entertainment industry to analyze viewer data to decide what programs to show during prime time and how to maximize returns by knowing where to insert advertisements; to predict the financial success of movies before they are produced to make investment decisions and to optimize returns; to forecast the demand at different locations and different times to better schedule entertainment events and to optimally allocate resources; and to develop optimal pricing policies to maximize revenues.

Homeland Security and Law Enforcement

Data mining has a number of homeland security and law enforcement applications. Data mining is often used to identify patterns of

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

terrorist behaviors; to discover crime patterns (e.g., locations, timings, criminal behaviors, and other related attributes) to help solve criminal cases in a timely manner; to predict and eliminate potential biological and chemical attacks on the nation's critical infrastructure by analyzing special-purpose sensory data; and to identify and stop malicious attacks on critical information infrastructures (often called *information warfare*).

Sports

Data mining has been used to improve the performance of National Basketball Association (NBA) teams in the United States. Major League Baseball teams use predictive analytics and data mining to optimally utilize limited resources for a winning season. (In fact, *Moneyball* is a popular movie about the use of analytics in baseball.) Most professional sports today employ data crunchers and use data mining to increase their chances of winning.

Data mining applications are not limited to professional sports. For example, Delen at al. (2012) developed models to predict NCAA Bowl Game outcomes, using a wide range of variables about the two opposing teams' previous games statistics. Wright (2012) used a variety of predictors for examination of the NCAA men's basketball championship bracket (a.k.a. March Madness). In short, data mining can be used to predict the outcomes of sporting events, to identify means to increase odds of winning against a specific opponent, and to make the most out of the available resources (e.g., financial, managerial, athletic) so that a team can produce the best possible outcomes.

What Kinds of Patterns Can Data Mining Discover?

Using the most relevant data (which may come from organizational databases or may be obtained from outside sources), data

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

mining builds models to identify patterns among the attributes (i.e., variables or characteristics) that exist in a data set. Models are usually the mathematical representations (simple linear correlations and/or complex highly nonlinear relationships) that identify the relationships among the attributes of the objects (e.g., customers) described in the data set. Some of these patterns are explanatory (explaining the interrelationships and affinities among the attributes), whereas others are predictive (projecting future values of certain attributes). In general, data mining seeks to identify three major types of patterns:

- *Associations* find commonly co-occurring groupings of things, such as "beers and diapers" or "bread and butter" commonly purchased and observed together in a shopping cart (i.e., market-basket analysis). Another type of association pattern captures the sequences of things. These sequential relationships can discover time-ordered events, such as predicting that an existing banking customer who already has a checking account will open a savings account followed by an investment account within a year.
- *Predictions* tell the nature of future occurrences of certain events based on what has happened in the past, such as predicting the winner of the Super Bowl or forecasting the absolute temperature on a particular day.
- *Clusters* identify natural groupings of things based on their known characteristics, such as assigning customers in different segments based on their demographics and past purchase behaviors.

These types of patterns have been manually extracted from data by humans for centuries, but the increasing volume of data in modern times has created a need for more automatic approaches. As data sets have grown in size and complexity, direct manual data analysis has increasingly been augmented with indirect, automatic data processing tools that use sophisticated methodologies, methods, and algorithms.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The manifestation of such evolution of automated and semi-automated means of processing large data sets is now commonly referred to as *data mining*.

As mentioned earlier, generally speaking, data mining tasks and patterns can be classified into three main categories: prediction, association, and clustering. Based on the way in which the patterns are extracted from the historical data, the learning algorithms of data mining methods can be classified as either supervised or unsupervised. With supervised learning algorithms, the training data includes both the descriptive attributes (i.e., independent variables or decision variables) and the class attribute (i.e., output variable or result variable). In contrast, with unsupervised learning, the training data includes only the descriptive attributes. Figure 2.3 shows a simple taxonomy for data mining tasks, along with the learning methods and popular algorithms for each of the data mining tasks. Out of the three main categories of tasks, prediction patterns/models can be classified as the outcome of a supervised learning procedure, while association and clustering patterns/models can be classified as the outcome of unsupervised learning procedures.

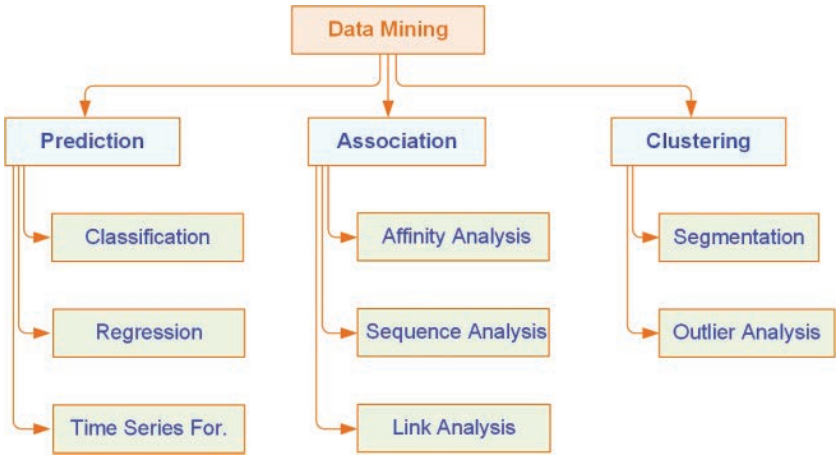


Figure 2.3 A Simple Taxonomy for Data Mining

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Prediction is commonly used to indicate telling about the future. It differs from simple guessing by taking into account the experiences, opinions, and other relevant information in conducting the task of foretelling. A term that is commonly associated with prediction is *forecasting*. Even though many people use these two terms synonymously, there is a subtle difference between them. Whereas prediction is largely experience and opinion based, forecasting is data and model based. That is, in the order of increasing reliability, one might list the relevant terms as guessing, predicting, and forecasting. In data mining terminology, *prediction* and *forecasting* are used synonymously, and the term *prediction* is used as the common representation of the act. Depending on the nature of what is being predicted, prediction can be named more specifically as classification (where the predicted thing, such as tomorrow's forecast, is a class label such as "rainy" or "sunny") or regression (where the predicted thing, such as tomorrow's temperature, is a real number, such as "65 degrees").

Classification, or supervised induction, is perhaps the most common of all data mining tasks. The objective of classification is to analyze the historical data stored in a database and automatically generate a model that can predict future behavior. This induced model consists of generalizations over the records of a training data set, which help distinguish predefined classes. The hope is that the model can then be used to predict the classes of other unclassified records and, more importantly, to accurately predict actual future events.

Common classification tools include neural networks and decision trees (from machine learning), logistic regression and discriminant analysis (from traditional statistics), and emerging tools such as rough sets, support vector machines, and genetic algorithms. Statistics-based classification techniques (e.g., logistic regression, discriminant analysis) have been criticized as making unrealistic assumptions about the data, such as independence and normality, which limit their use in classification-type data mining projects.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Neural networks involve the development of mathematical structures (somewhat resembling the biological neural networks in the human brain) that have the capability to learn from past experiences, presented in the form of well-structured data sets. They tend to be more effective when the number of variables involved is rather large and the relationships among them are complex and imprecise. Neural networks have disadvantages as well as advantages. For example, it is usually very difficult to provide a good rationale for the predictions made by a neural network. Also, neural networks tend to need considerable training. Unfortunately, the time needed for training tends to increase exponentially as the volume of data increases, and, in general, neural networks cannot be trained on very large databases. These and other factors have limited the applicability of neural networks in data-rich domains. (See Chapter 5, "Algorithms for Predictive Analytics," for more detailed coverage of neural networks.)

Decision trees classify data into a finite number of classes, based on the values of the input variables. Decision trees are essentially a hierarchy of if–then statements and are thus significantly faster than neural networks. They are most appropriate for categorical and interval data. Therefore, incorporating continuous variables into a decision tree framework requires discretization—that is, the conversion of continuous valued numeric variables to ranges and categories.

A related category of classification tools is rule induction. Unlike with a decision tree, with rule induction, the if–then statements are induced from the training data directly, and they need not be hierarchical in nature. Other, more recent techniques such as SVM, rough sets, and genetic algorithms are gradually finding their way into the arsenal of classification algorithms and are covered in more detail in Chapter 5 as part of the discussion on data mining algorithms.

Using *associations*—which are commonly called *association rules* in data mining—is a popular and well-researched technique for discovering interesting relationships among variables in large databases. Thanks to automated data-gathering technologies such as use of bar

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

code scanners, the use of association rules for discovering regularities among products in large-scale transactions recorded by point-of-sale systems in supermarkets has become a common knowledge-discovery task in the retail industry. In the context of the retail industry, association rule mining is often called *market-basket analysis*.

Two commonly used derivatives of association rule mining are link analysis and sequence mining. With link analysis, the links among many objects of interest are discovered automatically, such as the link between web pages and referential relationships among groups of academic publication authors. With sequence mining, relationships are examined in terms of their order of occurrence to identify associations over time. Algorithms used in association rule mining include the popular Apriori (where frequent item sets are identified), FP-Growth, OneR, ZeroR, and Eclat algorithms. Chapter 4, "Data and Methods for Predictive Analytics," provides an explanation of Apriori.

Clustering involves partitioning a collection of things (e.g., objects, events, etc., presented in a structured data set) into segments (or natural groupings) whose members share similar characteristics. Unlike in classification, in clustering, the class labels are unknown. As the selected algorithm goes through the data set, identifying the commonalities of things based on their characteristics, the clusters are established. Because the clusters are determined using a heuristic-type algorithm, and because different algorithms may end up with different sets of clusters for the same data set, before the results of clustering techniques are put into use, it may be necessary for an expert to interpret and potentially modify the suggested clusters. After reasonable clusters have been identified, they can be used to classify and interpret new data.

Not surprisingly, clustering techniques include optimization. The goal of clustering is to create groups so that the members within each group have maximum similarity and the members across groups have minimum similarity. The most commonly used clustering techniques include *k*-means (from statistics) and self-organizing maps (from

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

machine learning), which is a unique neural network architecture developed by Kohonen (1982).

Firms often effectively use their data mining systems to perform market segmentation with cluster analysis. Cluster analysis is a means of identifying classes of items so that items in a cluster have more in common with each other than with items in other clusters. This type of analysis can be used in segmenting customers and directing appropriate marketing products to the segments at the right time in the right format at the right price. Cluster analysis is also used to identify natural groupings of events or objects so that a common set of characteristics of these groups can be identified to describe them.

Two techniques often associated with data mining are visualization and time-series forecasting. Visualization can be used in conjunction with other data mining techniques to gain a clearer understanding of underlying relationships. As the importance of visualization has increased in recent years, the term *visual analytics* has emerged. The idea is to combine analytics and visualization in a single environment for easier and faster knowledge creation. Visual analytics is covered in detail in Chapter 4. In time-series forecasting, the data consists of values of the same variable that is captured and stored over time, at regular intervals. This data is then used to develop forecasting models to extrapolate the future values of the same variable.

Popular Data Mining Tools

A large number of software vendors provide powerful data mining tools. Some of these vendors are providers of only data mining and statistical analysis software, while others are large firms that provide wide ranges of software and hardware, along with consulting, in addition to data mining software products. Examples of the vendors that provide data mining tools include IBM (IBM SPSS Modeler, formerly known as SPSS PASW Modeler and Clementine), SAS

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

(Enterprise Miner), StatSoft (Statistica Data Miner—now a TIBCO company), KXEN (Infinite Insight—now a SAP company), Salford (CART, MARS, TreeNet, and RandomForest), Angoss (KnowledgeSTUDIO and KnowledgeSeeker), and Megaputer (PolyAnalyst). Noticeably but not surprisingly, the most popular data mining tools were originally developed by the well-established statistical software companies (SPSS, SAS, and StatSoft). This is largely because statistics is the foundation of data mining, and these companies have the means to cost-effectively develop them into full-scale data mining systems.

Most of the business intelligence (BI) tool vendors (e.g., IBM Cognos, Oracle Hyperion, SAP Business Objects, Microstrategy, Teradata, Microsoft) also have some level of data mining capabilities integrated into their software offerings. These BI tools are still primarily focused on descriptive analytics in the sense of multidimensional modeling and data visualization and are not considered to be direct competitors of the data mining tool vendors.

In addition to the commercial data mining tools, there are several open source and/or free data mining software tools available over the Internet. Historically, the most popular free (and open source) data mining tool is Weka, which was is developed by several researchers from the University of Waikato in New Zealand (and can be downloaded from cs.waikato.ac.nz/ml/weka/). Weka includes a large number of algorithms for different data mining tasks and has an intuitive user interface. Another quickly popularized free (for noncommercial use) data mining tool is RapidMiner, developed by RapidMiner.com (which can be downloaded from rapidminer.com). Its graphically enhanced user interface, use of a rather large number of algorithms, and incorporation of a variety of data visualization features set it apart from the rest of the other free data mining tools.

Another free and open source data mining tool with an appealing workflow-type graphical user interface is KNIME Analytics Platform (which can be downloaded from knime.org). A detailed description of KNIME can be found in Appendix A.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The main difference between commercial tools, such as Enterprise Miner, IBM SPSS Modeler, and Statistica, and free tools, such as Weka, RapidMiner, and KNIME, is often the computational efficiency. The same data mining task involving a large data set may take a lot longer to complete with free software, and for some algorithms, it may not even complete (i.e., it may crash due to inefficient use of computer memory). With the cloud-based analytics, this deficiency of open source tools is no longer as prominent as it used to be. For instance, an analytics model can be developed with a small data sample in KNIME Analytics Platform and then deployed and executed on the cloud platform with the complete/large dataset. In addition to software tools, code-based analytics tools and high-level programming languages (i.e., Python, R, JavaScript) are also gaining tremendous popularity in the world of analytics and data science. Table 2.1 lists the major data mining software products and their websites.

Table 2.1 Popular Data Mining Software Tools

Product	Website (URL)
KNIME Analytics Platform	knime.org
SAS Enterprise Miner	https://www.sas.com/en_us/software/enterprise-miner.html
IBM SPSS Modeler	ibm.com/products/spss-modeler
TIBCO Statistica	docs.tibco.com/products/tibco-statistica
RapidMiner	rapidminer.com
PolyAnalyst	megaputer.com/polyanalyst.php
Salford Predictive Modeler	salford-systems.com
XLMiner	https://www.solver.com/xlminer-platform
DataRobot Enterprise Analytics and AI	https://www.datarobot.com/
Databricks Unified Analytics Platform	https://databricks.com/
Apache Spark Analytics	https://spark.apache.org/
H2O Analytics	https://h2oanalytics.com/
Teradata Warehouse Miner	https://www.teradata.com/
Oracle Data Mining	oracle.com/database/technologies/advanced-analytics/odm.html

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Product	Website (URL)
R for Analytics	https://www.r-project.org/
Python for Analytics	https://www.python.org/
Open Source Analytical API Platform for JavaScript	https://cube.dev/

Microsoft's SQL Server includes a suite of business intelligence capabilities that has become increasingly popular for data mining studies. With SQL Server, data and analytic models are hosted in the same relational database environment, significantly increasing the efficiency of model execution while making model management a considerably easier task. The Microsoft Enterprise Consortium serves as the worldwide source for access to Microsoft's SQL Server software suite for academic purposes (i.e., teaching and research). The consortium was established to enable universities around the world to access enterprise technology without having to maintain the necessary hardware and software on their own campuses. The consortium provides a wide range of business intelligence development tools (e.g., data mining, cube building, business reporting) as well as a number of large, realistic data sets from companies such as Sam's Club, Dillard's, and Tyson Foods. The Microsoft Enterprise Consortium is free of charge and can be used only for academic purposes. The Sam M. Walton College of Business at the University of Arkansas hosts the enterprise system and allows consortium members and their students to access these resources using a simple remote desktop connection. For details about becoming a part of the consortium and easy-to-follow tutorials and examples, see https://walton.uark.edu/enterprise/Microsoft/index.php.

In May 2019, KDnuggets (a well-known web portal for data mining and analytics links and resources) conducted its 20th annual software poll on the following question: "What analytics, data science, machine learning software/tools have you used in the last three years (2017–2019) for a real project?" This poll received huge attention

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

from the analytics and data science community, attracting more than 1,800 unique voters. The poll measures both how widely a data analytics/data science software tool is used and how strongly the vendors advocate for their tool. Here are some of the interesting findings that came out of the poll:

- Many business analytics and data science software users use more than one tool to carry out data analytics projects. According to the poll, the average number of tools used by a person or vendor was 6.1 in 2019 (compared to 3.7 in 2014). This is a clear indication that most data scientists use a combination of tools (commercial, free/open source software, programming languages, and open access algorithms and model libraries as community projects). Using only one tool or language seems to be insufficient to deal with the requirements of the new generation of analytics projects.
- The popularity of free and open-source software tools and programming languages far exceeded that of the commercial tools. More than two-thirds of the most popular tools in the top 40 are either free/open source software with graphical user interfaces or programming languages and libraries of models/algorithms for data analytics. Overall, the most popular tool was Python (with 65% of the votes), as was the case in 2018.
- While the percentage of votes for big data tools (e.g., Apache Spark, Hadoop, Kafka) and technologies decreased, the deep learning tools, technologies, and libraries (e.g., TensorFlow, Keras, PyTorch) gained significant popularity.

Figure 2.4 shows the results of the poll for tools that placed in the top 40, based on the number of unique votes they received. The chart in this figure shows the number of votes for each of these tools.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

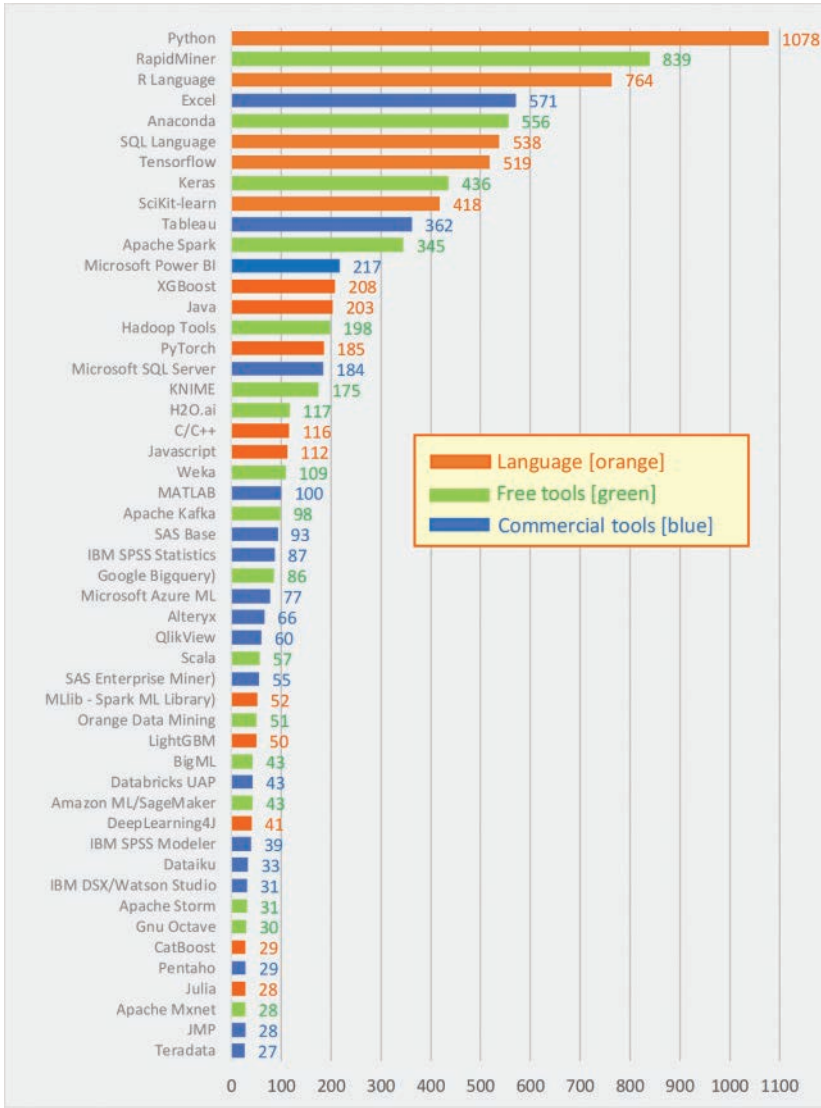


Figure 2.4 Most Popular Data Mining and Analytics Software Tools (User Poll Results)

Source: Used with permission of kdnuggets.com.

To reduce bias through multiple voting, in this poll, KDnuggets used email verification, which may potentially reduce the total number of votes but made the results less biased and more representative.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The Dark Side of Data Mining: Privacy Concerns

Data that is collected, stored, and analyzed in data mining often contains information about real people. Such information may include identification data (e.g., name, address, Social Security number, driver's license number, employee number), demographic data (e.g., age, sex, ethnicity, marital status, number of children), financial data (e.g., salary, gross family income, checking or savings account balance, home ownership, mortgage or loan account specifics, credit card limits and balances, investment account specifics), purchase history (i.e., what is bought from where and when, either from vendors' transaction records or from credit card transaction specifics), and other personal data (e.g., anniversary, pregnancy, illness, loss in the family, bankruptcy filings). Most of these data can be accessed through some third-party data providers. The main issue here is the privacy of the person to whom the data belongs. In order to maintain the privacy and protection of individuals' rights, data mining professionals have ethical and often legal obligations. One way to ethically handle private data is to de-identify customer records prior to applying data mining applications so that the records cannot be traced to an individual. Many publicly available data sources (e.g., CDC data, SEER data, UNOS data) are already de-identified. Prior to accessing these data sources, those mining data are often asked to agree not to try to identify the individuals behind those figures.

In a number of instances in the recent past, companies have shared their customer data with others without seeking the explicit consent of their customers. For instance, as you might recall, in 2003, JetBlue Airlines provided more than a million passenger records of its customers to Torch Concepts, a U.S. government contractor. Torch subsequently augmented the passenger data with additional information, such as family sizes and Social Security numbers—information purchased from a data broker called Acxiom. The consolidated personal

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

database was intended to be used for a data mining project in order to develop potential terrorist profiles. All this was done without notifying or obtaining consent from passengers. When news of the activities got out, however, dozens of privacy lawsuits were filed against JetBlue, Torch, and Acxiom, and several U.S. senators called for an investigation into the incident (Wald, 2004). Similar, but not as dramatic, privacy-related news came out in the recent past about popular social networking companies, which were allegedly selling customer-specific data to other companies for personalized targeted marketing.

A peculiar story about data mining and privacy concerns made headlines in 2012. In this instance, the company did not even use any private and/or personal data. Legally speaking, there was no violation of any laws. This story, which is about Target, goes as follows. In early 2012, an infamous story broke out about Target's practice of using predictive analytics. The story was about a teenager girl who was being sent advertising flyers and coupons by Target for the kind of things that a new mother-to-be would buy from a store like Target. An angry man went into a Target outside Minneapolis, demanding to talk to a manager: "My daughter got this in the mail!" he said. "She's still in high school, and you're sending her coupons for baby clothes and cribs? Are you trying to encourage her to get pregnant?" The manager didn't have any idea what the man was talking about. He looked at the mailer. Sure enough, it was addressed to the man's daughter and contained advertisements for maternity clothing, nursery furniture, and pictures of smiling infants. The manager apologized and then called a few days later to apologize again. On the phone, though, the father was somewhat abashed. "I had a talk with my daughter," he said. "It turns out there's been some activities in my house I haven't been completely aware of. She's due in August. I owe you an apology."

As it turns out, Target figured out a teen girl was pregnant before her father did! Here is how they did it. Target assigns every customer a guest ID number (tied to the person's credit card, name, or email address) that becomes a placeholder to keep a history of everything

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

the person has bought. Target augments this data with any demographic information that it has collected from the customer or bought from other information sources. By using this type of information, Target had looked at historical buying data for all the women who had signed up for Target baby registries in the past. The company had analyzed the data from all directions, and some useful patterns emerged. For example, lotions and special vitamins were among the products with interesting purchase patterns. Lots of people buy lotion, but Target noticed that women on the baby registry were buying larger quantities of unscented lotion around the beginning of their second trimester. Another analyst noted that sometime in the first 20 weeks, pregnant women loaded up on supplements like calcium, magnesium, and zinc. Many shoppers purchase soap and cotton balls, but when someone suddenly starts buying lots of scent-free soap and extra-big bags of cotton balls, in addition to hand sanitizers and washcloths, it signals that she could be getting close to her delivery date. Target was able to identify about 25 products that, when analyzed together, allowed them to assign each shopper a "pregnancy prediction" score. Target could even estimate her due date to within a small window, so the company could send coupons timed to very specific stages of her pregnancy.

If you looked at this practice from a legality perspective, you would conclude that Target did not use any information that violates customers' privacy rights; rather, it used transactional data that almost every other retail chain is collecting and storing (and perhaps analyzing) about customers. What was disturbing in this scenario was perhaps that pregnancy was the targeted concept. People tend to believe that certain events or concepts—such as terminal disease, divorce, and bankruptcy—should be off-limits or treated extremely cautiously.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Application Example: Data Mining for Hollywood Managers

Predicting box office receipts (i.e., the financial success) for a particular motion picture is an interesting and challenging problem. According to some domain experts, the movie industry is the "land of hunches and wild guesses" due to the difficulty associated with forecasting product demand, and this makes the movie business in Hollywood a risky endeavor. In support of such observations, Jack Valenti (the longtime president and CEO of the Motion Picture Association of America) once said that "no one can tell you how a movie is going to do in the marketplace…not until the film opens in a darkened theatre and sparks fly up between the screen and the audience." Entertainment industry trade journals and magazines have been full of examples, statements, and experiences that support this claim.

Like many other researchers who have attempted to shed light on this challenging real-world problem, Ramesh Sharda and Dursun Delen have been exploring the use of data mining to predict the financial performance of a motion picture at the box office before it even enters production (while the movie is nothing more than a conceptual idea). In their highly publicized prediction models, they convert the forecasting (or regression) problem into a classification problem; that is, rather than forecast the point estimate of box office receipts, they classify a movie based on its box office receipts in one of nine categories, ranging from "flop" to "blockbuster," making the problem a multinomial classification problem. Table 2.2 illustrates the definitions of the nine classes in terms of the range of box office receipts.

Table 2.2 Movie Classification, Based on Receipts

Class No.	1	2	3	4	5	6	7	8	9
Range (in millions)	< $1 (Flop)	> $1 < $10	> $10 < $20	> $20 < $40	> $40 < $65	> $65 < $100	> $100 < $150	> $150 < $200	> $200 (Blockbuster)

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Data

For the movie classification model, data was collected from a variety of movie-related databases (e.g., ShowBiz, IMDb, IMSDb, All-Movie) and consolidated into a single data set. The data set for the most recently developed models contained 2,632 movies released between 1998 and 2006. A summary of the independent variables along with their specifications is provided in Table 2.3. For more descriptive details and justification for inclusion of these independent variables, see Sharda and Delen (2006).

Table 2.3 Summary of the Predictive Variables

Independent Variable Name	Definition	Range of Possible Values
MPAA rating	Indicates the rating assigned by the Motion Picture Association of America (MPAA).	G, PG, PG-13, R, NR
Competition	Indicates the level at which each movie competes for the same pool of entertainment dollars against movies released at the same time.	High, Medium, Low
Star value	Signifies the presence of any box office superstars in the cast. A superstar can be defined as an actor who contributes significantly to the up-front sale of the movie.	High, Medium, Low
Genre	Specifies the content category the movie belongs to. Unlike with the other categorical variables, a movie can be classified in more than one content category at the same time (e.g., action as well as comedy). Therefore, each content category is represented with a separate binary variable.	Sci-Fi, Epic Drama, Modern Drama, Thriller, Horror, Comedy, Cartoon, Action, Documentary
Special effects	Signifies the level of technical content and special effects (e.g., animations, sound, visual effects) used in the movie.	High, Medium, Low

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Independent Variable Name	Definition	Range of Possible Values
Sequel	Specifies whether a movie is a sequel (value of Yes or 1) or not (value of No or 0).	Yes, No
Number of screens	Indicates the number of screens on which the movie is planned to be shown during its initial launch.	A positive integer

Methodology

Using a variety of data mining methods, including neural networks, decision trees, support vector machines, and three types of ensembles, Sharda and Delen developed the prediction models. The data from 1998 to 2005 was used as training data to build the prediction models, and the data from 2006 was used as the test data to assess and compare the models' prediction accuracy. Figure 2.5 shows the conceptual architecture for the proposed movie forecasting system, and Figure 2.6 shows the workflow of the model-building process.

Results

Table 2.4 shows the prediction results of all three data mining methods, as well as the results of the three different ensembles. The first performance measure is the percentage correct classification rate, which is called bingo. The table also reports the 1-away correct classification rate (i.e., within one category). The results indicate that SVM performed the best among the individual prediction models, followed by ANN; the worst of the three was the CART decision tree algorithm. In general, the ensemble models performed better than the individual predictions models, and of those, the fusion algorithm performed the best. What is probably more important to decision makers, and stands out in Table 2.4, is the significantly low standard deviation obtained from the ensembles compared to the individual models.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

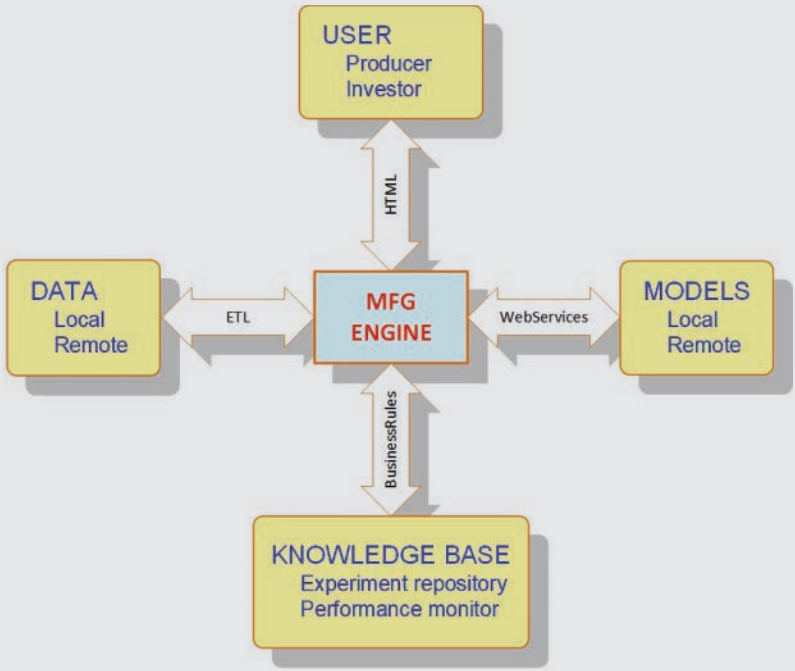


Figure 2.5 Conceptual Architecture for the Movie Prediction System

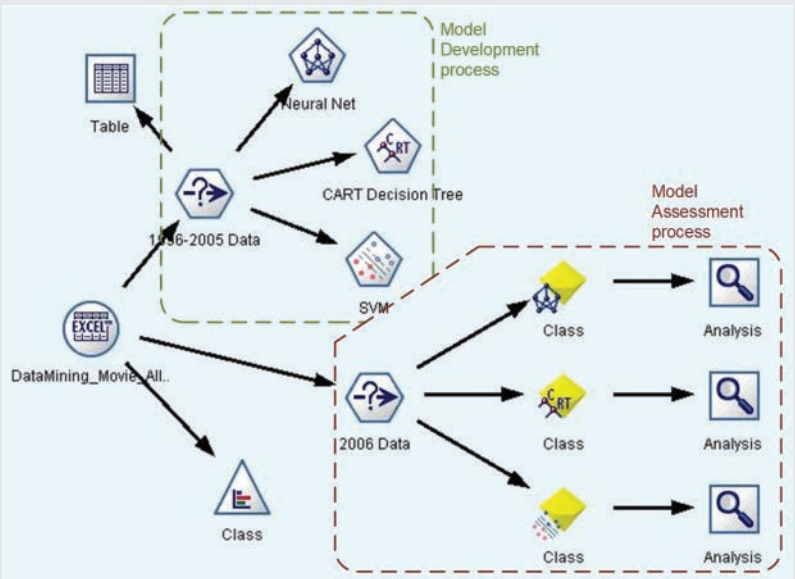


Figure 2.6 Workflow of the Model-Building Process

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Table 2.4 Tabulated Prediction Results for Individual and Ensemble Models

	Prediction Models					
	Individual Models			**Ensemble Models**		
Performance Measure	SVM	ANN	C&RT	Random Forest	Boosted Trees	Information Fusion
Count (bingo)	192	182	140	189	187	194
Count (1-away)	104	120	126	121	104	120
Accuracy (% bingo)	55.49%	52.60%	40.46%	54.62%	54.05%	56.07%
Accuracy (% 1-away)	85.55%	87.28%	76.88%	89.60%	84.10%	90.75%
Standard deviation	0.93	0.87	1.05	0.76	0.84	0.63

Conclusions

The researchers claim that these prediction results are better than any others reported in the published literature for this problem domain. Beyond the attractive accuracy of their prediction results of the box office receipts, these models could also be used to further analyze (and potentially optimize) the decision variables in order to maximize the financial return. Specifically, the parameters used for modeling could be altered using the already trained prediction models in order to better understand the impact of different parameters on the end results. During this process, which is commonly referred to as *sensitivity analysis*, the decision maker of a given entertainment firm could find out, with a fairly high accuracy level, how much value a specific actor (or a specific release date, or the addition of more technical effects, etc.) brings to the financial success of a film, making the underlying system an invaluable decision aid. Figure 2.7 shows a screenshot from the prototype implementation of the prediction system named Movie Forecast Guru.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.



Figure 2.7 A Screenshot of the Movie Forecast Guru

Summary

This chapter is dedicated to the definition and description of predictive analytics and its most prevalent enabler, data mining. As shown in the illustrations of the simple taxonomy for business analytics in Chapter 1, predictive analytics is positioned between descriptive and prescriptive analytics, with the aim of answering the question "What will happen?" Whereas descriptive analytics takes a retrospective view, predictive analytics takes a prospective approach to foresee what is most likely to happen in the future so that the incipient problems can be identified and prevented before they happen and potential opportunities can be recognized and leveraged before they become apparent to everyone (especially the competition). This chapter presents

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

these seemingly complex concepts using several illustrative examples and application cases.

References

Chapman, P., J. Clinton, R. Kerber, T. Khabaza, et al. (2013). CRISP-DM 1.0. www.the-modeling-agency.com/crisp-dm.pdf (accessed July 2020).

Davenport, T. H. (2006, January). "Competing on Analytics," *Harvard Business Review*. https://hbr.org/2006/01/competing-on-analytics (accessed September 2020).

Delen, D. (2019). *Prescriptive Analytics: The Final Frontier for Evidence-Based Management and Optimal Decision Making*. FT Press/Pearson.

Delen, D. (2009). "Analysis of Cancer Data: A Data Mining Approach," *Expert Systems*, 26(1): 100–112.

Delen, D., D. Cogdell, & N. Kasap. (2012). "A Comparative Analysis of Data Mining Methods in Predicting NCAA Bowl Outcomes," *International Journal of Forecasting*, 28: 543–552.

Delen, D., R. Sharda, & P. Kumar. (2007). "Movie Forecast Guru: A Web-Based DSS for Hollywood Managers," *Decision Support Systems*, 43(4): 1151–1170.

Delen, D., G. Walker, & A. Kadam. (2005). "Predicting Breast Cancer Survivability: A Comparison of Three Data Mining Methods," *Artificial Intelligence in Medicine*, 34(2): 113–127.

Fayyad, U., G. Piatetsky-Shapiro, & P. Smyth. (1996). "From Knowledge Discovery in Databases," *AI Magazine*, 17(3): 37–54.

Hill, K. (2012, February 16). "How Target Figured Out a Teen Girl Was Pregnant Before Her Father Did," *Forbes Magazine*. https://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did (accessed September 2020).

Kohonen, T. (1982). "Self-Organized Formation of Topologically Correct Feature Maps," *Biological Cybernetics*, 43(1): 59–69.

Nemati, H. R., & C. D. Barko. (2001). "Issues in Organizational Data Mining: A Survey of Current Practices," *Journal of Data Warehousing*, 6(1): 25–36.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Nolan, R. (2012, February 21). "Behind the Cover Story: How Much Does Target Know?" *The New York Times*.

SAS. (2020). *Introduction to SEMMA*. https://documentation.sas.com/?docsetId=emref&docsetTarget=n061bzurmej4j3n1jnj8bbjjm1a2.htm&docsetVersion=14.3&locale=en (accessed July 2020).

Sharda, R., & D. Delen. (2006). "Predicting Box-Office Success of Motion Pictures with Neural Networks," *Expert Systems with Applications*, 30: 243–254.

Wald, M. (2004). "U.S. Calls Release of JetBlue Data Improper," *The New York Times*, www.nytimes.com/2004/02/21/business/us-calls-release-of-jetblue-data-improper.html (accessed May 2020).

Wright, C. (2012). *Statistical Predictors of March Madness: An Examination of the NCAA Men's Basketball Championship*. http://economics-files.pomona.edu/GarySmith/Econ190/Wright March Madness Final Paper.pdf (accessed July 2020).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

3

Standardized Processes for Predictive Analytics

As has been the case in many other computational paradigms, extracting knowledge from large data repositories (e.g., data mining, predictive analytics, data science, business analytics) started as a trial-and-error experimental process. Many practitioners have approached the problem from the perspective of trying to characterize what works and what doesn't. For quite some time, data mining (or, more recently, data analytics and data science) projects were carried out as rather artistic, ad hoc, experimental endeavors. However, in order to methodically conduct these analytics, a standardized process needed to be developed and followed. Based on best practices, in the early days, data mining researchers in academia and industry proposed several processes—workflows in the form of simple step-by-step methodologies—to maximize the likelihood of success in conducting data analytics projects. These efforts have led to several standardized processes, some of which are described in this chapter.

The Knowledge Discovery in Databases (KDD) Process

One of the earliest and perhaps the first data mining process was called with the name of *knowledge discovery in databases* (*KDD*)

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

methodology, proposed by Fayyad et al. (1996). In the KDD methodology, data mining is portrayed as a single step in which patterns are extracted from preprocessed data. Fayyad et al. proposed KDD as a comprehensive end-to-end process that encompasses many individual steps to convert data into knowledge. Figure 3.1 provides a pictorial representation of the KDD process. In the figure, the processing steps are shown as labeled directional arrows, and the result of each step is shown as a graphical icon representing the artifact. As shown in Figure 3.1, the input to the KDD process is a collection of data coming from organizational databases and/or other external data sources. These data sources are often combined in a centralized data repository called a *data warehouse*, which enables the KDD process to be implemented effectively and efficiently because the data warehouse provides a single source for data to be mined. Once the data is consolidated in a unified data warehouse, with or without the use of a data warehouse, the problem-specific subset of data is extracted and prepared for further processing. Because the data is usually in a raw, incomplete, and dirty state, a through preprocessing and cleaning process needs to be conducted before the modeling can take place. Once the data is preprocessed and transformed into a form that's ready for modeling, a variety of modeling techniques are applied to the data to convert it into patterns, correlations, and predictive models. The discovered patterns must be validated, and then they need to be interpreted and internalized so that they can be transformed into actionable information (i.e., knowledge) for decision makers. One important part of this process is the feedback loop that allows the process flow to redirect backward, from any step to any other previous steps, for rework and readjustments.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

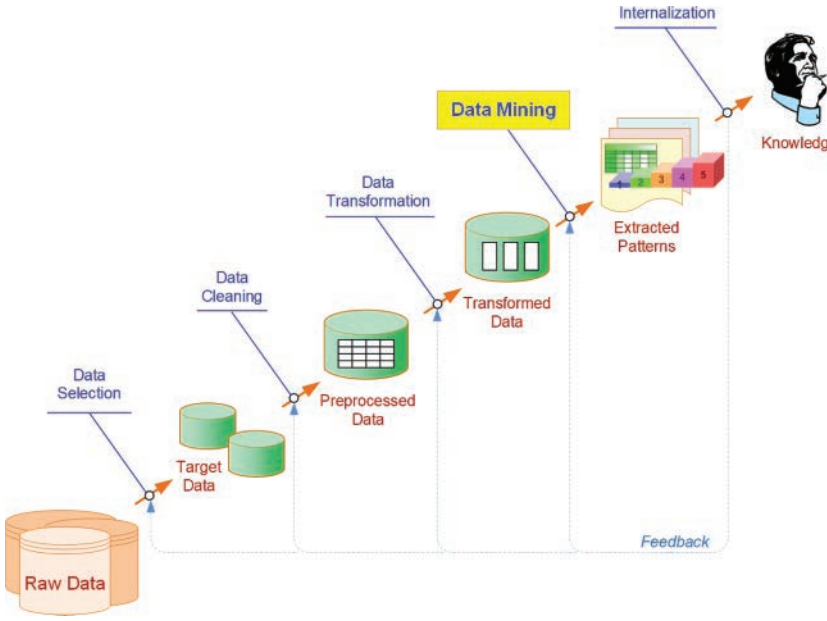


Figure 3.1 The Knowledge Discovery in Databases (KDD) Process

Cross-Industry Standard Process for Data Mining (CRISP-DM)

Another standardized data mining processes—arguably the most popular one—is called Cross-Industry Standard Process for Data Mining (CRISP-DM), and it was proposed in the mid- to late 1990s by a European consortium of companies. It was intended to serve as a nonproprietary standard methodology for data mining projects (Chapman et al., 2013). Today, the CRISP-DM process is used for business analytics and data science projects, and it is sometimes called CRISP-BA (for business analytics), CRISP-DS (for data science), or just CRISP (which makes it a more generalized term by omitting any analytics-related specification).

Figure 3.2 illustrates this six-step standardized CRISP-DM process, which starts with a good understanding of the business problem and the need for the data mining project (i.e., the application

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

domain) and ends with the deployment of the solution that satisfies the specific business need that started the data mining project in the first place. Even though these steps are shown in sequential order in the graphical representation, there is usually a great deal of back-tracking. A significant portion of the data analytics/data science relies on experience and best practices–driven experimentation; that is, there is no one magical way that works for every project and every data scientist. Therefore, depending on the problem situation and the skills, knowledge, and experience of the analyst, the process can be rather iterative (i.e., it may require going back and forth through the steps quite a few times) and time-intensive. Because each step is built on the outcome of the immediate predecessor, it's important to pay attention to the earlier steps in order to avoid putting an entire study on an incorrect path from the outset.

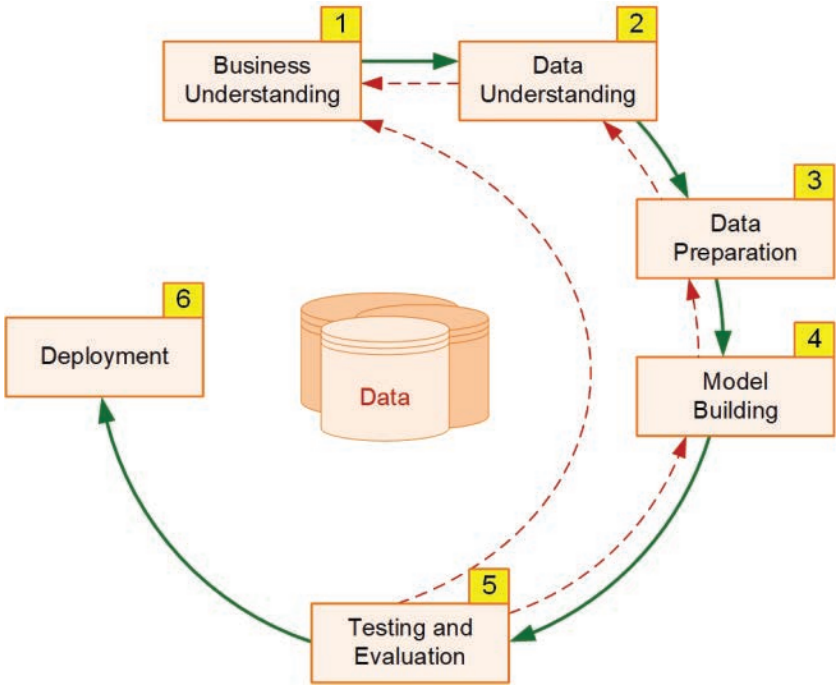


Figure 3.2 The CRISP-DM Standardized Process for Data Analytics/Data Science

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The following sections describe the steps of the CRISP-DM process in detail.

Step 1: Business Understanding

The key to success in any data mining project is to know what the study is for. Making this determination begins with gaining a thorough understanding of the managerial need for new knowledge and an explicit specification of the business objective regarding the study to be conducted. Specific goals such as "What are the common characteristics of the customers we have lost to our competitors recently?" or "What are typical profiles of our customers, and how much value does each of them provide to us?" are needed. Then a project plan for finding such knowledge is developed that specifies the people responsible for collecting the data, analyzing the data, and reporting the findings. At this early stage, a budget to support the study should also be established, at least at a high level and with rough numbers. For example, with a customer segmentation model developed for a retail catalog business, the identification of a business purpose meant identifying the type of customer that would be expected to yield a profitable return. A similar analysis would also be useful for credit card distributors. For business purposes, grocery stores often try to identify which items tend to be purchased together, so they can do better affinity positioning within the store or more intelligently guide promotional campaigns. Data mining has many useful business applications and can be applied to many business problems and opportunities. Intimately knowing the business purpose is critical to achieving success.

Step 2: Data Understanding

The second step in the CRISP-DM process is to make a perfect match between the business problem and the data being used to

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

address it. That is, a data mining study is specific to a well-defined business task, and different business tasks require different sets of data. Therefore, it is important to identify the relevant data from many available data sources. Several key points must be considered in the data identification and selection process. First and foremost, an analyst should be clear and concise in describing the data mining task so that the most relevant data can be identified. For example, a retail data mining project might seek to identify spending behaviors of female shoppers who purchase seasonal clothes based on their demographics, credit card transactions, and socioeconomic attributes. Furthermore, the analyst should gain an intimate understanding of the data sources—for example, where the relevant data is stored and in what form, whether data collection is automated or happens manually, who collects the data, and how often the data are updated. The analyst should also understand variables by seeking answers to questions such as "What are the most relevant variables?" "Are there any synonymous and/or homonymous variables?" and "Are the variables independent of each other—? That is, do they stand as a complete information source without overlapping or conflicting information?"

In order to better understand the data, an analyst often uses a variety of statistical and graphical techniques, such as simple statistical descriptors/summaries of each variable (e.g., for numeric variables, the average, minimum/maximum, median, and standard deviation are among the calculated measures, whereas for categorical variables the mode and frequency tables are calculated), correlation analysis, scatterplots, histograms, and box plots. Careful identification and selection of data sources and the most relevant variables can make it easier for data mining algorithms to quickly discover useful knowledge patterns.

Data sources for data selection can vary. Normally, data sources for business applications include demographic data (e.g., income, education, number of households, age), sociographic data (e.g., hobbies,

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

club memberships, entertainment), and transactional data (e.g., sales record, credit card spending, issued checks), among other types.

Data can be categorized as quantitative and qualitative. Quantitative data is measured using numeric values. It can be discrete (e.g., integers) or continuous (e.g., real numbers). Qualitative data, also known as categorical data, contains both nominal and ordinal data. Nominal data has finite non-ordered values. For example, marital status can potentially have four values: married, single, widowed and divorced. Ordinal data has finite ordered values. For example, customer credit ratings are considered ordinal data because the ratings can be excellent, fair, and bad. Quantitative data can be readily represented by using some sort of probability distribution. A probability distribution describes how the data is dispersed and shaped. For instance, normally distributed data is symmetric and is commonly referred to as being a bell-shaped curve. Qualitative data may be coded to numbers and then described by frequency distributions. Once the relevant data is selected according to the data mining business objective, data preprocessing should be conducted. (For more details on data in data mining, see Chapter 4, "Data and Methods for Predictive Analytics.")

Step 3: Data Preparation

The purpose of data preparation (commonly called data preprocessing) is to prepare the data identified in the previous step for analysis using data mining methods. Of the steps in CRISP-DM, data preprocessing consumes the most time and effort—roughly 80% of the total time spent on a data mining project. This step requires such enormous effort because real-world data is generally incomplete (lacking attribute values, lacking certain attributes of interest, or containing only aggregate data), noisy (containing errors or outliers), and inconsistent (containing discrepancies in codes or names).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Some parts of the data may have different formats because they are taken from different data sources. The selected data may be from flat files, voice message, images, and web pages, and it needs to be converted to a consistent and unified format. In general, data cleaning means filtering, aggregating, and filling in missing values (a.k.a. imputation). When filtering the data, an analyst examines the selected variables for outliers and redundancies. Outliers differ greatly from the majority of data, or data that is clearly out of range of the selected data groups. For example, if the age of a customer included in a data set is 190, this must be a data entry error and should be identified and fixed (perhaps taken out of a data mining project that examines the various aspects of customers, since age is perceived to be a critical customer characteristic). Outliers may occur for many reasons, such as human errors or technical errors, or they may naturally occur in a data set due to extreme events. If the age of a credit card holder is recorded as 12, this is likely a data entry error—most likely made by a human. However, there might actually be an independently wealthy preteen with important purchasing habits. Arbitrarily deleting this outlier could dismiss valuable information.

Data may also be redundant, with the same information recorded in several different ways. Daily sales of a particular product are redundant to seasonal sales of the same product because an analyst can derive the sales from either daily data or seasonal data. Aggregating data reduces data dimensions. Note that although an aggregated data set has a small volume, the information remains. If a marketing promotion for furniture sales is considered in the next three or four years, the available daily sales data can be aggregated as annual sales data. The size of the sales data is then dramatically reduced. Smoothing data helps find missing values of the selected data, and new or reasonable values can be added. These added values could be the average number of the variable (mean) or the mode. A missing value often means no solution is found when a data mining algorithm is applied to discover the knowledge patterns.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Step 4: Model Building

In the fourth step of the CRISP-DM process, various modeling techniques are selected and applied to an already prepared data set in order to address the specific business need. The model building step also encompasses the assessment and comparative analysis of the various types of models that can address the same type of data mining tasks (e.g., clustering, classification). Because there is not a universally known best method or algorithm for a specific data mining task, an analyst should use a variety of viable model types, along with a well-defined experimentation and assessment strategy, to identify the "best" method for a given data mining problem. Even for a single method or algorithm, a number of parameters need to be calibrated to obtain optimal results. Some methods may have specific requirements for the way the data is to be formatted; thus returning to the data preparation step is often necessary.

Depending on the business need, the data mining task can be a prediction (either classification or regression), an association, or a clustering/segmentation type. Each of these data mining tasks can use a variety of data mining methods and algorithms. For instance, classification-type data mining tasks can be accomplished by developing neural networks or by using decision trees, support vector machines, or logistic regression. These data mining methods and their respective algorithms are explained in Chapter 4 and Chapter 5, "Algorithms for Predictive Analytics."

The standard procedure for modeling in data mining is to divide a large preprocessed data set into subsets for training and validation or testing. Then the analyst can use a portion of the data (the training set) to develop the models (no matter what modeling technique and/or algorithm is used) and use the other portion of the data (the test set) for testing the model that was just built. The principle is that if you build a model on a particular set of data, it will of course test quite well on the data on which it was built. By dividing the data and

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

using part of it for model development and testing it on a separate set of data, an analyst can create convincing and reliable results for the accuracy and reliability of the model. The idea of splitting the data into components is often carried to additional levels, with multiple splits in the practice of data mining. For further details about data splitting and other evaluation methods, see Chapter 4.

Step 5: Testing and Evaluation

In the fifth step of the CRISP-DM process, the developed models are assessed and evaluated for accuracy and generality. This step assesses the degree to which the selected model (or models) meets the business objectives and whether more models need to be developed and assessed. Another option is to test the developed model(s) in a real-world scenario if time and budget constraints permit. Even though the outcome of the developed models is expected to relate to the original business objectives, other findings that are not necessarily related to the original business objectives but that might also unveil additional information or hints for future directions are often discovered.

Testing and evaluation is a critical and challenging step. No value is added by the data mining task until the business value obtained from discovered knowledge patterns is identified and recognized. Determining the business value from discovered knowledge patterns is somewhat similar to playing with puzzles. The extracted knowledge patterns are pieces of the puzzle that need to be put together in the context of the specific business purpose. The success of this identification operation depends on the interaction among data analysts, business analysts, and decision makers (e.g., business managers). Data analysts may fully understand the data mining objectives and what they mean to the business, and business analysts and decision makers may not have the technical knowledge to interpret the results

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

of sophisticated mathematical solutions; therefore, interaction among them is necessary. In order to properly interpret knowledge patterns, it is often necessary to use a variety of tabulation and visualization techniques (e.g., pivot tables, cross-tabulation of findings, pie charts, histograms, box plots, scatterplots).

Step 6: Deployment

Development and assessment of models is not the end of a data mining project. Even if the purpose of a model is to do a simple exploration of the data, the knowledge gained from such exploration needs to be organized and presented in a way that the end user can understand and benefit from. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process across the enterprise. In many cases, it is the customer, not the data analyst, who carries out the deployment steps. However, even if an analyst will not carry out the deployment effort, it is important for the customer to understand up front what actions need to be carried out in order to actually make use of the created models.

The deployment step of the CRISP-DM process may also include maintenance activities for the deployed models. Because the business is constantly changing, the data that reflects the business activities is also changing. Over time, the models (and the patterns embedded within them) built on the old data may become obsolete, irrelevant, or misleading. Therefore, monitoring and maintaining the models are important if the data mining results are to become a part of the day-to-day business and its environment. Careful preparation of a maintenance strategy helps avoid unnecessarily long periods of incorrect use of data mining results. In order to monitor the deployment of the data mining result(s), a project needs a detailed plan for the monitoring process, which may not be trivial with complex data mining models.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The CRISP-DM process is the most complete and most popular data mining methodology practice in industry as well as in academia. Rather than use it as is, practitioners add their own insights to make it specific to their style of practice.

SEMMA

In order to be applied successfully, a data analytics project must be viewed as a multifaceted process rather than as a mere set of tools or techniques. The success of the project depends on the logical and optimal use of both soft and hard skills and methods. The standardized processes are developed and proposed by groups of individuals and consortiums of companies to achieve such optimal execution of analytics projects. In addition to KDD and CRISP-DM, there is another well-known methodology, called SEMMA, primarily developed by the SAS Institute, which is one of the largest analytics companies in the world. The acronym SEMMA stands for Sample, Explore, Modify, Model, Assess (SAS, 2020). Beginning with a statistically representative sample of data, SEMMA intends to make it easy to apply exploratory statistical and visualization techniques, select and transform the most significant predictive variables, model the variables to predict outcomes, and confirm the model's accuracy. Figure 3.3 provides a pictorial representation of SEMMA.

By assessing the outcome of each stage in the SEMMA process, an analyst can determine how to model new questions raised by the previous results and thus return to earlier steps for additional refinement of the data. Like CRISP-DM, SEMMA is driven by a highly iterative experimentation cycle. The following sections describe the five steps of the SEMMA process.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

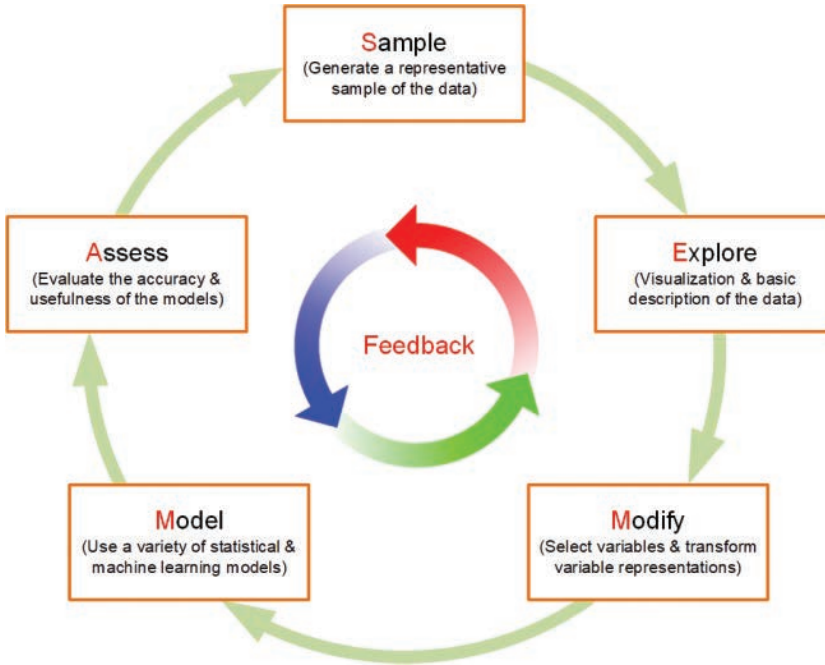


Figure 3.3 SEMMA: SAS Institute's Standardized Process for Data Analytics

Step 1: Sample

In the sample step, a portion of a large data set (big enough to contain significant information, yet small enough to manipulate quickly) is extracted. For optimal cost and computational performance, some (including the SAS Institute) advocate using a sampling strategy that involves applying a reliable, statistically representative sample of the full-detail data. In the case of very large data sets, mining a representative sample instead of the whole volume may drastically reduce the processing time required to get crucial business information. If general patterns appear in the data as a whole, they will be traceable in a representative sample. If a niche (a rare pattern) is so tiny that it is not represented in a sample and yet is so important that it influences the big picture, it should be discovered using exploratory data description methods. It is also a good idea to create partitioned data sets for better accuracy assessment.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The sample step of the SEMMA process involves three substeps:

- **Training.** Used for model fitting.
- **Validation.** Used for assessment and to prevent overfitting.
- **Test.** Used to obtain an honest assessment of how well a model generalizes.

For a more detailed discussion and relevant techniques for assessment and validation of data mining models, see Chapter 4.

Step 2: Explore

During the second step of the SEMMA process, the analyst searches for unanticipated trends and anomalies in order to gain a better understanding of the data set. The analyst explores the data visually or numerically for inherent trends or groupings. Exploration helps refine and redirect the discovery process. If visual exploration does not reveal clear trends, the analyst can explore the data by using statistical techniques, including factor analysis, correspondence analysis, and clustering. For example, in data mining for a direct mail campaign, clustering might reveal groups of customers with distinct ordering patterns. Limiting the discovery process to each of these distinct groups individually may increase the likelihood of exploring richer patterns that may not be strong enough to be detected if the whole data set is to be processed together.

Step 3: Modify

During the third step of the SEMMA process, the analyst creates, selects, and transforms the variables on which to focus the model construction process. Based on the discoveries in the exploration phase, the analyst may need to manipulate data to include information such as the grouping of customers and significant subgroups or to introduce new variables. It may also be necessary to look for outliers and reduce the number of variables in order to narrow them down to the

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

most significant ones. The analyst may also need to modify data when data that is mined changes. Because data mining is a dynamic, iterative process, an analyst can update data mining methods or models when new information is available.

Step 4: Model

In the fourth step of the SEMMA process, the analyst searches for a variable combination that reliably predicts a desired outcome. The goal is to construct models that explain patterns in the data. Modeling techniques in data mining include artificial neural networks, decision trees, rough set analysis, support vector machines, logistic models, and other statistical models, such as time-series analysis, memory-based reasoning, and principal component analysis. Each type of model has particular strengths and is appropriate in specific data mining situations and for different types of data. For example, artificial neural networks are very good at fitting highly complex nonlinear relationships, and rough set analysis is known to produce reliable results with uncertain and imprecise problem situations.

Step 5: Assess

In the final step of the SEMMA process, the analyst evaluates the usefulness and reliability of the findings from the data mining process. This is basically an estimation of how well the model performs. A common way to assess a model is to apply it to a portion of a data set that was put aside during the sampling stage and not used during model building. If the model is valid, it should work for this reserved sample as well as for the sample used to construct the model. Similarly, it is possible to test the model against known data. For example, if you know which customers in a file had high retention rates and your model predicts retention, you can check to see whether the model selects those customers accurately. In addition, practical applications of the model, such as partial mailings in a direct mail campaign, help prove the validity of a model.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

SEMMA Versus CRISP-DM

The SEMMA approach is quite compatible with the CRISP-DM approach. Both aim to streamline the knowledge discovery process. Both were created as broad frameworks that should be adapted to specific circumstances. In both, once models are obtained and tested, they can be deployed to gain value with respect to business or research application. Even though they have the same goal and are similar, SEMMA and CRISP-DM have a few differences, as outlined in Table 3.1.

Table 3.1 Comparison of CRISP-DM and SEMMA

Task	CRISP-DM	SEMMA	Comments
Project initiation	Business understanding	—	In this phase, CRISP-DM includes activities like project initiation, problem definition, and goal setting. SEMMA does not have a step for this phase.
Data access	Data understanding	Sample Explore	In this phase, both CRISP-DM and SEMMA have steps to access, sample, and explore the data.
Data transformation	Data preparation	Modify	In this phase, both CRISP-DM and SEMMA process the data to make it amenable to machine processing.
Model building	Model building	Model	In this phase, both CRISP-DM and SEMMA build and test various models.
Project evaluation	Testing and evaluation	Assess	In this phase, both CRISP-DM and SEMMA assess the findings against the project goals.
Project finalization	Deployment	—	CRISP-DM prescribes deploying the results, while SEMMA does not explicitly have a step for this.

Six Sigma for Data Mining

Six Sigma is a popular business management philosophy that focuses on reducing deviations (i.e., sigma) from perfection by rigorously and systematically using proven quality control and business

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

process improvement principles and techniques. This hugely popular management philosophy was first introduced by Motorola in the 1980s in the context of manufacturing process improvement and management. Since then it has been adopted by many companies and organizations in a wide variety of business contexts beyond manufacturing. Ideally, Six Sigma promotes near-zero defects and near-zero tolerance, which can roughly be translated in the business context as error-free perfect business execution. Six Sigma in the business world, also goes by the name DMAIC, involves the following five steps: Define, Measure, Analyze, Improve, and Control.

Because of its success in many other business problems and settings, DMAIC methodology is also applied to data analytics/data science projects. Figure 3.4 shows the DMAIC methodology as a simplified flow diagram. The following sections describe these steps.

Step 1: Define

The first step in the DMAIC process is about getting the project set up and started. It involves several steps: (1) gaining a thorough understanding of the business needs, (2) identifying the most pressing problem, (3) defining goals and objectives, (4) identifying and defining the data and other resources needed to investigate the business problem, and (5) developing a detailed project plan. As you may have noticed, there is significant overlap between this step and the first step in the CRISP-DM process.

Step 2: Measure

The second step of the DMAIC process involves assessing the mapping between organizational data repositories and the business problem. Since data mining requires relevant, clean, and usable data, identification and creation of such resources is of critical importance to the success of a data mining project. In this step, the identified data sources are consolidated and transformed into a format that is amenable to machine processing.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

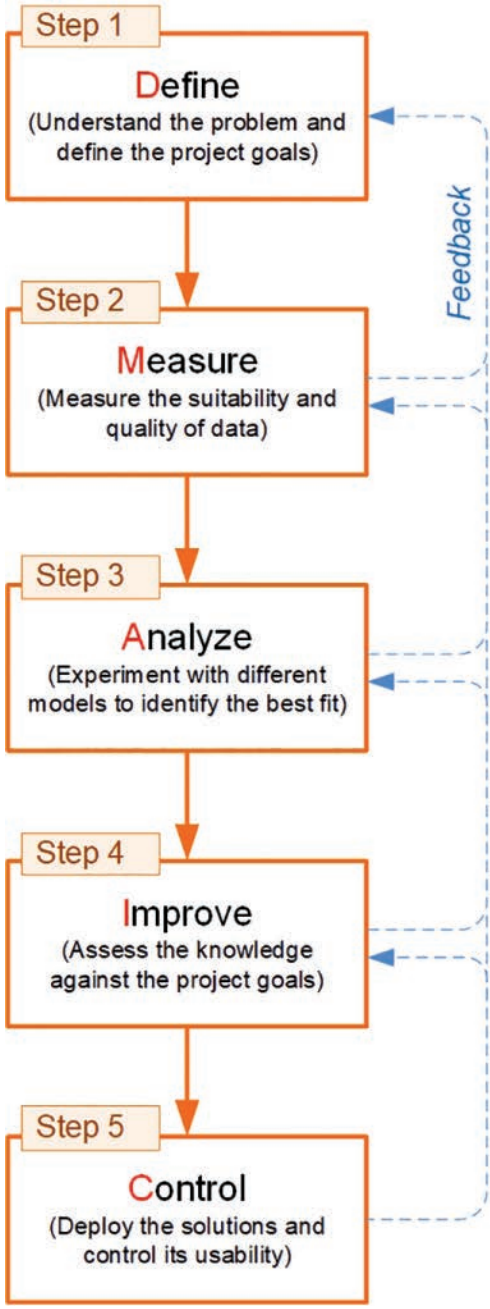


Figure 3.4 Six Sigma DMAIC Methodology

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Step 3: Analyze

The third step of the DMAIC process is to use a series of data mining techniques to develop models. There are many possible techniques, and most of them are machine learning techniques with many parameters to optimize. There is not a single best technique, so an analyst needs to apply several probable techniques and experiment with them to identify and develop the most appropriate model.

Step 4: Improve

The fourth step of the DMAIC process is to investigate the improvement possibilities. Improvements can be at the technique level or at the business problem level. For instance, if the model results are not satisfactory, other more sophisticated techniques (e.g., ensemble systems) can be used to boost the performance of the models. Also, if the modeling results are not clearly addressing the business problem, the analyst can loop back to previous steps to reexamine and improve the structure of the analysis. The analyst can also further investigate and restate the business problem.

Step 5: Control

The last step of the DMAIC process is to assess the project outcomes, and if they are found to be satisfactory, the models and result are disseminated to decision makers and/or integrated into the existing business intelligence systems for automation.

Six Sigma–based DMAIC methodology resembles the CRISP-DM process. There is no evidence to suggest that one was inspired by the other. However, because these two methodologies involve rather logical and straightforward steps in any business system analysis effort, they need not have inspired one another, and their similarity might simply be coincidental. DMAIC is rarely compared to CRISP-DM because these two methodologies are used to address different

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

types of business applications. Many organizations around the world are using DMAIC and CRISP-DM and their variants to conduct process improvement and data mining projects.

Which Methodology Is Best?

Some data mining processes are more elaborate than the others, and there is no sure way to compare them. They all have pros and cons. Some are more problem focused, while others are more analysis driven. A business that does data mining adopts one of these methodologies and then is likely to slightly modify it to make it fit the business or the data. KDnuggets, a well-known and well-respected web portal for data mining, conducted a survey in which it asked "Which one is the best?" Figure 3.5 shows the polling results of that survey (KDNuggets.com).

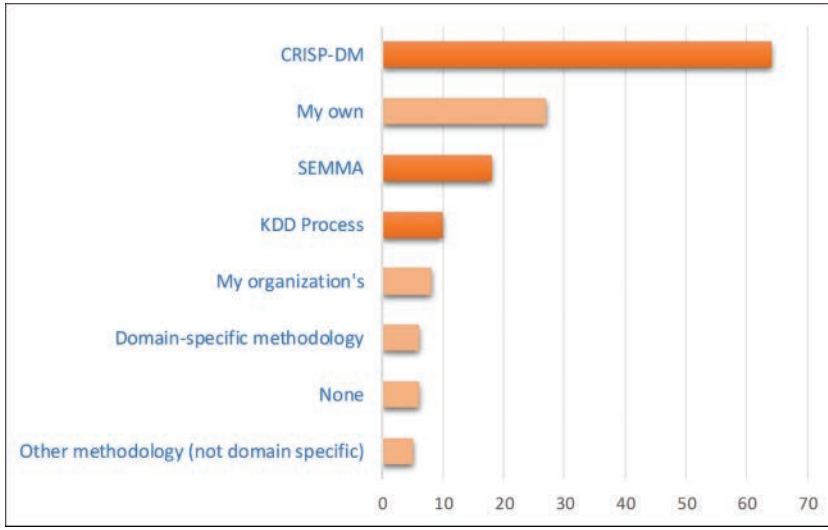


Figure 3.5 A Poll on Popular Data Mining Methodologies

Source: Used with permission from KDnuggets.com

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

As the survey results indicate, CRISP-DM dominates as the most popular data mining process. In addition, many of the ones that fall under "My own" are also customizations of CRISP-DM. Compared to the other methodologies, CRISP-DM is the most complete and most mature process for data mining.

An Exemplary Data Mining Case: Mining Cancer Data for New Knowledge

Cancer is one of the most deadly diseases. According to the American Cancer Society, half of all men and one-third of all women in the United States will develop cancer during their lifetimes, and approximately 1.5 million new cancer cases were diagnosed in 2014. Cancer is the second-most-common cause of death in the United States and in the world, exceeded only by cardiovascular disease. In 2014, more than 500,000 Americans are expected to die of cancer—more than 1,300 people a day—accounting for nearly 1 of every 4 deaths.

Cancer is a group of diseases generally characterized by uncontrolled growth and spread of abnormal cells. If that growth and/or spread is not controlled, it can result in death. Cancer is believed to be caused by both external factors (e.g., tobacco use, infectious organisms, exposure to chemicals or radiation) and internal factors (e.g., inherited mutations, hormones, immune conditions, mutations that occur due to metabolism). These causal factors may act together or in sequence to initiate or promote carcinogenesis. Cancer is treated with surgery, radiation, chemotherapy, hormone therapy, biological therapy, and targeted therapy. Survival statistics vary greatly by cancer type and stage at diagnosis.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The five-year relative survival rate for all cancers is improving. Between 1991 and 2013, the cancer mortality rate declined by 20%, translating to the avoidance of about 1.2 million deaths during that time frame. That's more than 400 lives saved per day! The improvement in survival reflects progress in diagnosing certain cancers at an earlier stage and improvements in treatment. Further improvements are needed to prevent and treat cancer.

Even though cancer research has traditionally been clinical and biological in nature, in recent years, data-driven analytic studies have become a common complement. In medical domains where data- and analytics-driven research has been applied successfully, novel research directions have been identified to further advance clinical and biological studies. Using various types of data, including molecular, clinical, literature-based, and clinical-trial data, along with suitable data mining tools and techniques, researchers have been able to identify novel patterns, paving the road toward a cancer-free society.

In one study, Delen, Walker, and Kadam (2005) used three data analytics methods—artificial neural networks, decision trees, and logistic regression—to develop prediction models for breast cancer survival using a large and feature-rich data set (more than 200,000 cases and hundreds of variables). Using a *10*-fold cross-validation method to measure the unbiased estimate of the prediction models for performance comparison purposes, the results indicated that the decision tree method (with C5 algorithm) was the best predictor, with 93.6% accuracy on the holdout sample (which was the best prediction accuracy reported in the literature at that time), followed by artificial neural networks, with 91.2% accuracy, and logistic regression, with 89.2% accuracy. (Figure 3.6 shows the complete results table.) Further analysis of prediction models revealed the prioritized importance of the prognostic factors, which can be used as a basis for further clinical and biological research studies.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

This earlier research paper has been cited more than 1,200 times, making it one of the seminal papers in this domain.

Fold No	Neural Networks (MLP)					Decision Tree Induction (C5)					Logistic Regression				
	Confusion Matrix		Accuracy	Sensitivity	Specificity	Confusion Matrix		Accuracy	Sensitivity	Specificity	Confusion Matrix		Accuracy	Sensitivity	Specificity
1	7571	844	0.9165	0.9535	0.8719	7828	587	0.9363	0.9586	0.9078	7672	743	0.8912	0.9015	0.8766
	369	5747				338	5778				838	5277			
2	7589	729	0.9271	0.9578	0.8905	7737	581	0.9403	0.9639	0.9113	7543	773	0.8906	0.9018	0.8756
	334	5926				290	5970				821	5439			
3	7567	768	0.9214	0.9537	0.8818	7741	594	0.9356	0.9584	0.9065	7602	732	0.8915	0.9011	0.8779
	367	5730				336	5761				834	5261			
4	7508	796	0.9169	0.9480	0.8798	7703	601	0.9356	0.9582	0.9076	7607	696	0.8951	0.9017	0.8860
	412	5824				336	5900				829	5407			
5	7609	809	0.9239	0.9549	0.8903	7789	629	0.9348	0.9607	0.9021	7659	757	0.8908	0.9022	0.8747
	359	6565				319	5796				830	5284			
6	7390	908	0.8914	0.9179	0.8581	7694	604	0.9363	0.9604	0.9062	7554	743	0.8900	0.8992	0.8771
	661	5491				317	5835				847	5305			
7	7298	751	0.9081	0.9290	0.8823	7464	585	0.9374	0.9605	0.9095	7333	716	0.8949	0.9037	0.8831
	558	5631				307	5882				781	5408			
8	7069	977	0.9024	0.9442	0.8565	7436	610	0.9353	0.9594	0.9068	7269	773	0.8894	0.9001	0.8756
	418	5832				315	5935				807	5443			
9	7290	958	0.9040	0.9454	0.8559	7621	627	0.9360	0.9631	0.9027	7501	747	0.8923	0.9036	0.8767
	421	5691				292	5820				800	5310			
10	7475	764	0.9098	0.9330	0.8809	7625	614	0.9349	0.9591	0.9052	7518	716	0.8938	0.9022	0.8824
	537	5651				325	5863				815	5372			
Mean			**0.9121**	**0.9437**	**0.8748**			**0.9362**	**0.9602**	**0.9066**			**0.8920**	**0.9017**	**0.8786**
St. Dev.			**0.0111**	**0.0131**	**0.0135**			**0.0016**	**0.0019**	**0.0028**			**0.0020**	**0.0014**	**0.0038**

Confusion matrix shows the classification of the cases in the test dataset. In confusion matrix, the columns denote the actual cases and the rows denote the predicted. Accuracy = (TP + TN)/(TP + FP + TN + FN); sensitivity = TP/(TP + FN); specificity = TN/(TN + FP).

Figure 3.6 Tabular Results for the 10-Fold Cross-Validation of All Model Types

In a more recent study, Zolbanin, Delen, and Zadeh (2015) studied the impact of comorbidity in cancer survivability. Although prior research has shown that diagnostic and treatment recommendations might be altered based on the severity of comorbidities, chronic diseases are still being investigated in isolation from one another in most cases. To illustrate the significance of concurrent chronic diseases in the course of treatment, their study used the Surveillance, Epidemiology, and End Results (SEER) program's cancer data to create two comorbid data sets: one for breast and female genital cancers and another for prostate and urinal cancers. Several popular machine learning techniques were applied to the resultant data sets to build predictive models. Figure 3.7 shows a graphical depiction of the analytics methodology used in this research study. Comparison of the results has shown that having more information about comorbid conditions of patients can improve the predictive power of a model, which in turn can help practitioners make better diagnostic and treatment decisions. Therefore, the study suggested that proper identification, recording, and use of patients'

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

comorbidity status can potentially lower treatment costs and ease the health care–related economic challenges.

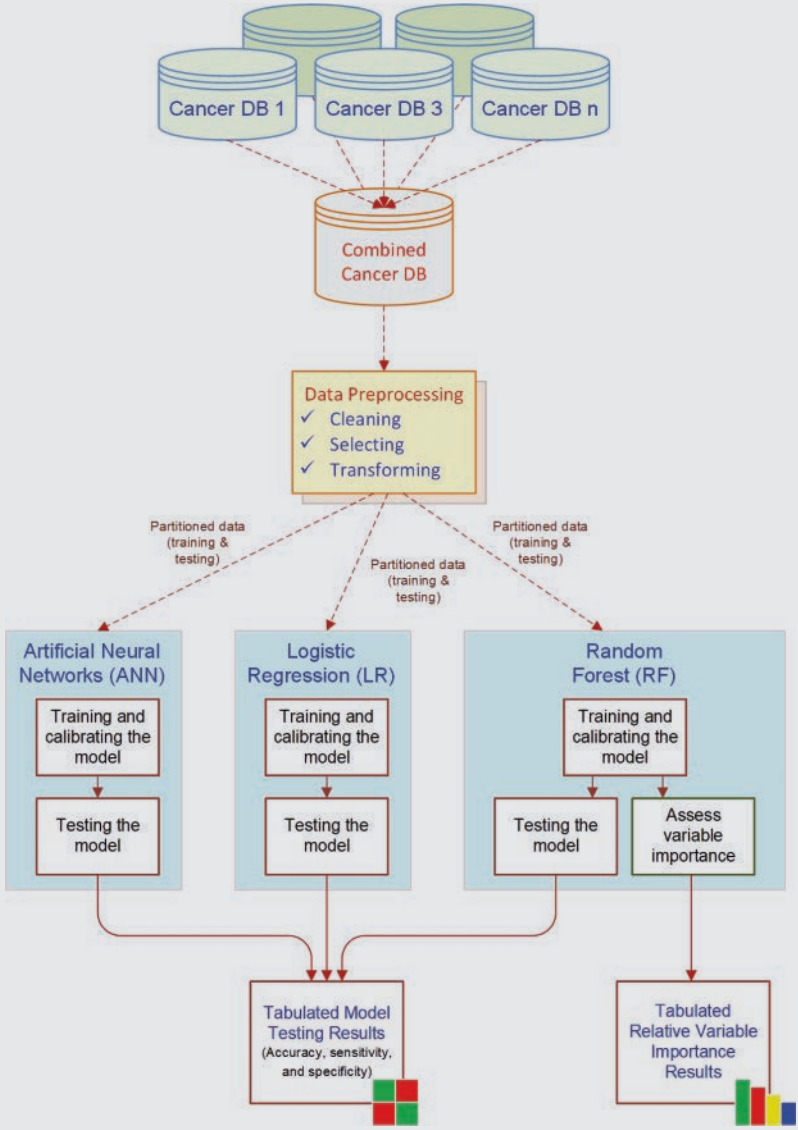


Figure 3.7 A Graphical Depiction of the Analytics Methodology Used in the Study

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

These examples (among many others in the medical literature) show that advanced analytics and data science techniques can be used to develop models that possess a high degree of predictive as well as explanatory power. Although data mining methods are capable of extracting patterns and relationships hidden deep in large and complex medical databases, without cooperation and feedback from medical experts, their results are not of much use. The patterns found via data mining methods should be evaluated by medical professionals who have years of experience in the problem domain and can decide whether they are logical, actionable, and novel enough to warrant new research directions. In short, data mining is not meant to replace medical professionals and researchers but to complement their invaluable efforts to provide data-driven new research directions and to ultimately save more human lives.

Summary

This chapter presents the most commonly used processes for carrying out predictive analytics and data mining projects. A standardized process is the manifestation and presentation of the best practices (in a complex practice area) using a logical, scientific, and practical approach. Following a standard process helps data scientists increase the odds of obtaining accurate, consistent, and useful project outcomes. Among the several standard processes, CRISP seems to be the most commonly used in both industry and academics, and it is therefore the one explained in greatest detail in this chapter.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

References

Chapman, P., J. Clinton, R. Kerber, T. Khabaza, et al. (2013). CRISP-DM 1.0. www.the-modeling-agency.com/crisp-dm.pdf (accessed July 2020).

Delen, D. (2009). "Analysis of Cancer Data: A Data Mining Approach," *Expert Systems*, 26(1): 100–112.

Delen, D., & N. Patil. (2006, January). "Knowledge Extraction from Prostate Cancer Data," *Proceedings of the 39th Annual Hawaii International Conference on Systems Sciences*, 5: 92b.

Delen, D., G. Walker, & A. Kadam. (2005). "Predicting Breast Cancer Survivability: A Comparison of Three Data Mining Methods," *Artificial Intelligence in Medicine*, 34(2): 113–127.

Fayyad, U., G. Piatetsky-Shapiro, and P. Smyth. (1996). "From Knowledge Discovery in Databases," *AI Magazine*, 17(3): 37–54.

KDnuggets. (2019). *Python Leads the 11 Top Data Science, Machine Learning Platforms: Trends and Analysis*. https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html (accessed June 2020).

SAS. (2020). *Introduction to SEMMA.* https://documentation.sas.com/?docsetId=emref&docsetTarget=n061bzurmej4j3n1jnj8bbjjm1a2.htm&docsetVersion=14.3&locale=en (accessed July 2020).

Zolbanin, H. M., D. Delen & A. H. Zadeh. (2015). "Predicting Overall Survivability in Comorbidity of Cancers: A Data Mining Approach," *Decision Support Systems*, 74: 150–161.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

4

Data and Methods for Predictive Analytics

Data mining and predictive analytics are popular because they make it possible to discover hidden patterns deep inside large data repositories that organizations are automatically accumulating these days. Data by itself is not valuable to organizations that are desperately looking for ways to make evidence-based intelligent decisions. It is only after methods are applied to data that the invaluable knowledge organizations need can be brought out. A large variety of methods and algorithms can be used to carry out different types of data mining tasks. Each of these methods has advantages and disadvantages for a given project, depending on the specific characteristics of the problem being addressed and the data set being used. Identifying the best method to use usually requires quite a bit of experimentation and sound comparative assessment.

This chapter provides detailed yet nontechnical descriptions of these methods and commonly applied assessment techniques. Before we get to that, though, we need to first talk about data as it relates to data mining.

The Nature of Data in Data Analytics

Data refers to a collection of facts usually obtained as a result of experiences, observations, or experiments. Data may consist of numbers, letters, words, images, voice recordings, and other variables.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Data is often viewed as the lowest level of abstraction from which information and then knowledge are derived.

At the highest level of abstraction, we can classify data as structured and unstructured (or semi-structured). Unstructured/semi-structured data is composed of any combination of text, image, voice, and web content. (Unstructured/semi-structured data is covered in more detail in Chapter 7, "Text Analytics, Topic Modeling, and Sentiment Analysis.") Structured data is the data that data mining algorithms use, and it can be classified as categorical or numeric. Categorical data can be subdivided into nominal or ordinal data, and numeric data can be subdivided into interval or ratio data. Figure 4.1 shows a simple taxonomy of data in data mining.

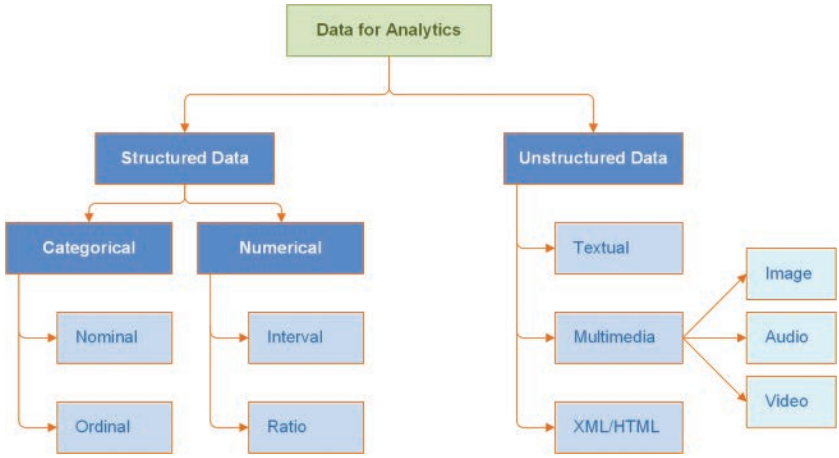


Figure 4.1 A Simple Taxonomy of Data in Data Mining

Categorical data falls into multiple classes used to divide a variable into specific groups. Examples of categorical variables include race, sex, age group, and educational level. Although age group and educational level may also be considered in a numeric manner by using exact values for age and highest grade completed, it is often more informative to categorize such variables into a relatively small

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

number of ordered classes. The categorical data may also be called *discrete data*, implying that it represents a finite number of values with no continuum between them. Even if the values used for the categorical (or discrete) variables are numeric, these numbers are nothing more than symbols and do not imply the possibility of calculating fractional values.

Nominal data includes measurements of simple codes assigned to objects as labels, which are not measurements. For example, the variable marital status can be generally categorized as (1) single, (2) married, or (3) divorced. Nominal data can be represented with binomial values having two possible values (e.g., yes/no, true/false, good/bad), or multinomial values having three or more possible values (e.g., brown/green/blue, white/black/Latino/Asian, single/married/divorced).

Ordinal data includes codes assigned to objects or events as labels that also represent the rank order among them. For example, the variable credit score can be generally categorized as (1) low, (2) medium, or (3) high. Similar ordered relationships can be seen in variables such as age group (i.e., child/young/middle-aged/elderly) and educational level (i.e., high school/college/graduate school). Some data mining algorithms, such as ordinal multiple logistic regression, take into account this additional rank order information to build a better classification model.

Numeric data represents the numeric values of specific variables. Examples of numerically valued variables include age, number of children, total household income (in U.S. dollars), travel distance (in miles), and temperature (in degrees Fahrenheit). Numeric values representing a variable can be integer (taking only whole numbers) or real (taking also the fractional number). Numeric data may also be called *continuous data*, implying that the variable contains continuous measures on a specific scale that allows insertion of interim values.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Unlike a discrete variable, which represents finite, countable data, a continuous variable represents scalable measurements, and it is possible for the data to contain an infinite number of fractional values.

Interval data includes variables that can be measured on interval scales. A common example of interval scale measurement is temperature on the Celsius scale. In this particular scale, the unit of measurement is 1/100 of the difference between the melting temperature and the boiling temperature of water in atmospheric pressure; that is, there is not an absolute zero value.

Ratio data includes measurement variables commonly found in the physical sciences and engineering. Mass, length, time, plane angle, energy, and electric charge are examples of physical measures that are ratio data. The scale type takes its name from the fact that measurement is the estimation of the ratio between a magnitude of a continuous quantity and a unit magnitude of the same kind. Informally, the distinguishing feature of a ratio scale is the possession of a nonarbitrary zero value. For example, the Kelvin temperature scale has a nonarbitrary zero point of absolute zero, which is equal to –273.15 degrees Celsius. This zero point is nonarbitrary because the particles that comprise matter at this temperature have zero kinetic energy.

Other data types, including textual, multimedia (e.g., image, audio, video), and XML/HTML need to be converted into some form of categorical or numeric representation before they can be processed by data mining algorithms. (For more details about how to mine knowledge from unstructured and semi-structured data, see Chapter 7.) Data can also be geographically oriented, so that the numeric and nominal data is enriched with locational/geographic information; this is often referred to as *spatial data*. Furthermore, regardless of its structure, data can also be classified as static or dynamic (i.e., temporal or time-series data), based on its relationship to time.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Some data mining methods and algorithms are very particular about the type of data they can handle. Providing them with incompatible data types may lead to incorrect models or (more often) halt the model development process. For example, some data mining methods (e.g., neural networks, support vector machines, logistic regression) need all the variables (both input and output) represented as numerically valued variables. The nominal or ordinal variables are converted into numeric representations using some type of 1-of-*N* pseudo variables; for example, a categorical variable with three unique values may be transformed into three pseudo variables with binary values (1 or 0). Because this process may increase the number of variables, it is important to be cautious about the effect of such representations, especially for the categorical variables that have large numbers of unique values.

Similarly, some data mining methods, such as ID3 (a classic decision tree algorithm) and rough sets (a relatively new rule-induction algorithm), need all the variables represented as categorically valued variables. Early versions of these methods required the user to discretize numeric variables into categorical representations before processing them with the algorithm. The good news is that most implementations of these algorithms in widely available software tools accept a mix of numeric and nominal variables and internally make the necessary conversions before processing the data.

Preprocessing of Data for Analytics

The quality and usefulness of models created using data mining methods depends largely on the quality of the data used to build them. In that sense, perhaps the GIGO rule (i.e., garbage-in, garbage-out) is more applicable to data mining than to any other application domain. Therefore, the purpose of data preparation (commonly called *data preprocessing*) is to eliminate the possibility of GIGO errors by

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

methodically cleaning and transforming data into a standard form so that it can be properly analyzed using appropriate data mining methods. Compared to the other activities in the data mining process, data preprocessing demands and consumes the most time and effort, often accounting for more than 80% of the total time spent on a project. Preprocessing requires such an enormous effort because real-world data is generally incomplete (lacking attribute values, lacking certain attributes of interest, or containing only aggregate data), dirty/noisy (containing errors or outliers), and inconsistent (containing discrepancies in codes or names). Therefore, diligent effort is necessary to clean and transform the data into a form that can be mined while making sure that the preprocessing does not bias the data; it is important to preserve the inherent patterns and relationships of the data. Figure 4.2 shows the four-step systematic process commonly followed to convert raw real-world data into well-formed data sets that can be mined.

In the first phase of data preprocessing, the relevant data is collected from the identified sources, the necessary records and variables are selected (filtering out the unnecessary data, based on an intimate understanding of the data), and the records coming from multiple data sources are integrated. During this phase, domain-specific understanding of the data is applied to properly handle the synonyms and homonyms in the data.

In the second phase of data preprocessing, known as data scrubbing, the data is cleaned. In this step, the values in the data set are identified and dealt with. In some cases, missing values are an anomaly in the data set, in which case they need to be imputed (i.e., filled with a most probable value) or ignored; in other cases, the missing values are a natural part of the data set (e.g., the household income field is often left unanswered by people who are in the top income tier). In this step, the analyst should also identify noisy values in the data (i.e., outliers) and smooth them out. In addition, inconsistencies (unusual values within a variable) in the data should be handled using domain knowledge and/or expert opinion.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

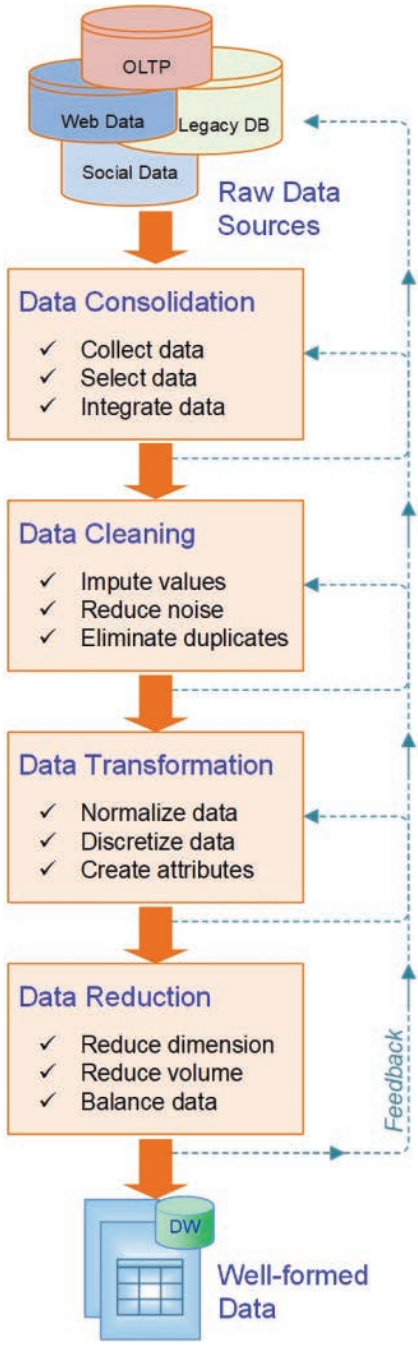


Figure 4.2 A Systematic Process for Data Preparation

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

In the third phase of data preprocessing, the data is transformed for better processing. For instance, in many cases, data is normalized between a certain minimum and a certain maximum for all variables in order to mitigate the potential bias of one variable dominating other variables having smaller values; for example, the analyst would not want data with large numeric values, such as for household income, to dominate the number of dependents or years in service, which may potentially be more important. Another transformation that takes place at this point is discretization and/or aggregation. In some cases, the numeric variables are converted to categorical values (e.g., low, medium, high); in other cases, a nominal variable's unique value range is reduced to a smaller set, using concept hierarchies (e.g., instead of using the individual states with 50 different values, using several regions for a variable that shows location) in order to have a data set that is more amenable to computer processing. Yet other cases might call for creating new variables based on the existing ones in order to magnify the information in a collection of variables in the data set. For instance, in an organ transplantation data set, an analyst might choose to use a single variable showing the blood type match (1: match, 0: no match) as opposed to using separate multinomial values for the blood types of the donor and the recipient. Such simplification may increase the information content while reducing the complexity of the relationships in the data.

The final phase of data preprocessing is data reduction. Data miners like to have large data sets, but having too much data can be a problem. In the simplest sense, we can visualize the data commonly used in data mining projects as a flat file consisting of two dimensions: variables (the number of columns) and cases/records (the number of rows). In some cases (e.g., with image processing, genome projects with complex microarray data), the number of variables can be rather large, and the analyst must reduce the number to a manageable size. Because the variables are treated as different dimensions that describe the phenomenon from different perspectives, in data

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

mining, this process is commonly called *dimensional reduction*. Even though there is not a single best way to accomplish this task, an analyst can use the findings from previously published literature, consult domain experts, run appropriate statistical tests (e.g., principal component analysis, independent component analysis), or, ideally, use a combination of these techniques to successfully reduce the dimensions in the data into a more manageable and relevant subset.

In addition to the complications related to multidimensionality, real-world data may also have issues with very large numbers of samples. That is, some data sets may include millions or billions of *records* (also called *samples* or *cases*). Even though computing power is increasing steadily, providing more horsepower for data analysis, processing such a large number of records may still not be feasible or practical; a single analysis may take a prohibitively long time to complete. In such cases, an analyst may need to analyze only a subset of the data. The underlying assumption of sampling is that the subset of the data will contain all relevant patterns of the complete data set. In a homogeneous data set, such an assumption may hold well, but real-world data is hardly ever homogeneous. An analyst should be extremely careful in selecting a subset of data that reflects the essence of the complete data set and is not specific to a subgroup or subcategory. The data is usually sorted on some variable, and taking a section of the data from the top or bottom may lead to a biased data set on specific values of the indexed variable; therefore, an analyst should always try to randomly select the records from the sample set. For skewed data, straightforward random sampling may not be sufficient, and stratified sampling—in which a proportional representation of different subgroups in the data is represented in the sample data set—may be required. In addition, it is a good practice to balance highly skewed data by either oversampling the less represented classes or undersampling the more represented classes. Research has shown that balanced data sets tend to produce better prediction models than unbalanced ones.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The process shown in Figure 4.2 is not a simple unidirectional, one-pass (top-to-bottom) effort. Rather, it is an iterative process that usually requires going backward (following the feedback loops) to make corrections and adjustments. The quality of outcomes (i.e., how well formed the data ends up being) depends on the meticulous execution of this highly repetitive process between the later and earlier phases. The core activities in data preprocessing are summarized in Table 4.1, which maps the main tasks (along with their problem descriptions) to a representative list of methods and algorithms.

Table 4.1 Summary of Data Preprocessing Tasks and Potential Methods

Main Task	Subtasks	Popular Methods
Data consolidation	Accessing and collecting the data	SQL queries, software agents, web crawlers, data access APIs
	Selecting and filtering the data	Domain expertise, SQL queries, statistical tests
	Integrating and unifying the data	SQL queries, domain expertise, ontology-driven data mapping
Data cleaning	Handling missing values in the data	Filling in the missing values (imputations) with the most appropriate values (mean, median, min/max, mode, etc.); recoding the missing values with a constant such as ML; removing the record of the missing value; doing nothing
	Identifying and reducing noise in the data	Identifying outliers in data with simple statistical techniques (e.g., averages, standard deviations) or with cluster analysis; removing the outliers or smoothing by using binning, regression, or simple averages
	Finding and eliminating erroneous data	Identifying erroneous values in data (other than outliers), such as odd values, inconsistent class labels, odd distributions; using domain expertise to correct the values or remove the records that contain erroneous values
Data transformation	Normalizing the data	Reducing the range of values in each numerically valued variable to a standard range (e.g., 0 to 1, –1 to +1) by using a variety of normalization or scaling techniques

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Main Task	Subtasks	Popular Methods
	Discretizing or aggregating the data	If needed, converting the numeric variables into discrete representations using range- or frequency-based binning techniques; for categorical variables, reducing the number of values via application of proper concept hierarchies
	Constructing new attributes	Deriving new and more informative variables from the existing ones, using a wide range of mathematic functions (as simple as addition and multiplication or as complex as a hybrid combination of log transformations)
Data reduction	Reducing the number of attributes	Using principal component analysis, independent component analysis, chi-square testing, correlation analysis, or decision tree induction
	Reducing the number of records	Using random sampling, stratified sampling, or expert knowledge-driven purposeful sampling
	Balancing skewed data	Oversampling the less represented class records or undersampling the more represented class records

Data Mining Methods

A wide variety of data mining methods and algorithms are being developed and used by businesses and scientific organizations to shed light on a range of complex problems. These problems can be mapped to a few high-level data mining tasks. Depending on the need and the nature of the problem, the data mining task can be classified as prediction (either classification or regression), association, or clustering. Each of these data mining tasks can use a variety of data mining methods and algorithms. Figure 4.3 shows a graphical representation of these data mining tasks, methods, and algorithms. The following sections describe these broad categories of data mining tasks (i.e., prediction, association, and clustering) and the algorithms most commonly used with them (e.g., decision trees for prediction, the Apriori algorithm

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

for association, *k*-means for clustering). Chapter 5, "Algorithms for Predictive Analytics," explains several other data mining algorithms, including *k*-nearest neighbor, artificial neural networks, support vector machines, linear regression, logistic regression, and time-series forecasting. Most data mining software tools use more than one technique (or algorithm) for each of these data mining tasks/methods.

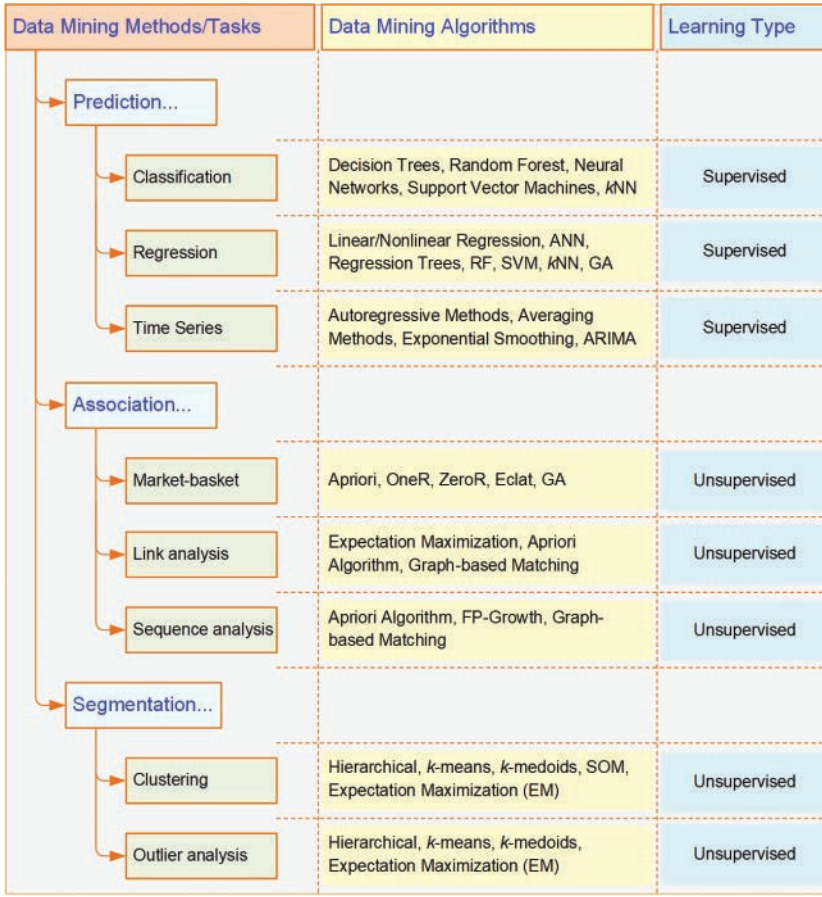


Figure 4.3 Data Mining Tasks/Methods and Representative Algorithms

Prediction

As discussed in Chapter 1, "Introduction to Analytics," prediction is used to foresee the future and answer the question "What will

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

happen?" or, more appropriately, "What is most likely to happen?" Business planning and problem solving rely heavily on the estimation of the future event and occurrences. Predictive analytics (and underlying prediction modeling techniques) are the enablers of these estimates. Regardless of the modeling technique used to do the predictive modeling, the underlying assumption is that what we already know about the past (in terms of recorded transaction and events in the form of historical data) is what we use as the main input to predict the future. We assume that the past will repeat itself in the future in some way, and it usually does; but if it doesn't, then there is no way of predicting the future.

Based on the variable we are predicting, the method of prediction can be labeled as *classification* (if we are predicting a discrete outcome), *regression* (if we are predicting a numeric value), or *time-series forecasting* (if we are predicting the future projection of a time-dependent value series). For instance, predicting whether a stock will close up or down at the end of the day is considered a classification, predicting the value difference of the stock at the end of the day is considered a regression, and predicting tomorrow's closing price of the stock based on the closing prices in the recent past is a time-series prediction. Even though they all are estimating the future value of the variable we are interested in, the value type and the nature of the variable determine the type of the prediction.

What differentiates classification from regression is the nature of the target variable: Whereas classification predicts a label (a discrete/nominal variable), regression predicts a number (a continues/numeric variable). There is no differentiation on the input variable; both types of methods of prediction can handle any combination of nominal and numeric variables (although some data transformation procedures need to be performed to convert the input data from discrete to numeric or numeric to discrete, based on the needs of the specific predictive modeling technique). Most of the predictive modeling methods covered in this chapter and Chapter 5 can

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

handle both classification- and regression-type prediction problems. For instance, while decision trees are used for classification, a similar structure called regression trees can be used for regression problems. Similarly, while multiple linear regression can be used for regression, multiple logistic regression is used for classification-type problems. Other popular techniques such as artificial neural networks, support vector machines, and *k*-nearest neighbor can easily be used for both classification- and regression-type prediction problems. In order to keep the coverage of the popular methods and algorithms for predictive modeling succinct and simple, in the following section we focus solely on the most commonly used method of prediction: classification.

Classification

Classification is perhaps the most commonly used predictive analytics and data mining method for real-world problems. Even in a regression-type prediction problem, where the target variable is a continuous numeric variable, a discretization is used to convert the target variable to a binomial or multinomial discrete representation, thereby converting the regression problem to a classification problem. Often, such a conversion is applied to make the prediction problem more understandable, domain related, robust, and user friendly, as in the case of the movie box-office prediction case study presented at the end of Chapter 3, "Standardized Processes for Predictive Analytics."

As a popular member in the machine learning family of techniques, classification learns patterns from past data (a set of information—traits, variables, and features—on characteristics of the previously labeled items, objects, or events) in order to place new instances (with unknown labels) into their respective groups or classes. For example, we could use classification to predict whether the weather on a particular day will be "sunny," "rainy," or "cloudy." Popular classification tasks include credit approval (i.e., good or bad credit risk), store location (e.g., good/moderate/bad), target marketing (e.g., likely

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

customer/no hope), fraud detection (i.e., yes/no), and telecommunication (e.g., likely/not likely to turn to another phone company). As mentioned earlier, if what is being predicted is a class label (e.g., "sunny," "rainy," or "cloudy"), the prediction problem is called a *classification*, whereas if it is a numeric value (e.g., temperature such as 68 degrees F), the prediction problem is called a *regression*.

Even though clustering (another popular data mining method) can also be used to determine groups (or class memberships) of things, there is a significant difference between the classification and clustering. Classification learns the function between the characteristics of things (i.e., independent variables) and their membership (i.e., output variable) through a supervised learning process in which both types (input and output) of variables are presented to the algorithm; in clustering, the membership of the objects is learned through an unsupervised learning process in which only the input variables are presented to the algorithm. Unlike classification, clustering does not have a supervising (or controlling) mechanism that enforces the learning process; instead, clustering algorithms use one or more heuristics (e.g., multidimensional distance measure) to discover natural groupings of objects.

The most common two-step methodology for classification-type prediction involves model development/training/learning and model testing/validation. In the model development phase, a collection of input data, including the actual class labels, is used. After a model has been trained, the model is tested against the holdout sample for accuracy assessment and eventually deployed for actual use, where it predicts classes of new data instances (where the class label is unknown).

Identifying the best prediction model out of all that is available is not an easy task. Although prediction accuracy is considered the most important, depending on the prediction problem specifics and the needs/wants of the decision maker and the application domain, other criteria need to be considered. At this stage, as a data scientist, what you will be dealing with is often called a *multi-criteria*

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

decision-making situation. The importance assigned to each criterion will be the source of objectively determining "the best overall" prediction model. A large number of factors/criteria can be considered when assessing prediction models. Following are some of the most commonly used ones:

- **Prediction accuracy.** In the context of classification-type prediction problems, this is the model's ability to correctly predict the class label of new or previously unseen data. Prediction accuracy is the most commonly used assessment factor for classification models. The next section provides a detailed explanation of this assessment metric.
- **Interpretability.** Interpretability is the level of understanding and insight provided by the model (e.g., how and/or what the model concludes on certain predictions).
- **Speed.** Speed refers to the computational cost involved in generating (i.e., training the model) and using (i.e., testing and/or deploying) the prediction model, where faster is better.
- **Robustness.** Robustness refers to a model's ability to make reasonably accurate predictions, given noisy data or data with missing and/or erroneous values.
- **Scalability.** Scalability is the ability to construct a prediction model efficiently; while the training data set increases in size (in terms of both number of rows/samples and columns/variables), the time it takes to generate a prediction model increases linearly.

Out of these top five assessment metrics, because of the changing perception toward predictive analytics and machine learning today, the most commonly considered ones are prediction accuracy and interpretability. While the other three metrics are becoming less popular due to recent advancements in hardware and software, prediction accuracy and interpretability are gaining more importance as

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

these models are becoming more and more critical parts of business-related decision support systems.

How to Estimate the True Accuracy of Classification Models

In classification problems, the primary source for all accuracy estimation metrics is a *confusion matrix*, also called a *classification matrix* or a *contingency table*. Figure 4.4 shows a confusion matrix for a two-class classification problem. The number of instances recorded along the diagonal from the upper left to the lower right represents correctly made prediction decisions, and the number of instances recorded outside this diagonal represents errors (i.e., incorrectly made predictions).

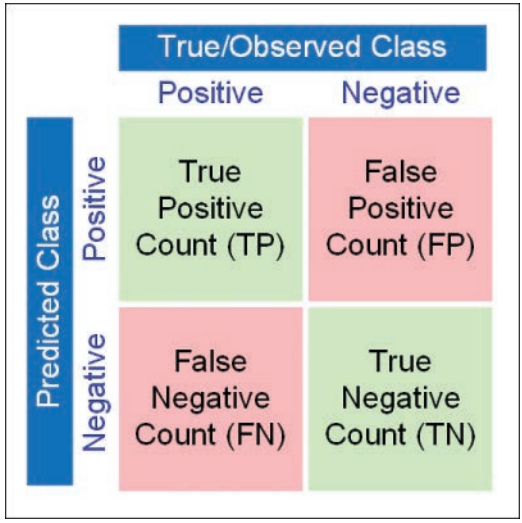


Figure 4.4 A Confusion Matrix for Binary Classification

When a classification problem is not binary, the confusion matrix gets bigger: It becomes an *n*-by-*n* square matrix for an output variable where *n* is the unique number of class values. In addition, accuracy metrics become limited to per-class accuracy rates and the overall classifier accuracy. Table 4.2 lists some of the most commonly used accuracy metrics for classification models.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Table 4.2 Common Accuracy Metrics for Classification Models

Metric	Description
$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$	The ratio of correctly classified instances (positives and negatives) divided by the total number of instances
$True\ Positive\ Rate = \frac{TP}{TP + FN}$ *"Sensitivity" or "Recall"*	The ratio of correctly classified positives divided by the total positive count (i.e., recall or sensitivity)
$True\ Negative\ Rate = \frac{TP}{TP + FN}$ *"Specificity"*	The ratio of correctly classified negatives divided by the total negative count (i.e., false-alarm rate or specificity)
$Precision = \frac{TP}{TP + FP}$	The ratio of correctly classified positives divided by the sum of correctly classified positives and incorrectly classified positives
$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$	The harmonic mean of precision and recall that produces a numeric value between 0 and 1
$Overall\ Accuracy = \frac{\sum True\ Classification_i}{Number\ of\ Cases}$ *"Multiclass classification"*	Overall classification accuracy for a multiclass classification problem
$True\ Classification\ Rate_i = \frac{True\ Classification_i}{\sum False\ Classification_i}$ *"Multiclass classification"*	Classification accuracy for class *i* for a multiclass problem

Estimating the accuracy of a classification model (or classifier) induced by a supervised learning algorithm is important for two reasons. First, the estimation results can be used to assess the future prediction accuracy, which could imply the level of confidence in the classifier's output in the actual prediction system. Second, the estimation results can be used for choosing a classifier from a given

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

set (identifying the "best" classification model among the many trained).

The following sections describe some of the most popular cross-validation methodologies used for the assessment of classification-type prediction models.

Simple Split

The simple split methodology (or holdout methodology or test sample estimation) partitions data into two mutually exclusive subsets called a training set and a test set (or holdout set). It is common to designate two-thirds of the data as the training set and the remaining one-third as the test set. The training set is used by the inducer (i.e., model builder), and the built classifier is then tested on the test set. An exception to this rule occurs when the classifier is an artificial neural network. In that case, the data is partitioned into three mutually exclusive subsets: training, validation, and testing sets. The validation set is used during model building to prevent overfitting. (For more on artificial neural networks, see Chapter 5.) Figure 4.5 illustrates the simple (or single) split methodology.

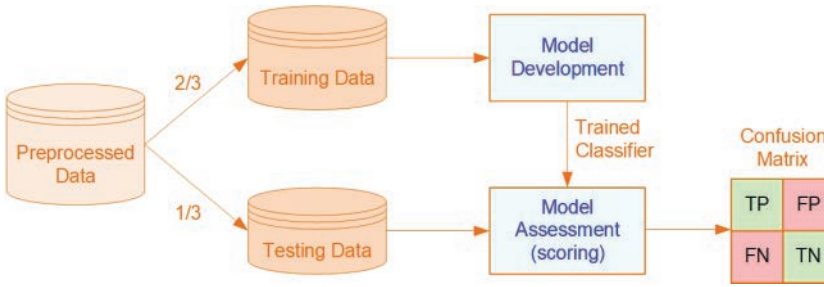


Figure 4.5 Simple Random Data Splitting for Training and Testing

The main criticism of this method is that it makes the assumption that the data in the two subsets is of the same kind (i.e., has exactly the same properties). Because this is a simple random partitioning, in most realistic data sets where the data is skewed on the

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

classification variable, such an assumption may not hold true. In order to improve this situation, stratified sampling is suggested, where the strata become the output variable. Even though this is an improvement over the simple split methodology, it still has a bias associated with the single random partitioning.

k-Fold Cross-Validation

In order to minimize the bias associated with the random sampling of the training and holdout data samples in comparing the predictive accuracy of two or more methods, an analyst can use a methodology called *k*-fold cross-validation. In this methodology, also called *rotation estimation*, the complete data set is randomly split into *k* mutually exclusive subsets of approximately equal size. The classification model is trained and tested *k* times. Each time, it is trained on all but one fold and then tested on the remaining single fold. The cross-validation estimate of the overall accuracy of a model is calculated by simply averaging the *k* individual accuracy measures. This is the formula for this aggregation:

$$Cross\ Validation\ Accuracy = \frac{1}{k}\sum_{i=1}^{k} Accuracy_i \ \left(k = \text{Number of folds}\right)$$

Figure 4.6 provides a graphical illustration of *k*-fold cross-validation, where *k* is set to 10 (i.e., 10-fold cross-validation).

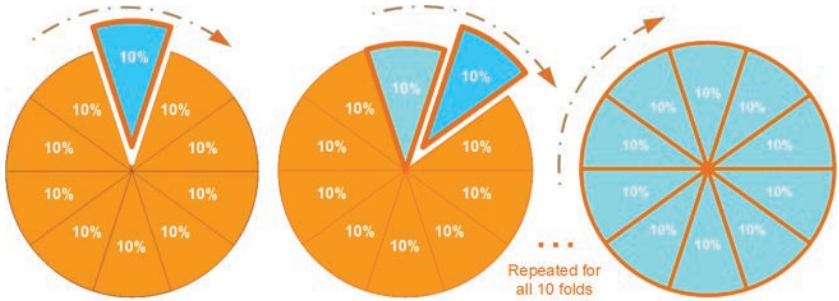


Figure 4.6 A Graphical Illustration of 10-Fold Cross-Validation Methodology

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Area Under the ROC Curve

The area under the ROC (receiver operating characters—a term borrowed from radar detection) curve is a graphical assessment technique for binary classification problems, in which the true positive rate (i.e., sensitivity) is plotted on the *y*-axis, and the false positive rate (1: specificity) is plotted on the *x*-axis. Because AUC (area under the ROC curve) is scale-agnostic/invariant, it can be used to objectively rank multiple prediction methods. Although it is a very good metric for judging how well each prediction model performs and helps rank order the model types, it does not provide an absolute value for the models' predictive accuracy. That is, the AUC value is not to be confused with the overall prediction accuracy of the model. (However, as you will deduce from the experimentations, the AUC value and overall accuracy values will be highly correlated: The model types that produce higher values of AUC will most likely also create higher overall accuracy values.) AUC would be a very good metric to use in a marketing campaign, for instance, because you could use the predictions ranked by probability to create an ordered list of users to whom to send the marketing promotions.

Another benefit of using AUC is that it is classification-threshold invariant; that is, it measures the quality of the model's prediction power, regardless of what classification threshold is chosen. In contrast, F1 score or overall accuracy both are prone to the choice of threshold value. Specifically, AUC determines the overall unbiased predictive power of a classifier, wherein the AUC value 1 indicates a perfect classifier, and the value 0.5 indicates no predictive power (i.e., no better than random chance); however, in reality, the AUC values would range between these two extreme cases, 0.5 and 1.0. For example, in Figure 4.7, ROC curve *A* has a better AUC and hence better classification performance than *B* (the AUC of which is illustrated with the blue shaded area), while *C* is the baseline, representing the ROC curve not any better than random chance. Although in Figure 4.7 the three lines are shown perfectly smooth for purposes

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

of illustration, in real-world applications, the ROC curves look rather rugged and discrete.

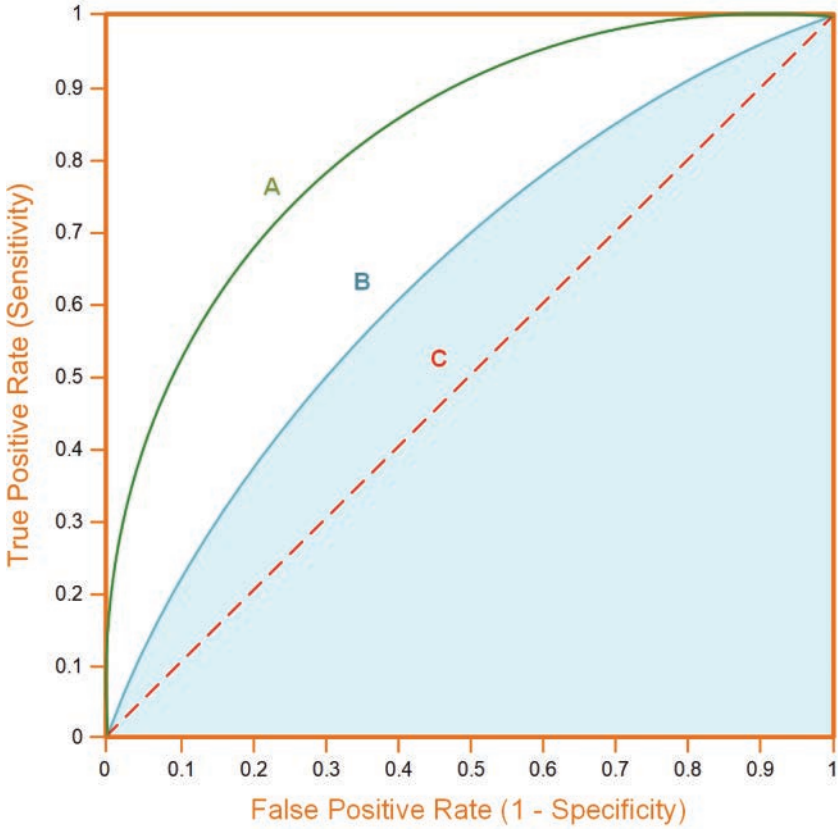


Figure 4.7 An Illustration of Three Areas Under the ROC Curves

Other Cross-Validation Methodologies

Although they are not as common as the techniques described so far in this chapter, some other assessment techniques can also be used for classification-type problems. The following are some of the exemplary ones:

- **Leave-one-out.** The leave-one-out method is similar to a *k*-fold cross-validation, where *k* takes the value 1; that is, every

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

data point is used for testing once on as many models developed as there are data points. This is a time-consuming methodology, but it can be a viable option for small data sets.

- **Bootstrapping.** With bootstrapping, a fixed number of instances from the original data are sampled (with replacement) for training, and the rest of the data set is used for testing. This process is repeated as many times as desired.
- **Jackknifing.** This methodology is similar to the leave-one-out methodology; you can use it to calculate accuracy by leaving out one sample at each iteration of the estimation process.

Classification Methods

Many methods and algorithms are used for classification modeling, including the following:

- **Decision tree analysis.** Decision tree analysis (a machine learning technique) is arguably the most popular classification technique in the data mining arena. The following section provides a detailed description of this technique.
- **Statistical analysis.** For many years—until the emergence of machine learning techniques—statistical techniques were the primary classification algorithms. Statistical classification techniques include *logistic regression* and *discriminant analysis*, both of which make the assumptions that the relationships between the input and output variables are linear in nature, the data is normally distributed, and the variables are not correlated and are independent of each other. The questionable nature of these assumptions has led to the shift toward machine learning techniques.
- **Neural networks.** Using neural networks is among the most popular machine learning techniques that can be used for

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

classification-type problems. A detailed description of this technique is presented in Chapter 5.

- **Support vector machines.** Along with neural networks, support vector machines are becoming increasingly popular as powerful classification algorithms. A detailed description of this technique is presented in Chapter 5.
- **Ensemble models.** Either homogeneous or heterogeneous ensemble models can be used for better predictive ability and robust performance. Chapter 5 provides a detailed description of model ensembles.
- ***k*-nearest-neighbor algorithm.** This deceptively simple yet highly efficient algorithm uses similarity as the basis for its classification method. Chapter 5 provides a detailed description of this technique.
- **Case-based reasoning.** This approach, which is conceptually similar to the nearest-neighbor algorithm, uses historical cases to recognize commonalities in order to assign a new case to the most probable category.
- **Bayesian classifiers.** This approach uses probability theory to build classification models based on past occurrences that are capable of placing a new instance into a most probable class (or category).
- **Genetic algorithms.** Genetic algorithms use the analogy of natural evolution to build directed-search-based mechanisms to classify data samples.
- **Rough sets.** This method takes into account the partial membership of class labels to predefined categories in building models (collection of rules) for classification problems.

Complete coverage of all these classification techniques is beyond the scope of this book, but the following section describes the most

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

popular one, decision trees, and Chapter 5 covers several of the others (the most notable ones).

Decision Trees

Before describing the details of decision trees, we need to discuss some simple terminology. First, decision trees include many input variables that may have an impact on the classification of different patterns. These input variables are usually called *attributes*. For example, if we were to build a model to classify loan risks on the basis of just two characteristics—income and credit rating—these two characteristics would be the attributes, and the resulting output would be the class label (e.g., low, medium, or high risk). Second, a tree consists of branches and nodes. A branch represents the outcome of a test to classify a pattern (on the basis of a test), using one of the attributes. A leaf node at the end represents the final class choice for a pattern (a chain of branches from the root node to the leaf node, which can be represented as a complex if–then statement).

The basic idea behind a decision tree is that it recursively divides a training set until each division consists entirely or primarily of examples from one class. Each nonleaf node of the tree contains a split point, which is a test on one or more attributes and determines how the data is to be divided further. Decision tree algorithms, in general, build an initial tree from the training data such that each leaf node is pure, and they then prune the tree to increase its generalization and, hence, the prediction accuracy of the test data.

In the growth phase, the tree is built by recursively dividing the data until each division is either pure (i.e., contains members of the same class) or relatively small. The basic idea is to ask questions whose answers provide the most information, as is done in the game "Twenty Questions."

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The split used to partition the data depends on the type of attribute used in the split. For a continuous attribute A, splits are of the following form:

value(A) *x*

where *x* is some "optimal" split value of A. For example, the split based on income could be Income > 50000. For the categorical attribute A, splits are of the following form:

value(A) belongs to *x*

where *x* is a subset of A. For example, the split could be on the basis of education (whether the individual has a college degree).

A general algorithm for building a decision tree is as follows:

1. Create a root node and assign all the training data to it.
2. Select the best splitting attribute.
3. Add a branch to the root node for each value of the split. Split the data into mutually exclusive (i.e., non-overlapping) subsets along the lines of the specific split of the branches.
4. Repeat steps 2 and 3 for each leaf node until the stopping criteria is reached (e.g., the node is dominated by a single class label).

Many different algorithms have been proposed for creating decision trees. These algorithms differ primarily in terms of the way in which they determine the splitting attribute (and its split values), the order of splitting the attributes (splitting the same attribute only once or many times), the number of splits at each node (binary versus ternary), the stopping criteria, and the pruning of the tree (pre- versus postpruning). Some of the most well-known algorithms are ID3 (followed by C4.5 and C5 as the improved versions of ID3) from machine learning, Classification and Regression Trees (CART) from statistics, and the Chi-squared Automatic Interaction Detector (CHAID) from pattern recognition.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

When building a decision tree, the goal at each node is to determine the attribute and the split point of that attribute that best divides the training records in order to purify the class representation at that node. To evaluate the goodness of the split, some splitting indices have been proposed. Two of the most common ones are the Gini index and information gain. The Gini index is used in the CART and SPRINT (Scalable Parallelizable Induction of Decision Trees) algorithms. Versions of information gain are used in ID3 (and its newer versions, C4.5 and C5).

The Gini index has been used in economics to measure the diversity of a population. The same concept can be used to determine the purity of a specific class as a result of a decision to branch along a particular attribute or variable. The best split is the one that increases the purity of the sets resulting from a proposed split.

Information gain is the splitting mechanism used in ID3, which is perhaps the most widely known decision tree algorithm. It was developed by Ross Quinlan in 1986, and since then Quinlan has evolved this algorithm into the C4.5 and C5 algorithms. The basic idea behind ID3 and its variants is to use a concept called *entropy* in place of the Gini index. Entropy measures the extent of uncertainty or randomness in a data set. If all the data in a subset belongs to just one class, there is no uncertainty or randomness in that data set, so the entropy is zero. The objective of this approach is to build subtrees so that the entropy of each final subset is zero or close to zero.

Cluster Analysis for Data Mining

Cluster analysis is an essential data mining method for classifying items, events, or concepts into common groupings called *clusters*. The method is commonly used in biology, medicine, genetics, social network analysis, anthropology, archaeology, astronomy, character recognition, and even MIS development. As the popularity of data mining has increased, the underlying techniques have been applied

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

to business, especially to marketing. Cluster analysis has been used extensively for fraud detection (both credit card and ecommerce fraud) and market segmentation of customers in contemporary CRM systems. More applications in business continue to be developed as the strength of cluster analysis is recognized and used.

Cluster analysis is an exploratory data analysis tool for solving classification problems. The objective is to sort cases (e.g., people, things, events) into groups, or clusters, so that the degree of association is strong among members of the same cluster and weak among members of different clusters. Each cluster describes the class to which its members belong. An obvious one-dimensional example of cluster analysis is to establish score ranges into which to assign class grades for a college class. This is similar to the cluster analysis problem that the U.S. Treasury faced when establishing new tax brackets in the 1980s. A fictional example of clustering occurs in J. K. Rowling's Harry Potter books. The Sorting Hat determines to which House (e.g., dormitory) to assign first-year students at the Hogwarts School. Another example involves determining how to seat guests at a wedding. For data mining, cluster analysis is important because it may reveal natural similarity patterns in data that were not previously apparent but are sensible and useful once found.

Cluster analysis results may be used to do the following:

- Identify a classification scheme (e.g., types of customers)
- Suggest statistical models to describe populations
- Indicate rules for assigning new cases to classes for identification, targeting, and diagnostic purposes
- Provide measures of definition, size, and change in what were previously broad concepts
- Find typical cases to label and represent classes
- Decrease the size and complexity of the problem space for other data mining methods
- Identify outliers in a specific domain (e.g., rare-event detection)

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

How to Determine the Number of Clusters

Clustering algorithms usually require specifying the number of clusters to find. If this number is not known from prior knowledge, it should be chosen in some way. Unfortunately, there is not an optimal way of calculating what this number is supposed to be. Therefore, several different heuristic methods have been proposed. The following are among the most commonly referenced ones:

- Look at the percentage of variance, expressed as a function of the number of clusters; that is, choose a number of clusters so that adding another cluster would not give much better modeling of the data. Specifically, in graphing the percentage of variance explained by the clusters, there is a point at which the marginal gain will drop (giving an angle in the graph), and this indicates the number of clusters to be chosen.
- Set the number of clusters to $(n/2)^{1/2}$, where n is the number of data points.
- Use the Akaike information criterion (AIC), which is a measure of the goodness of fit (based on the concept of entropy) for determining the number of clusters.
- Use the Bayesian information criterion (BIC), which is a model-selection criterion (based on maximum likelihood estimation) for determining the number of clusters.

Clustering Methods

Cluster analysis involves grouping a set of records (i.e., samples or objects) in such a way that objects in the same group (called a *cluster*) are more similar to each other than to those in other groups (or clusters). Most cluster analysis methods involve the use of a distance measure to calculate the closeness between pairs of items. Popular distance measures include Euclidian distance (the ordinary

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

distance between two points that one would measure with a ruler) and Manhattan distance (also called rectilinear distance, or taxicab distance, between two points). There are a number of cluster analysis techniques, each having a slightly different way of identifying natural grouping of objects or records. Figure 4.8 shows a simple taxonomy of clustering techniques.

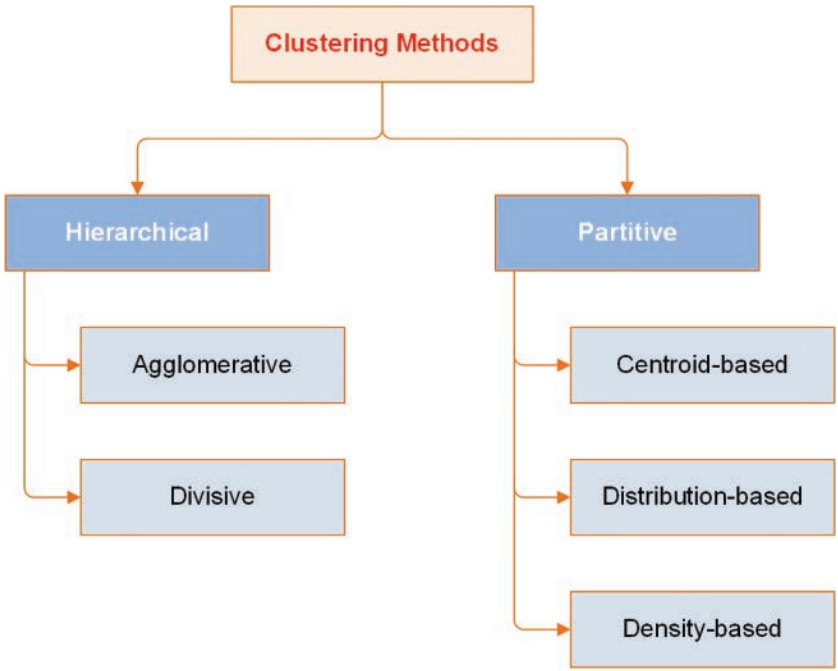


Figure 4.8 A Simple Taxonomy of Clustering Methods

As shown in Figure 4.8, at the highest level, the clustering methods can be classified as hierarchical and partitive. *Hierarchical clustering methods* are based on the basic idea that nearby objects are more related to each other than are those that are farther away. Therefore, a cluster can be defined by the maximum distance needed to connect all related objects. At different distances, different clusters would form, and they can be represented using a dendrogram, which explains where the name *hierarchical clustering* comes from. In hierarchical clustering, the clusters can be identified either from

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

the bottom up (i.e., agglomerative—starting from the bottom, where each object is a cluster of its own and then adding similar objects to create more general clusters) or from the top down (i.e., divisive—starting from the top cluster and then splitting the objects into more specific clusters).

Partitive clustering methods, on the other hand, are based on the idea that there are natural groups of objects in a multidimensional space. These methods follow the process of starting from an arbitrary number of clusters, assigning objects to most probable clusters, and then updating the number of clusters and evolving the assignment of objects to clusters until a stable cluster space is reached. The specific algorithms under partitive clustering use different ways to identify the clusters. Since there is not a single best method or algorithm for clustering, an analyst working with a clustering task should try several competing algorithms, assess their goodness of fit, and identify the best one.

Assessment of a Clustering Technique

How do we measure the goodness of fit of clusters created by a clustering method? How do we know if they are valid? How do we know to what extent they represent the natural grouping of the data? We need measures to compare how well different clustering algorithms perform on a given set of data. We can then choose the "best" one to use for a classification-type data mining problem. The measures for assessing the goodness of fit of clustering methods are often grouped into two categories: internal assessment and external assessment.

Internal Assessment

Internal assessment methods usually assign the best assessment score to the method (or algorithm) that produces clusters with high similarity within a cluster and low similarity between clusters. The

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

internal evaluation measures are best suited to situations in which one algorithm performs better than another, but this may not imply that one algorithm produces more "valid" results than another. Validity is different from comparative assessment because it suggests that the identified cluster structure truly exists in the data set. This assessment method measures intraclass and interclass similarities using some type of distance measure in a multidimensional feature space.

One drawback of using internal criteria in cluster evaluation is that high scores on an internal measure do not necessarily result in effective information-retrieval applications. In addition, this evaluation is biased toward algorithms that use the same cluster model. For example, *k*-means clustering naturally optimizes object distances, and a distance-based internal criterion will likely overrate the resulting clustering.

External Assessment

Whereas in internal evaluation, clustering results are evaluated based on similarity measures, in external evaluation, the cluster results are measured based on data (with known class labels) that was not used for clustering. This assessment consists of a set of preclassified records, with known classes or class labels created by domain experts. Thus, the comparison sets can be thought of as a gold standard for evaluation. These types of evaluation methods measure how close the clustering is to the predetermined labeled classes. As logical as it may sound, widespread practicality of this method is questionable. Finding prelabeled classes for real-world data sets is difficult and may even be impossible. Also, because the true classes contain some type of natural structure, the attributes/variables presented to the clustering algorithm may not be complete enough to allow separation of true clusters.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

k-Means Clustering Algorithm

The *k*-means algorithm (where *k* is the predetermined number of clusters) is arguably the most referenced clustering algorithm. It has its roots in traditional statistical analysis. As part of the family of centroid-based partitive clustering methods, the *k*-means algorithm, as its name implies, assigns each data point (e.g., customer, event, object) to the cluster whose center (also called *centroid*) is the nearest. The center is calculated as the average of all the points in the cluster; that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster. The algorithm steps are listed here and shown graphically in Figure 4.9:

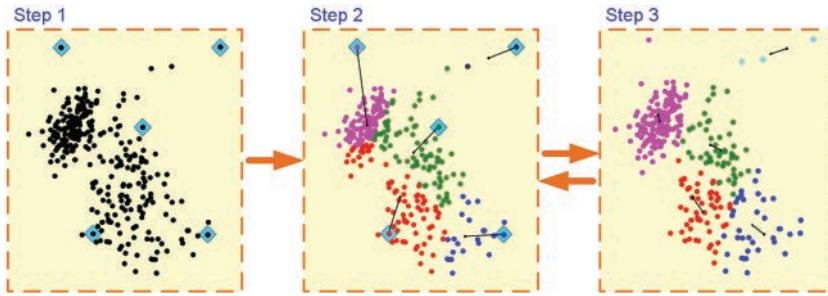


Figure 4.9 Steps in the k-Means Clustering Algorithm

- **Initialization step.** Choose the number of clusters (i.e., the value of *k*).
  1. Randomly generate *k* random points as initial cluster centers.
  2. Assign each point to the nearest cluster center.
  3. Recompute the new cluster centers.
- **Repetition step.** Repeat steps 2 and 3 until some convergence criterion is met (e.g., the assignment of points to clusters becomes stable).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Association

Association—also known as association rule mining, affinity analysis, or market-basket analysis—is a popular data mining method that is commonly used as an example to explain what data mining is and what it can do. You might have heard about the famous (or infamous, depending on how you look at it) relationship discovered between the sales of beer and diapers at grocery stores. As the story goes, a large supermarket chain (maybe Walmart, maybe not; there is no consensus on which supermarket chain it was) did an analysis of customers' buying habits and found a statistically significant correlation between purchases of beer and purchases of diapers. It was theorized that the reason for this was that fathers (presumably young men) were stopping off at the supermarket to buy diapers for their babies (especially on Thursdays), and since they could no longer go down to the sports bar as often, they would buy beer as well. As a result of this finding, the supermarket chain is alleged to have placed the diapers next to the beer, resulting in increased sales of both.

In essence, association rule mining aims to find interesting relationships (affinities) between variables (items) in large databases. Because of its successful application to retail business problems, it is commonly called market-basket analysis. The main idea in market-basket analysis is to identify strong relationships among different products (or services) that are usually purchased together (i.e., show up in the same basket together, either a physical basket at a grocery store or a virtual basket at an ecommerce website). For example, 65% of those who buy comprehensive automobile insurance also buy health insurance; 80% of those who buy books online also buy music online; 60% of those who have high blood pressure and are overweight have high cholesterol; 70% of the customers who buy laptop computer and virus protection software also buy extended service plans.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The input to market-basket analysis is simple point-of-sale transaction data, where a number of products and/or services purchased together (just like the content of a purchase receipt) are tabulated under a single transaction instance. The outcome of the analysis is invaluable information that can be used to better understand customer purchase behavior in order to maximize the profit from business transactions. A business can take advantage of such knowledge by (1) putting the items next to each other to make it more convenient for the customers to pick them up together and not forget to buy one when buying the others (increasing sales volume); (2) promoting the items as a package (e.g., not putting one on sale if the other is on sale); and (3) placing the items apart from each other so that the customer has to walk the aisles to search for them and may, in the process, see and buy other items.

Applications of market-basket analysis include cross-marketing, cross-selling, store design, catalog design, ecommerce site design, optimization of online advertising, product pricing, and sales/promotion configuration. In essence, market-basket analysis helps businesses infer customer needs and preferences from their purchase patterns. Outside the business realm, association rules can be used to discover relationships between symptoms and illnesses, between diagnosis and patient characteristics and treatments (which can be used in medical decision support systems [DSSs]), and between genes and their functions (which can be used in genomics projects), among others. Here are a few common areas and uses for association rule mining:

- **Sales transactions.** Knowledge of combinations of retail products purchased together can be used to improve product placement on the sales floor (e.g., placing products that go together in close proximity) and promotional pricing of products (e.g., not having promotions on two products that are often purchased together).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

- **Credit card transactions.** Knowledge of items purchased with a credit card can provide insight into other products the customer is likely to purchase or fraudulent use of a credit card number.
- **Banking services.** The sequential patterns of services used by customers (e.g., checking account followed by savings account) can be used to identify other services they may be interested in (e.g., investment account).
- **Insurance service products.** Bundles of insurance products bought by customers (e.g., car insurance followed by homeowner's insurance) can be used to propose additional insurance products (e.g., life insurance); on the other hand, unusual combinations of insurance claims can signal fraud.
- **Telecommunication services.** Commonly purchased groups of options (e.g., Internet service, television, and phone) help better structure product bundles and related promotional offers to maximize revenue.
- **Medical records.** Certain combinations of conditions can indicate increased risk of various complications; or certain treatment procedures at certain medical facilities can be tied to certain types of infection.

A good question to consider with respect to the patterns/relationships that association rule mining can discover is "Are all association rules interesting and useful?" In order to answer such a question, association rule mining uses three common metrics: support, confidence, and lift. Before defining these terms, let us look at what an association rule looks like. For example, in the following rule, X (products and/or service; called the left-hand side [LHS], or antecedent) is associated with Y (products and/or service; called the right-hand side [RHS], or consequent):

$$X \Rightarrow Y[S(\%), C(\%)]$$

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

S is the support, and *C* is the confidence for this particular rule. Here are the simple formulas for support, confidence, and lift:

$$Support = Support(X \Rightarrow Y) = \frac{Number\ of\ Baskets\ That\ Contain\ Both\ X\ and\ Y}{Total\ Number\ of\ Baskets}$$

$$Confidence = Confidence(X \Rightarrow Y) = \frac{Supp(X \Rightarrow Y)}{Supp(X)}$$

$$Lift(X \Rightarrow Y) \frac{Conf(X \Rightarrow Y)}{Expected\ Conf(X \Rightarrow Y)} = \frac{\frac{S(X \Rightarrow Y)}{S(X)}}{\frac{S(X) * S(Y)}{S(X)}} = \frac{S(X \Rightarrow Y)}{S(X) * S(Y)}$$

The support (*S*) of a collection of products and/or services is the measure of how often those products and/or services (i.e., LHS + RHS = laptop computer, antivirus software, and extended service plan) appear together in the same transaction—that is, the proportion of transactions in the data set that contain all the products and/or services mentioned in a specific rule. In this example, 30% of all transactions in the hypothetical store database had all three products present on a single sales ticket. The confidence of a rule is the measure of how often the products and/or services on the RHS (consequent) go together with the products and/or services on the LHS (antecedent)—that is, the proportion of transactions that include the LHS while also including the RHS. In other words, it is the conditional probability of finding the RHS of the rule present in transactions where the LHS of the rule already exists. The lift value of an association rule is the ratio of the confidence of the rule and the expected confidence of the rule. The expected confidence of a rule is defined as the product of the support values of the LHS and the RHS divided by the support of the LHS.

Several algorithms are available for discovering association rules. Some well-known algorithms include Apriori, Eclat, and FP-Growth. These algorithms do only half the job: they identify the frequent item sets in the database. Once the frequent item sets are identified, they

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

need to be converted into rules with antecedent and consequent parts. Determination of the rules from frequent item sets is a straightforward matching process, but the process may be time-consuming with large transaction databases. Even though there can be many items in each section of the rule, in practice the consequent part usually contains a single item. The following section explains one of the most popular algorithms for identification of frequent item sets.

Apriori Algorithm

The Apriori algorithm is the algorithm most commonly used to discover association rules. Given a set of item sets (e.g., sets of retail transactions, each listing individual items purchased), the algorithm attempts to find subsets that are common to at least a minimum number of the item sets (i.e., comply with a minimum support). The Apriori algorithm uses a bottom-up approach, where frequent subsets are extended one item at a time (a method known as *candidate generation*, whereby the size of frequent subsets increases from one-item subsets to two-item subsets, then three-item subsets, etc.), and groups of candidates at each level are tested against the data for minimum support. The algorithm terminates when no further successful extensions are found.

As an illustrative example, consider the following. A grocery store tracks sales transactions by SKU (stock-keeping unit) and thus knows which items are typically purchased together. Figure 4.10 shows the database of transactions, along with the subsequent steps in identifying the frequent item sets. Each SKU in the transaction database corresponds to a product, such as "1 = butter," "2 = bread," "3 = water," and so on. The first step in the Apriori algorithm is to count up the frequencies (i.e., the supports) of each item (one-item item sets). For this overly simplified example, let us set the minimum support to 3 (or 50%, meaning an item set is considered to be a frequent item set if it shows up in at least 3 out of 6 transactions in the database). Because all

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

the one-item item sets have at least 3 in the support column, they are all considered frequent item sets. However, if any of the one-item item sets were not frequent, they would not have been included as possible members of possible two-item pairs. In this way, Apriori prunes the tree of all possible item sets. As Figure 4.10 shows, using one-item item sets, all possible two-item item sets are generated, and the transaction database is used to calculate their support values. Because the two-item item set {1, 3} has a support value less than 3, it should not be included in the frequent item sets that will be used to generate the next-level item sets (three-item item sets). The algorithm seems deceptively simple—but only for small data sets. In much larger data sets, especially those with huge numbers of items present in low quantities and small numbers of items present in big quantities, the search and calculation become computationally intensive.

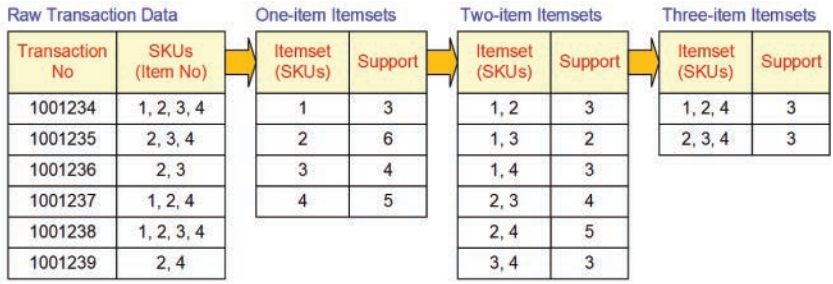


Figure 4.10 The Process of Identifying Frequent Item Sets with the Apriori Algorithm

Data Mining and Predictive Analytics Misconceptions and Realities

Data mining is a powerful analytical tool that enables business executives to advance from describing the nature of the past to predicting the future. It helps marketers find patterns that unlock the mysteries of customer behavior. The results of data mining can be used to increase revenue, reduce expenses, identify fraud, and locate business opportunities, offering a whole new realm of competitive advantage.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

As an evolving and maturing field, data mining is often associated with a number of misconceptions, including the ones listed in Table 4.3.

Table 4.3 A Few Misconceptions and Realities About Data Mining

Misconception	Reality
Data mining provides instant, crystal ball–like predictions.	Data mining is a multistep process that requires deliberate, proactive design and use.
Data mining is not yet viable for business applications.	The current state-of-the-art is ready to go for almost any business.
Data mining requires a separate, dedicated database.	Because of advances in database technology, a dedicated database is not required, even though it may be desirable.
Only those with advanced degrees can do data mining.	Web-based tools enable managers of all educational levels to do data mining.
Data mining is only for large firms that have lots of customer data.	If its data accurately reflects the business or its customers, a company can use data mining.

Data mining visionaries have gained enormous competitive advantage by understanding that these misconceptions are, in fact, misconceptions.

The following 10 data mining mistakes are often made in practice, and you should try to avoid them:

1. Selecting the wrong problem for data mining.
2. Ignoring what your sponsor thinks data mining is and what it really can and cannot do.
3. Leaving insufficient time for data preparation. The data preparation stage takes more effort than is generally understood.
4. Looking only at aggregated results and not at individual records. IBM's DB2 IMS can highlight individual records of interest.
5. Being sloppy about keeping track of the data mining procedure and results.
6. Ignoring suspicious findings and quickly moving on.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

7. Running mining algorithms repeatedly and blindly. It is important to think hard about the next stage of data analysis. Data mining is a very hands-on activity.
8. Believing everything you are told about the data.
9. Believing everything you are told about your own data mining analysis.
10. Measuring your results differently from the way your sponsor measures them.

Application Example: Predicting NCAA Bowl Game Results

Predicting the outcome of a college football game (or any sports game, for that matter) is an interesting and challenging problem. Challenge-seeking researchers from both academics and industry have spent a great deal of effort on forecasting the outcomes of sporting events. Large quantities of historic data exist in different media outlets (and are often publicly available) regarding the structure and outcomes of sporting events, in the form of a variety of numerically or symbolically represented factors that are assumed to contribute to those outcomes. However, despite the large quantity of studies in sports (more than 43,000 hits on digital literature databases), a small percentage of papers have focused exclusively on the characteristics of sports forecasts. Instead, many papers have been written about the efficiency of sports markets. Since most previous betting-markets studies have been concerned with economic efficiency (Van Bruggen et al., 2010), they have not evaluated the actual (or implied) forecasts associated with those events. As it turns out, it is possible to derive a considerable amount of information about the forecasts and the forecasting process from studies that have tested the markets for economic efficiency (Stekler et al., 2010).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The end-of-season bowl games are very important to colleges both financially (bringing in millions of dollars of additional revenue) and reputationally (for recruiting quality students and highly regarded high school athletes for their athletic programs). Teams that are selected to compete in a given bowl game split a purse, the size of which depends on the specific bowl. Some bowls are more prestigious and have higher payouts for the two teams. Therefore, securing an invitation to a bowl game is a big goal of any division I-A college football program. The bowl game decision makers are given the authority to select and invite bowl-eligible (a team that has six wins against its Division I-A opponents in that season) successful teams (as per the ratings and rankings) that will play in an exciting and competitive game, attract fans of both schools, and keep the remaining fans tuned in via a variety of media outlets for advertising.

In one data mining study, Delen et al. (2012) used eight years of bowl game data along with three popular data mining techniques (i.e., decision trees, neural networks, and support vector machines) to predict both the classification type outcome of a game (win versus loss) and the regression type outcome (i.e., projected point difference between the scores of the two opponents). What follows is a brief description of their study.

Methodology

In this research, Delen and his colleagues used the popular data mining methodology CRISP-DM (Cross-Industry Standard Process for Data Mining), which is a six-step process: (1) understand the domain and develop the goals for the study; (2) identify, access, and understand the relevant data sources; (3) preprocess, clean, and transform the relevant data; (4) develop models using comparable analytical techniques; (5) evaluate and assess the validity and utility of the models against each other and against the goals of the study; and (6) deploy the models for use in decision-making processes.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

This popular methodology, which is covered in detail in Chapter 3, provided Delen et al. with a systematic and structured way to conduct the underlying data mining study and, hence, improved the likelihood of obtaining accurate and reliable results.

To objectively assess the predictive power of the different model types, Delen et al. used a cross-validation methodology, which is a popular statistical technique that is often used in data mining to compare multiple models in terms of their predictive accuracies. Traditional cross-validation methodology splits the data into two mutually exclusive subsets: training and testing (or three, in the case of neural networks that include validation sets). This type of single random split may lead to non-homogeneous representation in the subsets, especially in a small data set. In order to minimize the bias associated with the random sampling of the training and testing data samples, Delen et al. used *k*-fold cross-validation (discussed earlier in this chapter). Figure 4.11 provides a graphical representation of the methodology the researchers used.

Data

The sample data for this study was collected from a variety of sports databases available online, including ESPN.com, Covers.com, ncaa.org, and rauzulusstreet.com. The data set included 244 bowl games, representing a complete set of eight seasons of college football bowl games played between 2002 and 2009. It also included an out-of-sample data set (2010–2011 bowl games) for additional validation purposes. Exercising one of the popular data mining rules of thumb, the authors included as much relevant information in the model as possible. Therefore, after an in-depth variable identification and collection process, they ended up with a data set that included 36 variables. The first 6 variables were the identifying variables: name and year of the bowl game, home and away team names, and their athletic conferences (see variables 1–6 in

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Table 4.4). The next 28 were input variables, which included variables delineating a team's seasonal statistics on offense and defense, game outcomes, team composition characteristics, athletic conference characteristics, and how they fared against the odds (see variables 7–34 in Table 4.4). The last 2 variables were the output variables: ScoreDiff (i.e., the score difference between the home team and the away team represented with an integer number) and WinLoss (i.e., whether the home team won or lost the bowl game, represented with a nominal label).

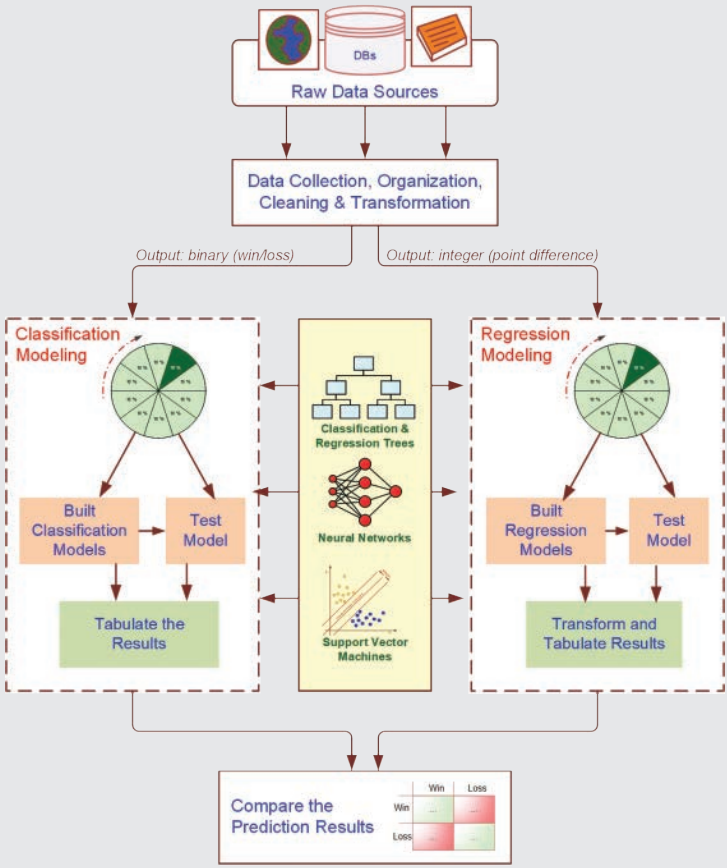


Figure 4.11 Graphical Illustration of the Methodology Employed in the Study

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

In the formulation of the data set, each row (a.k.a. tuple, case, sample, or example) represented a bowl game, and each column stood for a variable (i.e., identifier/input or output type). In order to represent the game-related comparative characteristics of the two opponent teams, in the input variables, the authors calculated and used the differences between the measures of the home and away teams. All these variable values were calculated from the home team's perspective. For instance, the variable PPG (average number of points a team scored per game) represents the difference between the home team's PPG and the away team's PPG. The output variables represent whether the home team is expected to win or lose the bowl game. That is, if the ScoreDiff variable takes a positive integer number, then the home team is expected to win the game by that margin; if the ScoreDiff variable takes a negative integer number, then the home team is expected to lose the game by that margin. In the case of WinLoss, the value of the output variable is a binary label, Win or Loss, indicating the outcome of the game for the home team.

Table 4.4 Description of the Variables Used in the Study

No	Cat	Variable Name	Description
1	ID	YEAR	Year of the bowl game
2	ID	BOWL GAME	Name of the bowl game
3	ID	HOMETEAM	Home team (as listed by the bowl organizers)
4	ID	AWAYTEAM	Away team (as listed by the bowl organizers)
5	ID	HOMECONFERENCE	Conference of the home team
6	ID	AWAYCONFERENCE	Conference of the away team
7	I1	DEFPTPGM	Defensive points per game
8	I1	DEFRYDPGM	Defensive rush yards per game
9	I1	DEFYDPGM	Defensive yards per game
10	I1	PPG	Average number of points a given team scored per game

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

No	Cat	Variable Name	Description
11	I1	PYDPGM	Average total pass yards per game
12	I1	RYDPGM	Team's average total rush yards per game
13	I1	YRDPGM	Average total offensive yards per game
14	I2	HMWIN%	Home winning percentage
15	I2	LAST7	How many games the team won out of their past 7 games
16	I2	MARGOVIC	Average margin of victory
17	I2	NCTW	Non-conference team winning percentage
18	I2	PREVAPP	Whether the team appeared in a bowl game previous year
19	I2	RDWIN%	Road winning percentage
20	I2	SEASTW	Winning percentage for the year
21	I2	TOP25	Winning percentage against AP top 25 teams for the year
22	I3	TSOS	Strength of schedule for the year
23	I3	FR%	Percentage of games played by freshman class players for the year
24	I3	SO%	Percentage of games played by sophomore class players for the year
25	I3	JR%	Percentage of games played by junior class players for the year
26	I3	SR%	Percentage of games played by senior class players for the year
27	I4	SEASOvUn%	Percentage of times a team went over the O/U* in the current season
28	I4	ATSCOV%	Against the spread cover percentage of the team in previous bowl games
39	I4	UNDER%	Percentage of times a team went under in previous bowl games
30	I4	OVER%	Percentage of times a team went over in previous bowl games
31	I4	SEASATS%	Percentage of covering against the spread for the current season
32	I5	CONCH	Did the team win their respective conference championship game

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

No	Cat	Variable Name	Description
33	I5	CONFSOS	Conference strength of schedule
34	I5	CONFWIN%	Conference winning percentage
35	O1	ScoreDiff**	Score difference (HomeTeamScore – AwayTeamScore)
36	O2	WinLoss**	Whether the home team wins or loses the game

* Over/Under: Whether a team will go over or under the expected score difference
** Output variables: ScoreDiff for regression models and WinLoss for binary classification models
I1: Offense/defense; I2: Game outcome; I3: Team configuration; I4: Against the odds; I5: Conference stats
ID: Identifier variables; O1: Output variable for regression models; O2: Output variable for classification models

Methods

In this study, three popular prediction techniques were used to build models (and to compare them to each other): artificial neural networks, decision trees, and support vector machines. These prediction techniques were selected based on their ability to model both classification- and regression-type prediction problems and their popularity in recently published data mining literature. More details about these popular data mining methods can be found in Chapter 5.

Evaluation

In order to compare the predictive accuracy of all models to one another, the researchers used a stratified k-fold cross-validation methodology. In a stratified version of k-fold cross-validation, the folds are created such that they contain approximately the same proportion of predictor labels (i.e., classes) as the original data set. In this study, the value of k was set to 10 (i.e., the complete set of 244 samples was split into 10 subsets, each having about 25 samples), which is a common practice in predictive data mining applications. A graphical depiction of the 10-fold cross-validation is shown earlier in this chapter (refer Figure 4.11). To compare the prediction

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

models that were developed using the previously mentioned three data mining techniques, the researchers chose to use three common performance criteria: accuracy, sensitivity, and specificity. The simple formulas for these metrics were also explained earlier in this chapter.

Results

The prediction results of the three modeling techniques are presented in Table 4.5 and Table 4.6. Table 4.5 presents the 10-fold cross-validation results of the classification methodology where the three data mining techniques are formulated to have a binary-nominal output variable (i.e., WinLoss). Table 4.6 presents the 10-fold cross-validation results of the regression-based classification methodology where the three data mining techniques are formulated to have a numeric output variable (i.e., ScoreDiff). In the regression-based classification prediction, the numeric output of the models is converted to a classification type by labeling the positive WinLoss numbers with Win and negative WinLoss numbers with Loss and then tabulating them in the confusion matrices. Using the confusion matrices, the overall prediction accuracy, sensitivity, and specificity of each model type are calculated and presented in these two tables.

As the results indicate, the classification-type prediction methods performed better than regression-based classification-type prediction methodologies. Among the three data mining technologies, classification and regression trees produced better prediction accuracy in both prediction methodologies. Overall, classification and regression tree classification models produced a 10-fold cross-validation accuracy of 86.48%, followed by support vector machines (with a 10-fold cross-validation accuracy of 79.51%) and neural networks (with a 10-fold cross-validation accuracy of

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

75.00%). Using a *t*-test, the researchers found that these accuracy values were significantly different at the 0.05 alpha level; that is, the decision tree was a significantly better predictor of this domain than were the neural network and support vector machine; the support vector machine was a significantly better predictor than the neural networks.

Table 4.5 Prediction Results for the Direct Classification Methodology

Prediction Method (Classification*)		Confusion Matrix		Accuracy** (in %)	Sensitivity (in %)	Specificity (in %)
		Win	Loss			
ANN (MLP)	Win	92	42	75.00	68.66	82.73
	Loss	19	91			
SVM (RBF)	Win	105	29	79.51	78.36	80.91
	Loss	21	89			
DT (C&RT)	Win	113	21	**86.48**	84.33	89.09
	Loss	12	98			

* The output variable is a binary categorical variable (Win or Loss).
** Differences were significant ($p < 0.01$).

Table 4.6 Prediction Results for the Regression-Based Classification Methodology

Prediction Method (Regression-Based*)		Confusion Matrix		Accuracy** (in %)	Sensitivity (in %)	Specificity (in %)
		Win	Loss			
ANN (MLP)	Win	94	40	72.54	70.15	75.45
	Loss	27	83			
SVM (RBF)	Win	100	34	74.59	74.63	74.55
	Loss	28	82			
DT (C&RT)	Win	106	28	77.87	76.36	79.10
	Loss	26	84			

* The output variable is a numeric/integer variable (ScoreDiff).
** Differences were significant ($p < 0.01$).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The results of the study showed that the classification-type models predicted the game outcomes better than did regression-based classification models. Even though these results are specific to the application domain and the data used in this study—and therefore should not be generalized beyond the scope of the study—they are exciting because decision trees are not only better predictors but also better for understanding and deployment, compared to the other two machine learning techniques employed in this study. More details about this study can be found in Delen et al. (2012).

Summary

Analytics in general, and predictive analytics in specific, rely on data. Data is the raw material for the knowledge nuggets and actionable insights discovered via the data mining process. This chapter starts with a thorough description of data, presenting a taxonomy and related definition of different data types, and explains the typical steps involved in the preprocessing of data. The output of the preprocessing steps is the analytics-ready data set.

In addition to the data and the process involved in the preparation of it, this chapter provides a comprehensive view of the methods used in data mining. Employing a taxonomical approach, this chapter explains the three main categories of patterns discovered in data mining: prediction, association, and clustering. Also covered in this chapter are the most commonly used assessment methods and cross-validation methodologies for classification-type prediction models.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

References

Bhandari, I., E. Colet, J. Parker, Z. Pines, et al. (1997). "Advanced Scout: Data Mining and Knowledge Discovery in NBA Data," *Data Mining and Knowledge Discovery*, 1(1): 121–125.

Buck, N. (2000/2001, December/January). "Eureka! Knowledge Discovery," *Software Magazine*.

Chan, P. K., W. Phan, A. Prodromidis, & S. Stolfo. (1999). "Distributed Data Mining in Credit Card Fraud Detection," *IEEE Intelligent Systems*, 14(6): 67–74.

Chapman, P., J. Clinton, R. Kerber, T. Khabaza, et al. (2013). *CRISP-DM 1.0*. www.the-modeling-agency.com/crisp-dm.pdf (accessed July 2020).

Davenport, T. H. (2006, January). "Competing on Analytics," *Harvard Business Review*. https://hbr.org/2006/01/competing-on-analytics (accessed September 2020).

Delen, D., D. Cogdell, & N. Kasap. (2012). "A Comparative Analysis of Data Mining Methods in Predicting NCAA Bowl Outcomes," *International Journal of Forecasting*, 28: 543–552.

Delen, D., R. Sharda, & P. Kumar. (2007). "Movie Forecast Guru: A Web-Based DSS for Hollywood Managers," *Decision Support Systems*, 43(4): 1151–1170.

Nemati, H. R., & C. D. Barko. (2001). "Issues in Organizational Data Mining: A Survey of Current Practices," *Journal of Data Warehousing*, 6(1): 25–36.

Quinlan, J. R. (1986). "Induction of Decision Trees," *Machine Learning*, 1: 81–106.

Sharda, R., & D. Delen. (2006). "Predicting Box-Office Success of Motion Pictures with Neural Networks," *Expert Systems with Applications*, 30: 243–254.

Stekler, H. O., D. Sendor, & R. Verlander. (2010). "Issues in Sports Forecasting," *International Journal of Forecasting*, 26: 606–621.

Van Bruggen, G. H., M. Spann, G. L. Lilien, & B. Skiera. (2010). "Prediction Markets as Institutional Forecasting Support Systems," *Decision Support Systems*, 49: 404–416.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Wald, M. L. (2004, February 21). "U.S. Calls Release of JetBlue Data Improper," *The New York Times*.

Wilson, R., & R. Sharda. (1994). "Bankruptcy Prediction Using Neural Networks," *Decision Support Systems*, 11: 545–557.

Wright, C. (2012). *Statistical Predictors of March Madness: An Examination of the NCAA Men's Basketball Championship*. http://economics-files.pomona.edu/GarySmith/Econ190/Wright March Madness Final Paper.pdf (accessed July 2020).

Zaima, A. (2003). "The Five Myths of Data Mining," *What Works: Best Practices in Business Intelligence and Data Warehousing*, 15: 42–43.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

5

Algorithms for Predictive Analytics

Predictive analytics algorithms are mostly derived from either traditional statistical methods or from contemporary machine learning techniques. According to a wealth of comparative studies published in the literature, while statistical methods are well established and theoretically sound, machine learning techniques are more accurate, informative, and actionable/practical. The statistical methods that have had the biggest impact in the evolution of predictive analytics and data mining include discriminant analysis, linear regression, and logistic regression. The most popular machine learning techniques used in numerous successful predictive analytics projects include decision trees, *k*-nearest neighbor, artificial neural networks, and support vector machines. All of these machine learning techniques can handle both classification as well as regression-type prediction problems. Often, they are applied to complex prediction problems where other, more traditional, techniques are not capable of producing satisfactory results.

Because of the popularity of machine learning techniques in the predictive analytics literature, in this chapter, we explain some of the most common ones in detail but without getting too technical or algorithmic. We also briefly explain other mentioned statistical techniques—linear and logistic regression and time-series analysis—to provide balanced coverage of the algorithmic spectrum of predictive analytics.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Naive Bayes

Naive Bayes is a simple probability-based classification method derived from the well-known Bayes' theorem. The method requires output variables to have nominal/categorical values. Although the input variables can be a mix of numeric and nominal types, the numeric output variable needs to be discretized via some type of binning method before it can be used in the Naive Bayes classifier. The "Naive" part of the name Naive Bayes comes from the strong, somewhat unrealistic assumption of independence among the input variables. Simply put, a Naive Bayes classifier assumes that the input variables do not depend on each other, and the presence (or absence) of a particular variable in the mix of the predictors does not have anything to do with the presence or absence of any other variables.

Naive Bayes classification models can be developed very efficiently (i.e., rather rapidly with very little computational effort) and effectively (i.e., quite accurately) in a supervised machine learning environment. That is, using a set of training data (which may not be very large), the parameters for Naive Bayes classification models can be obtained using the maximum likelihood method. In other words, because of the independence assumption, we can develop Naive Bayes models without strictly complying with all the rules and requirements of Bayes' theorem. First, let us review the Bayes' theorem.

Bayes' Theorem

In order to appreciate the Naive Bayes classification method, it is important to understand the basic idea of Bayes' theorem and the exact Bayesian classifier (the one without the strong naive independence assumption). Bayes' theorem (also called Bayes' rule), named after the British mathematician Thomas Bayes (1701–1761), is a mathematical formula for determining conditional probabilities. In this formula, shown shortly, Y denotes the hypothesis and X denotes the data/evidence. This vastly popular theorem provides a way to revise and improve prediction probabilities by using additional evidence.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The following formulas show the relationship between the probabilities of two events Y and X. $P(Y)$ is the prior probability of Y. It is "prior" in the sense that it does not take into account any information about X. $P(Y|X)$ is the conditional probability of Y, given X. It is also called the posterior probability because it is derived from (or depends on) the specified value of X. $P(X|Y)$ is the conditional probability of X given Y. It is also called the likelihood. $P(X)$ is the prior probability of X, which is also called the evidence and acts as the normalizing constant.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \rightarrow Posterior = \frac{Likelihood \times Prior}{Evidence}$$

$P(Y|X)$: *Posterior Probability of Y Given X*

$P(X|Y)$: *Conditional Probability of X Given Y* (*Likelihood*)

$P(Y)$: *Posterior Probability of Y*

$P(X)$: *Prior Probability of X* (*Evidence, or Unconditional Probability, of X*)

To numerically illustrate these formulas, let us look at a simple example. Say that, based on the weather report, we know that there is a 40% chance of rain on Saturday. From the historical data, we also know that if it rains on Saturday, there is a 10% chance it will also rain on Sunday; and if it doesn't rain on Saturday, there is an 80% chance it will rain on Sunday. Let us say that "Raining on Sunday" is event Y, and "Raining on Monday" is event X. Based on the description, we can write the following:

$P(Y)$ = Probability of Raining on Saturday = 0.40

$P(X|Y)$ = Probability of Raining on Sunday if It Rained on Saturday = 0.10

$P(X)$ = Probability of Raining on Monday = Sum of the Probability of "Raining on Saturday and Raining on Sunday" and "Not Raining on Saturday and Raining on Sunday" = $0.40 \times 0.10 + 0.60 \times 0.80 = 0.52$

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Now if we were to calculate the probability for "It Rained on Saturday?" given "Rained on Sunday," we would use Bayes' theorem, which would allow us to calculate the probability of an earlier event, given the result of a later event:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} = \frac{0.10 \times 0.40}{0.52} = 0.0769$$

Therefore, in this example, if it rained on Sunday, there's a 7.69% chance it rained on Saturday.

Naive Bayes Classifier

The Bayes classifier uses Bayes' theorem without the simplifying strong independence assumption. In a classification-type prediction problem, the Bayes classifier works as follow: Given a new sample to classify, it finds all other samples exactly like it (i.e., all predictor variables having the same values as the sample being classified), determines the class labels that they all belong to, and classifies the new sample into the most representative class. If none of the samples has the exact value match with the new class, then the classifier will fail in assigning the new sample into a class label (because it could not find any strong evidence to do so). Here is a very simple example. Using the Bayes classifier, say that we want to decide whether to play golf (Yes or No) for the following situation (Outlook is Sunny, Temperature is Hot, Humidity is High, and Windy is No).

Process of Developing Naive Bayes Classifier

Similar to other machine learning methods, Naive Bayes employs a two-phase model development and scoring/deployment process. The first phase is training, where the model/parameters are estimated, and the second phase is testing, where the classification/prediction is performed on new cases. The process is described in the following sections.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Training Phase

The following steps are involved in the training phase:

Step 1. Obtain the data, clean the data, and organize it in a flat file format (i.e., columns as variables and rows as cases).

Step 2. Make sure the variables are all nominal variables. If any of the variables is numeric/continuous, then the numeric variables need to go through a data transformation; that is, you need to convert a numeric variable into a nominal variable by using discretization, such as binning.

Step 3. Calculate the prior probability of all class labels for the dependent variable.

Step 4. Calculate the likelihood of all predictive variables and their possible values with respect to the dependent variable. In the case of mixed variables types (categorical and continuous), the likelihood (conditional probability) of each variable is estimated with the proper method for the specific variable type. Likelihoods for nominal and numeric predictor variables are calculated as follows:

- For categorical variables, the likelihood (i.e., the conditional probability) is estimated as the simple fraction of the training samples for the variable value with respect to the dependent variable.
- For numeric variables, the likelihood is calculated by (1) calculating the mean and variance for each predictor variable for each dependent variable value (i.e., class) and then (2) calculating the likelihood by using the following formula:

$$P(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}}$$

Quite often, the continuous/numeric independent/input variables are discretized (using an appropriate binning method), and then a categorical variable estimation method is used to calculate

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

the conditional probabilities (likelihood parameters). If performed properly, this method tends to produce better-predicting Naive Bayes models.

Testing Phase

Using the two sets of parameters produced in steps 3 and 4 of the training phase, any new sample can be classified into a class label by using the following formula:

$$Posterior = \frac{Prior \times Likelihood}{Evidence}$$

$$P(C|F_1,\ldots,F_n) = \frac{P(C)\ P(F_1,\ldots,F_n \mid C)}{P(F_1,\ldots,F_n)}$$

Because the denominator is constant (i.e., the same for all class labels), we can remove it from the formula, which leaves us with the following simpler formula, which is essentially the joint probability:

$$classify(f_1,\ldots,f_n) = \underset{c}{\text{argmax}}\ p(C=c) \prod_{i=1}^{n} p(F_i = f_i \mid C=c)$$

Naive Bayes is not very commonly used in predictive analytics projects today due to its relatively poor prediction performance in a wide variety of application domains. However, one of its extensions, called Bayesian network, is gaining surprisingly rapid popularity among data scientists in the analytics world.

Nearest Neighbor

Predictive analytics algorithms tend to be highly mathematical and computationally intensive. Two popular ones—artificial neural networks and support vector machines—are involved in overly time-demanding, computationally intensive iterative mathematical derivations. In contrast, the *k*-nearest neighbor (*k*-NN) algorithm seems overly simplistic for a competitive prediction method. It is easy to

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

understand (and explain to others) what it does and how it does it. *k*-NN is a prediction method for classification- as well as regression-type prediction problems. *k*-NN is a type of instance-based learning (or lazy learning) in which the function is only approximated locally, and all computations are deferred until the actual prediction.

The *k*-NN algorithm is among the simplest of all machine learning algorithms. For instance, in classification-type prediction, a case is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (where *k* is a positive integer). If $k = 1$, then the case is simply assigned to the class of its nearest neighbor. To illustrate the concept with an example, Figure 5.1 shows a simple two-dimensional space representing the values for the two variables *x* and *y*; the star represents a new case (or object), and circles and squares represent known cases (or examples). The task is to assign the new case either to circles or squares, based on its closeness (similarity) to one or the other. If you set the value of *k* to 1 ($k = 1$), the assignment should be made to squares because the closest example to a star is a square. If you set the value of *k* to 3 ($k = 3$), then the assignment should be made to a circle because there are two circles and one square, and hence by simple majority vote rule, the circle gets the assignment of the new case. Similarly, if you set the value of *k* to 5 ($k = 5$), then the assignment should be made to the square class. This overly simplified example is meant to illustrate the importance of the value assigned to *k*.

To use the same method for regression-type prediction tasks, simply average the values of the *k* nearest neighbors and assign that result to the case being predicted. It can be useful to weight the contributions of the neighbors so that the nearer neighbors contribute more to the average than do the more distant ones. A common weighting scheme is to give each neighbor the weight $1/d$, where *d* is the distance to the neighbor. This scheme is essentially a generalization of linear interpolation.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

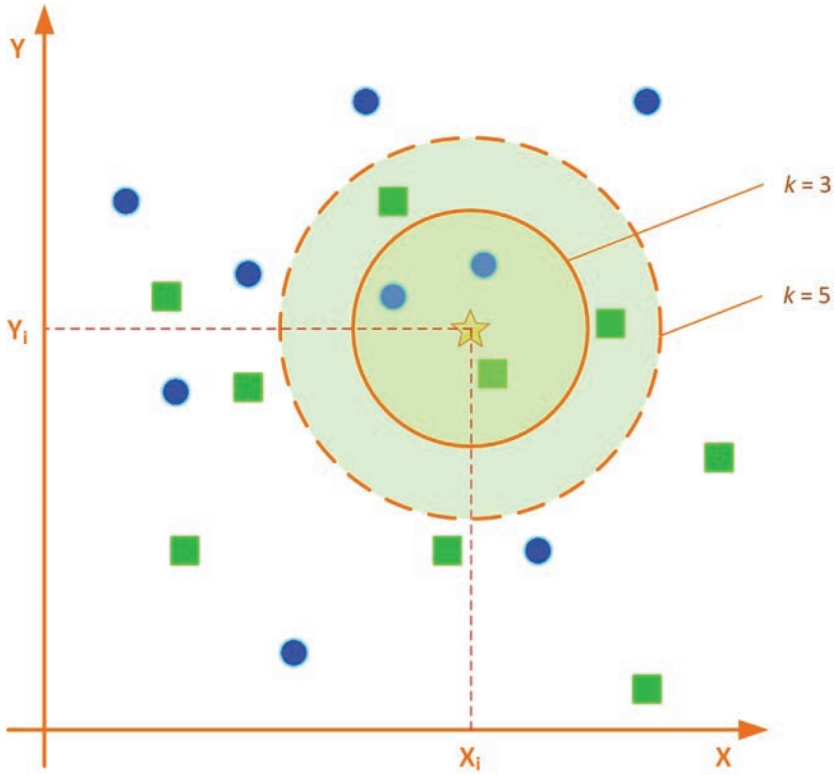


Figure 5.1 The Importance of the Value of *k* in the *k*-NN Algorithm

The neighbors are taken from a set of cases for which the correct classification (or, in the case of regression, the numeric value of the output value) is known. This can be thought of as the training set for the algorithm, even though no explicit training step is required. The *k*-nearest neighbor algorithm is sensitive to the local structure of the data.

Similarity Measure: The Distance Metric

One of the two critical decisions that an analyst has to make while using *k*-NN is to determine the similarity measure; the other is to determine the value of *k*, which is explained below. In the

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

k-NN algorithm, the similarity measure is a mathematically calculable distance metric. Given a new case, *k*-NN makes predictions based on the outcome of the *k* neighbors closest in distance to that point. Therefore, to make predictions with *k*-NN, an analyst needs to define a metric for measuring the distance between the new case and the cases from the examples. One of the most popular choices for measuring this distance is known as Euclidean (Equation 5.3), which is simply the linear distance between two points in a dimensional space; the other popular one is the rectilinear (a.k.a. city-block or Manhattan distance) (Equation 5.2). Both of these distance measures are special cases of Minkowski distance (Equation 5.1).

Here is the equation for finding the Minkowski distance:

$$d(i,j) = \sqrt[q]{\left(\left|x_{i_1} - x_{j1}\right|^q + \left|x_{i_2} - x_{j2}\right|^q + \quad + \left|x_{ip} - x_{jp}\right|^q\right)} \qquad (5.1)$$

where $i = (x_{i1}, x_{i2},\ldots,x_{ip})$ and $j = (x_{j1}, x_{j2},\ldots,x_{jp})$ are two p-dimensional data objects (e.g., a new case and an example in the data set), and q is a positive integer.

If $q = 1$, then d is called *Manhattan distance* and is found using this equation:

$$d(i,j) = \left|x_{i_1} - x_{j1}\right| + \left|x_{i_2} - x_{j2}\right| + \quad + \left|x_{ip} - x_{jp}\right| \qquad (5.2)$$

If $q = 2$, then d is called *Euclidean distance* and is found using this equation:

$$d(i,j) = \sqrt{\left(\left|x_{i_1} - x_{j1}\right|^2 + \left|x_{i_2} - x_{j2}\right|^2 + \quad + \left|x_{ip} - x_{jp}\right|^2\right)} \qquad (5.3)$$

Obviously, these measures apply only to numerically represented data. What happens with nominal data? There are ways to measure distance for nonnumeric data as well. In the simplest case, for a multivalue nominal variable, if the value of that variable for the new case and the example case is the same, the distance would be zero; otherwise, it would be one. In cases such as text classification, more sophisticated metrics exist, such as the overlap metric (or Hamming

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

distance). Often, the classification accuracy of k-NN can be improved significantly if the distance metric is determined through an experimental design in which different metrics are tried and tested to identify the best one for the given problem.

Parameter Selection

The best choice for k depends on the data; generally, larger values of k reduce the effect of noise on the classification (or regression) but also make boundaries between classes less distinct. An "optimal" value of k can be found by using some heuristic techniques, such as cross-validation. The special case where the class is predicted to be the class of the closest training sample (i.e., when $k = 1$) is called the *nearest-neighbor algorithm*.

Cross-Validation

Cross-validation is a well-established experimentation technique that can be used to determine optimal values for a set of unknown model parameters. It applies to most, if not all, of the machine learning techniques, where there are a number of model parameters to be determined. The general idea of this experimentation method is to divide the data sample into a number of randomly drawn, disjointed subsamples (i.e., v number of folds). For each potential value of k, the k-NN model is used to make predictions on the vth fold while using the $v - 1$ folds as the examples, and evaluate the error. The common choice for this error is the root mean squared error (RMSE) for regression-type predictions and percentage of correctly classified instances (i.e., hit rate) for classification-type predictions. This process of testing each fold against the remaining examples repeats v times. At the end of the v number of cycles, the computed errors are accumulated to yield a goodness measure of the model (i.e., how well the model predicts using the current value of k). At the end, the k value that produces the smallest overall error is chosen as the

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

optimal value for that problem. Figure 5.2 shows a simple process in which training data is used to determine optimal values for *k* and the distance metric, which are then used to predict new incoming cases.

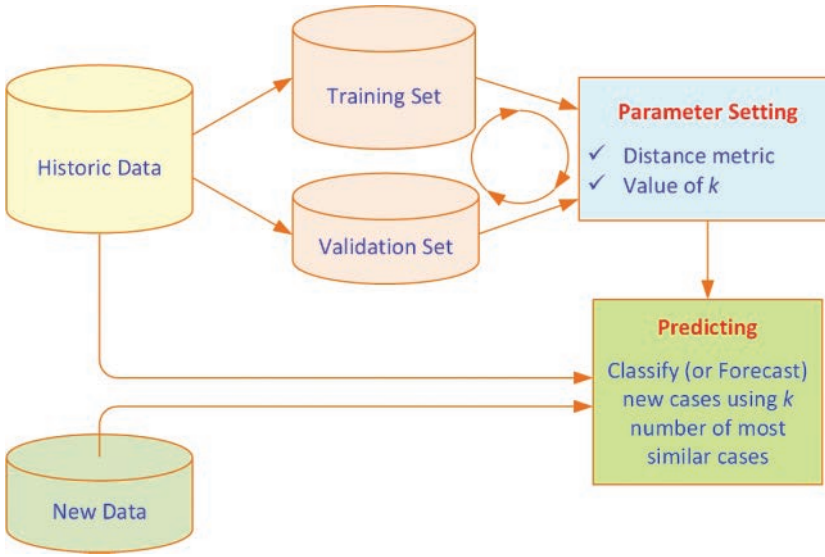


Figure 5.2 The Process of Determining the Optimal Values for the Distance Metric and *k*

As in the simple example just provided, the accuracy of the *k*-NN algorithm can be significantly different with different values of *k*. Furthermore, the predictive power of the *k*-NN algorithm degrades in the presence of noisy, inaccurate, or irrelevant features. Much research effort has been put into feature selection and normalization/scaling to ensure reliable prediction results. A particularly popular approach is the use of evolutionary algorithms (e.g., genetic algorithms) to optimize the set of features included in the *k*-NN prediction system. In binary (two-class) classification problems, it is helpful to choose an odd number for *k* to avoid tied votes.

A drawback to the basic majority voting classification in *k*-NN is that the classes with the more frequent examples tend to dominate the prediction of the new vector, as they tend to come up in the *k* nearest neighbors when the neighbors are computed due to their

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

large number. One way to overcome this problem is to weigh the classification, taking into account the distance from the test point to each of its *k* nearest neighbors. Another way to overcome this drawback is to use one level of abstraction in data representation.

The naive version of the *k*-NN algorithm is easy to implement by computing the distances from the test sample to all stored vectors, but it is computationally intensive, especially when the size of the training set grows. Many nearest-neighbor search algorithms have been proposed over the years; they generally seek to reduce the number of distance evaluations actually performed. Using an appropriate nearest-neighbor search algorithm makes *k*-NN computationally tractable even for large data sets.

Artificial Neural Networks

Neural networks represent a brain metaphor for information processing. These models are biologically inspired rather than exact replicas of how the brain actually functions. Neural networks have been shown to be very promising systems in many forecasting and business classification applications due to their ability to "learn" from the data, their nonparametric nature (i.e., their lack of rigid assumptions), and their ability to generalize. Neural computing refers to a pattern-recognition methodology for machine learning. The model that results from neural computing is often called an *artificial neural network* (*ANN*) or a *neural network*. Neural networks have been used in many business applications for pattern recognition, forecasting, prediction, and classification. Neural network computing is a key component of any data science and business analytics toolkit. Applications of neural networks abound in finance, marketing, manufacturing, operations, information systems, and so on.

In this section, we provide a brief introduction to the vast variety of neural network models, methods, and applications. We cover neural networks—especially the feed-forward multilayer perceptron-type

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

prediction modeling specific neural network architecture—in more detail in Chapter 6, "Advanced Topics in Predictive Modeling," as a primer to the understanding of deep learning and deep neural networks.

The human brain possesses bewildering capabilities for information processing and problem solving that modern computers cannot compete with in many aspects. It has been postulated that a model or a system that is enlightened and supported by the results from brain research, with a structure similar to that of biological neural networks, could exhibit similar intelligent functionality. Based on this bottom-up approach, ANNs (also known as connectionist models, parallel distributed processing models, neuromorphic systems, or simply neural networks) have been developed as biologically inspired and plausible models for various tasks.

Biological neural networks are composed of many massively interconnected neurons. Each neuron possesses axons and dendrites, fingerlike projections that enable the neuron to communicate with its neighboring neurons by transmitting and receiving electrical and chemical signals. More or less resembling the structure of their biological counterparts, ANNs are composed of interconnected, simple processing elements called artificial neurons. When processing information, the processing elements in an ANN operate concurrently and collectively, similar to biological neurons. ANNs possess some desirable traits similar to those of biological neural networks, such as the abilities to learn, to self-organize, and to support fault tolerance.

Researchers have been examining the winding journey of ANNs for more than half a century. The formal study of ANNs began with the pioneering work of McCulloch and Pitts in 1943. Inspired by the results of biological experiments and observations, McCulloch and Pitts (1943) introduced a simple model of a binary artificial neuron that captured some of the functions of biological neurons. Using information-processing machines to model the brain, McCulloch and Pitts built their neural network model using a large number of

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

interconnected artificial binary neurons. From these beginnings, neural network research became quite popular in the late 1950s and early 1960s. After a thorough analysis of an early neural network model (called the *perceptron*, which used no hidden layer) and a pessimistic evaluation of the research potential by Minsky and Papert in 1969, interest in neural networks diminished.

During the past two decades, there has been an exciting resurgence in ANN studies due to the introduction of new network topologies, new activation functions, and new learning algorithms, as well as progress in neuroscience and cognitive science. Advances in theory and methodology have overcome many of the obstacles that hindered neural network research a few decades ago. Thanks to the appealing results of numerous studies, neural networks are gaining acceptance and popularity. In addition, the desirable features in neural information processing make neural networks attractive for solving complex problems. ANNs have been applied to numerous complex problems in a variety of application settings. The successful use of neural network applications has inspired renewed interest from industry and business. With the emergence of deep neural networks (as part of the rather recent deep learning phenomenon), the popularity of neural networks (with a deeper architectural representation and much enhanced analytics capabilities) has hit an unprecedented height, and the expectations for this new generation of neural networks are great. Deep neural networks are covered in detail in Chapter 6.

Biological Versus Artificial Neural Networks

The human brain is composed of special cells called neurons. These cells do not die and replenish when a person is injured. (All other cells reproduce to replace themselves and then die.) This phenomenon may explain why humans retain information for an extended period of time and start to lose it when they get old—as the brain cells gradually start to die. Information storage spans sets of neurons. The

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

brain has anywhere from 50 billion to 150 billion neurons, of which there are more than 100 different kinds. Neurons are partitioned into groups called networks. Each network contains several thousand highly interconnected neurons. Thus, the brain can be viewed as a collection of neural networks.

The ability to learn and to react to changes in our environment requires intelligence. The brain and the central nervous system control thinking and intelligent behavior. People who suffer brain damage have difficulty learning and reacting to changing environments. Even so, undamaged parts of the brain can often compensate with new learning.

Figure 5.3 shows a portion of a network composed of two cells. Each cell includes a nucleus (the central processing portion of the neuron). To the left of cell 1, the dendrites provide input signals to the cell. To the right, the axon sends output signals to cell 2 via the axon terminals. These axon terminals merge with the dendrites of cell 2. Signals can be transmitted unchanged, or they can be altered by synapses. A synapse is able to increase or decrease the strength of the connection between neurons and cause excitation or inhibition of a subsequent neuron. This is how information is stored in the neural networks.

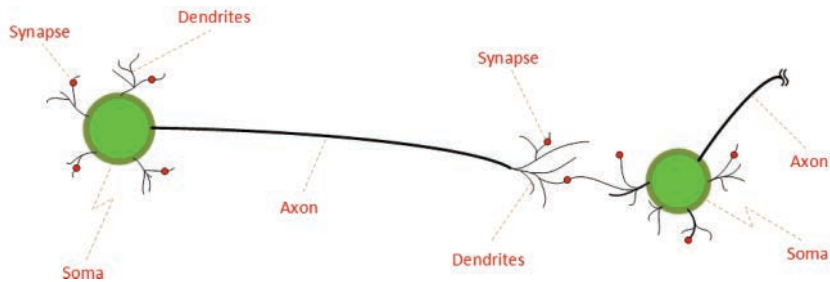


Figure 5.3 A Biological Neural Network: Two Interconnected Cells/Neurons

An ANN emulates a biological neural network. Neural computing actually uses a very limited set of concepts from biological neural systems. Table 5.1 provides a simple terminology mapping between

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

biological and artificial neural networks. It is more an analogy to the human brain than an accurate model of it. Neural concepts usually are implemented as software simulations of the massively parallel processes involved in processing interconnected elements (also called artificial neurons, or neurodes) in a network architecture. The artificial neuron receives inputs analogous to the electrochemical impulses that dendrites of biological neurons receive from other neurons. The output of the artificial neuron corresponds to signals sent from a biological neuron over its axon. These artificial signals can be changed by weights in a manner similar to the physical changes that occur in the synapses (see Figure 5.4).

Table 5.1 Mapping Between Biological and Artificial NNs

Biological NN	Artificial NN
Soma	Node
Dendrites	Input
Axon	Output
Synapse	Weight
Slow	Fast
Many neurons (109)	Few neurons (a dozen to hundreds of thousands)

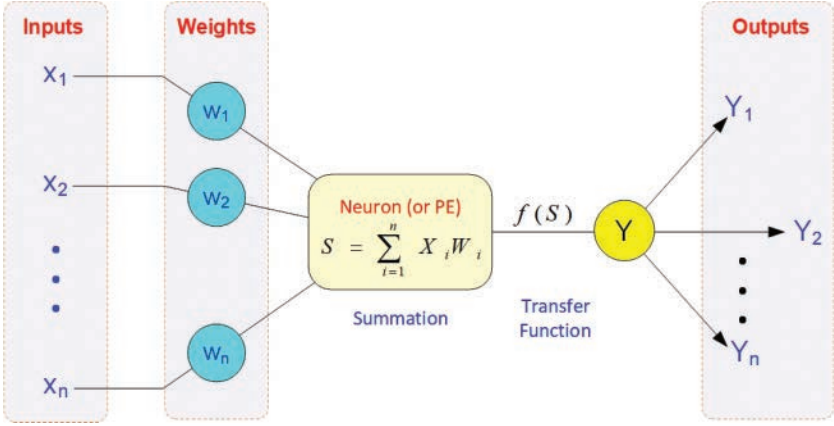


Figure 5.4 Processing Information in an Artificial Neuron

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Several ANN paradigms have been proposed for applications in a variety of problem domains. Perhaps the easiest way to differentiate among the various neural models is on the basis of how they structurally emulate the human brain, the way they process information, and how they learn to perform their designated tasks.

Because they are biologically inspired, the main processing elements of a neural network are individual neurons, analogous to the brain's neurons. These artificial neurons receive information from other neurons or external input stimuli, perform a transformation on the inputs, and then pass on the transformed information to other neurons or external outputs. This is similar to how it is presently thought that the human brain works. Passing information from neuron to neuron can be thought of as a way to activate, or trigger, a response from certain neurons based on the information or stimulus received.

How information is processed by a neural network is inherently a function of its structure. Neural networks can have one or more layers of neurons. These neurons can be highly or fully interconnected, or only certain layers can be connected. Connections between neurons have associated weights. In essence, the "knowledge" possessed by the network is encapsulated in these interconnection weights. Each neuron calculates a weighted sum of the incoming neuron values, transforms this input, and passes on its neural value as the input to subsequent neurons. Typically, although not always, this input/output transformation process at the individual neuron level is performed in a nonlinear fashion.

Support Vector Machines

Support vector machines (SVMs) are among the most popular machine learning techniques of the recent past, largely because of their superior predictive performance and their theoretical

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

foundation. SVMs are among the supervised learning methods that produce input/output functions from a set of labeled training data. The function between the input and output vectors can be either a classification function (used to assign cases into predefined classes) or a regression function (used to estimate the continuous numeric value of the desired output). For classification, nonlinear kernel functions are often used to transform the input data (naturally representing highly complex nonlinear relationships) to a high-dimensional feature space in which the input data becomes linearly separable. Then, the maximum-margin hyperplanes are constructed to optimally separate the output classes from each other in the training data.

For a given classification-type prediction problem, generally speaking, many linear classifiers (hyperplanes) can separate the data into multiple subsections, each representing one of the classes. For example, in Figure 5.5, the two classes are represented with circles and squares. However, only one hyperplane achieves the maximum separation between the classes; in Figure 5.5b, the hyperplane and the two maximum-margin hyperplanes are separating the two classes.

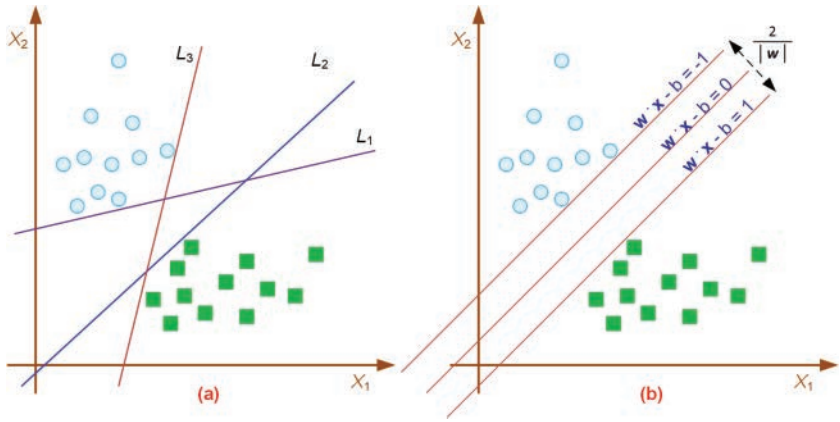


Figure 5.5 Separation of the Two Classes with Support Vector Machines

Data used in SVMs may have more than two dimensions (i.e., two distinct classes). In that case, we would be interested in separating

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

data using $n - 1$ dimensional hyperplanes, where n is the number of dimensions (i.e., class labels). This may be seen as a typical form of linear classifier, where we are interested in finding $n - 1$ hyperplanes so that the distances from the hyperplanes to the nearest data points are maximized. The assumption is that the larger the margin or distance between these parallel hyperplanes, the better the generalization power of the classifier (i.e., prediction power of the SVM model). If such hyperplanes exist, they can be mathematically represented using quadratic optimization modeling. These hyperplanes are known as the maximum-margin hyperplanes, and a linear classifier is known as a maximum-margin classifier.

In addition to their solid mathematical foundation in statistical learning theory, SVMs have also demonstrated highly competitive performance in numerous real-world prediction problems, such as medical diagnosis, bioinformatics, face/voice recognition, demand forecasting, image processing, and text mining, which has established SVMs as one of the most popular analytics tools for knowledge discovery and data mining. Similarly to artificial neural networks, SVMs can be used to approximate any multivariate function to any desired degree of accuracy. Therefore, they are of particular interest in modeling highly nonlinear, complex problems, systems, and processes.

The Process of Building SVM Models

Recently, support vector machines have often been used for classification-type problems because of their predictive accuracy and self-expandability. Even though people consider them easier to use than artificial neural networks, users who are not familiar with the intricacies of SVMs often get unsatisfactory results. In this section, we provide a simple process-based model building with the use of SVMs that is quite likely to produce surprisingly good results. A pictorial representation of the three-step process is given in Figure 5.6.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

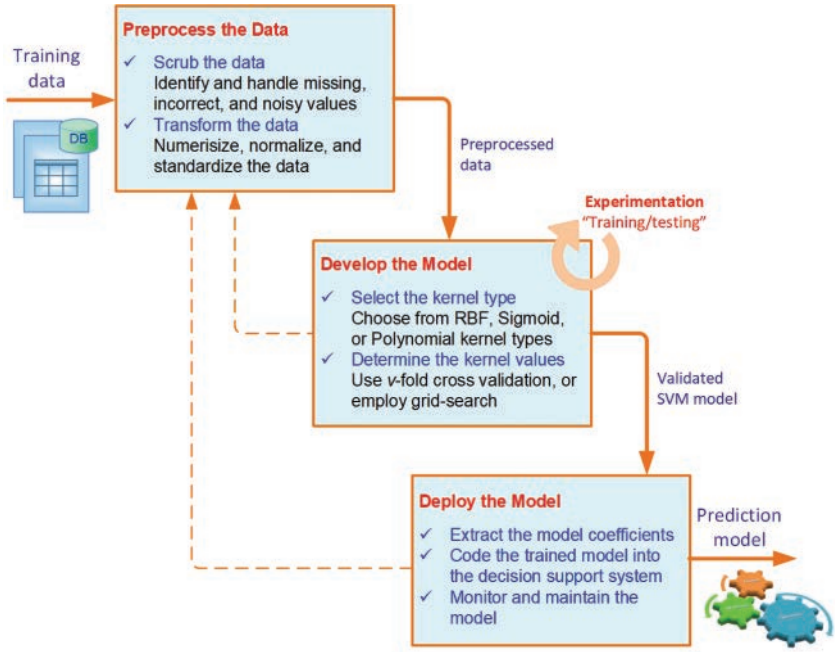


Figure 5.6 A Simple Process for Developing and Using SVM Models

The following sections provide short descriptions of the steps in this process.

Step 1: Preprocess the Data

Because real-world data is anything but perfect, a data mining analyst needs to do the due diligence to scrub and transform the data for SVM. As is the case with any other data mining methods and algorithms, this step is likely to be the most time-demanding and least enjoyable—yet most essential—part of working with SVMs. Some of the tasks in this step include handling missing, incomplete, and noisy values and numerisizing, normalizing, and standardizing the data:

- **Numerisizing the data.** For an SVM, each data instance needs to be represented as a vector of real numbers. Hence, if there are categorical attributes, an analyst has to first convert them into numeric data. A common recommendation is to

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

use m pseudo binary variables to represent an m-class attribute (where $m \geq 3$). In practice, only one of the m variables assumes the value 1, and others assume the value 0, based on the actual class of the case; this is also called 1-of-m representation. For example, a three-category attribute such as {red, green, blue} can be represented as [0,0,1], [0,1,0], and [1,0,0].

- **Normalizing the data.** As is the case with artificial neural networks, SVMs also require normalization and/or scaling of numeric values. The main advantage of normalization is to avoid having attributes in greater numeric ranges dominate those in smaller numeric ranges. Another advantage is that it helps perform numeric calculations during the iterative process of model building. Because kernel values usually depend on the inner products of feature vectors (e.g., the linear kernel, the polynomial kernel), large attribute values might slow the training process. The usual recommendation is to normalize each attribute to the range [–1, +1] or [0, 1]. Of course, an analyst has to use the same normalization method to scale testing data before testing.

Step 2: Develop the Model

After the data is preprocessed, the model building step starts. Compared to the other two steps, this is the most enjoyable part of the process, where the prediction model comes alive. Because SVM has a number of parameter options, the process requires a lengthy process of determining the optimal combination of those parameters for the best possible performance. The most important of those parameters are the kernel type and consequently kernel-related subparameters.

There are four common kernels, and an analyst must decide which one to use (or whether to try them all, one at a time, using a simple experimental design approach). Once the kernel type is selected, the analyst needs to select the value of penalty parameter C

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

and the kernel parameters. Generally speaking, RBF is a reasonable first choice for the kernel type. The RBF kernel aims to nonlinearly map data into a higher-dimensional space, and this (unlike with a linear kernel) handles the cases in which the relationship between input and output vectors is highly nonlinear. Besides, the linear kernel is just a special case of the RBF kernel. There are two parameters to choose for RBF kernels: C and γ. It is not known beforehand which C and γ are the best for a given prediction problem; therefore, some kind of parameter search method needs to be used. The goal for the search is to identify optimal values for C and γ so that the classifier can accurately predict unknown data (i.e., testing data). The two most commonly used search methods are cross-validation and grid search.

Step 3: Deploy the Model

Once an optimal SVM prediction model is developed, the next step is to integrate it into the decision support system. For that, there are two options: (1) converting the model into a computational object (e.g., a web service, a JavaBean object, or a COM object) that takes the input parameter values and provides output prediction or (2) extracting the model coefficients and integrating them directly into the decision support system. SVM models are useful (i.e., accurate and actionable) only if the behavior of the underlying domain stays the same. If the behavior changes, so does the accuracy of the model. Therefore, it's important to continuously assess the performance of the models and decide when they are no longer accurate and, hence, need to be retrained.

Support Vector Machines Versus Artificial Neural Networks

Even though some people characterize SVMs as a special case of ANNs, most recognize these as two competing machine learning techniques with different qualities. Let's look at a few points that help SVMs stand out against ANNs. Historically, the development

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

of ANNs followed a heuristic path, with applications and extensive experimentation preceding theory. In contrast, the development of SVMs involved sound statistical learning theory first, followed by implementation and experiments. A significant advantage of SVMs is that whereas ANNs may suffer from multiple local minima, the solutions to SVMs are global and unique. Two more advantages of SVMs are that they have simple geometric interpretations and give sparse solutions. The reason that SVMs often outperform ANNs in practice is that they successfully deal with the "overfitting" problem, which is a big issue with ANNs.

Besides the above-mentioned advantages of SVMs (from a practical point of view), they also have some limitations. An important issue that is not entirely solved is the selection of the kernel type and kernel function parameters. Other, perhaps more important, limitations of SVMs are the speed and size, in terms of both training and testing cycles. Model building with SVMs involves complex and time-demanding calculations. From a practical point of view, perhaps the most serious problems with SVMs are the high algorithmic complexity and extensive memory requirements of the required quadratic programming in large-scale tasks. Despite these limitations, because SVMs are based on a sound theoretical foundation and the solutions they produce are global and unique in nature (as opposed to getting stuck in suboptimal alternatives, as local minima do), today's SVMs are very popular for prediction modeling in the data mining arena. Their use and popularity will only increase as the popular commercial data mining tools start to incorporate them into their modeling arsenals.

Linear Regression

Regression, especially linear regression, is perhaps the most widely used analysis technique in statistics. Historically speaking, the roots of regression date back to the 1920s and 1930s, to the early

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

work on inherited characteristics of sweet peas by Sir Francis Galton and subsequently Karl Pearson. Since then, regression has become the statistical technique for characterization of relationships between explanatory (input) variables and response (output) variables.

As popular as it is, regression is essentially a relatively simple statistical technique for modeling the dependence of a variable (response or output variable) on one or more explanatory (input) variables. Once identified, the relationship between the variables can be formally represented as a linear/additive function or equation. Like many other modeling techniques, regression aims to capture the functional relationship between and among the characteristics of the real world and describe this relationship with a mathematical model, which may then be used to discover and understand the complexities of reality—that is, explore and explain relationships or forecast future occurrences.

Regression can be used for two different purposes: hypothesis testing—investigating potential relationships between different variables—and prediction/forecasting—estimating values of response variables based on one or more explanatory variables. These two uses are not mutually exclusive. The explanatory power of regression is also the foundation of its prediction ability. In hypothesis testing (theory building), regression analysis can reveal the existence and strength and directions of relationships between a number of explanatory variables (often represented as x_i) and the response variable (often represented as y). In prediction, regression uses an equation to identify additive mathematical relationships between one or more explanatory variables and a response variable. Once determined, this equation can be used to forecast the values of the response variable for a given set of values of the explanatory variables.

Correlation Versus Regression

Because regression analysis originated in correlation studies, and because both methods attempt to describe the association between

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

two (or more) variables, the terms *regression* and *correlation* are often confused, even by scientists. Correlation makes no a priori assumption as to whether one variable is dependent on the other(s) and is not concerned with the relationship between variables; instead, it gives an estimate of the degree of association between the variables. On the other hand, regression attempts to describe the dependence of a response variable on one or more explanatory variables, where it implicitly assumes that there is a one-way causal effect from the explanatory variables to the response variable, regardless of whether the path of effect is direct or indirect. Also, whereas correlation is interested in low-level relationships between two variables, regression is concerned with the relationships between all explanatory variables and the response variable.

Simple Versus Multiple Regression

If a regression equation is built between one response variable and one explanatory variable, then it is called *simple regression*. For instance, the regression equation built to predict or explain the relationship between the height of a person (explanatory variable) and the weight of a person (response variable) is a good example of simple regression. Multiple regression is an extension of simple regression in which there are multiple explanatory variables. For instance, in the previous example, if we were to include not only the height of a person but also other personal characteristics (e.g., BMI, gender, ethnicity) to predict the weight of the person, then we would be performing multiple regression analysis. In both cases, the relationships between the response variables and the explanatory variables are linear and additive in nature. If the relationships are not linear, then we might want to use one of many other nonlinear regression methods to better capture the relationships between the input and output variables.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

How to Develop a Linear Regression Model

To understand the relationship between two variables, the simplest thing to do is to draw a scatterplot, with the *y*-axis representing the values of the response variable and the *x*-axis representing the values of the explanatory variable (see Figure 5.7). Such a scatterplot would show the changes in the response variable as a function of the changes in the explanatory variable. In the example shown in Figure 5.7, there seems to be a positive relationship between the two: As the explanatory variable values increase, so does the response variable.

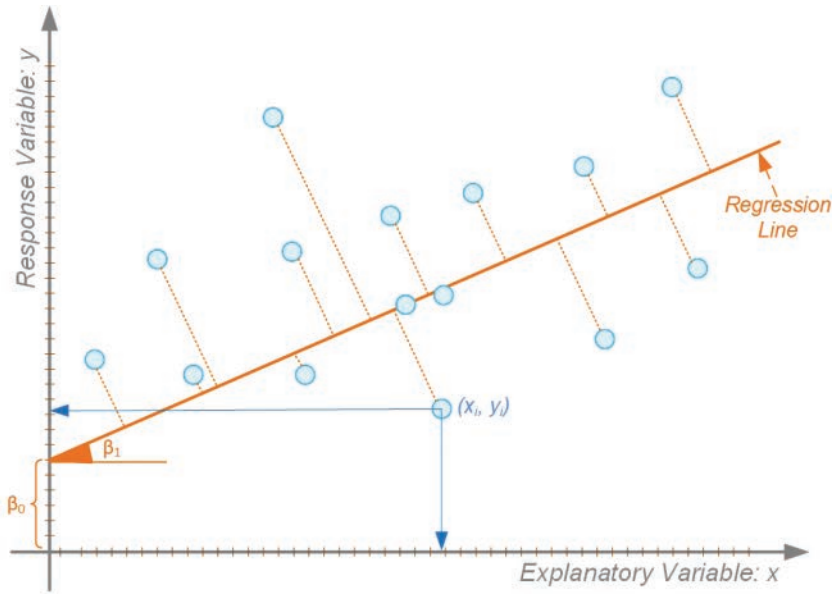


Figure 5.7 A Scatter Plot with a Simple Linear Regression Line

Simple regression analysis aims to find a mathematical representation of this relationship. In reality, it tries to find the signature of a straight line passing right in between the plotted dots (representing the observation/historical data) in such a way that it minimizes the distance between the dots and the line (the predicted values on the

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

theoretical regression line). Several methods/algorithms have been proposed to identify the regression line, and the one that is most commonly used is called the ordinary least squares (OLS) method. The OLS method aims to minimize the sum of squared residuals (i.e., squared vertical distances between the observation and the regression point) and leads to a mathematical expression for the estimated value of the regression line (known as the β parameter). For simple linear regression, the aforementioned relationship between the response variable (y) and the explanatory variable(s) (x) can be shown as a simple equation, as follows:

$$y = \beta_0 + \beta_1 x$$

In this equation, β_0 is called the *intercept*, and β_1 is called the *slope*. Once OLS determines the values of these two coefficients, the simple equation can be used to forecast the values of y for given values of x. The sign and the value of β_1 also reveal the direction and the strength of the relationship between the two variables.

If the model is of a multiple linear regression type, then more coefficients need to be determined—one for each additional explanatory variable. As the following formula shows, the additional explanatory variable would be multiplied with new β_i coefficients and summed together to establish a linear additive representation of the response variable:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

How to Tell Whether a Model Is Good Enough

For a variety of reasons, sometimes models do not do a good job representing reality. Regardless of the number of explanatory variables included, there is always a possibility of not having a good model, and therefore a linear regression model needs to be assessed

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

for its fit (i.e., the degree to which it represents the response variable). In the simplest sense, a well-fitting regression model results in predicted values close to the observed data values. For the numeric assessment, three statistical measures are often used in evaluating the fit of a regression model: R^2 (R-squared), the overall F-test, and the root mean squared error (RMSE). All three of these measures are based on sums of square errors (i.e., how far the data is from the mean and how far the data is from the model's predicted values). Different combinations of these two values provide different information about how the regression model compares to the mean model.

Of the three, R^2 has the most useful and understandable meaning because of its intuitive scale. The value of R^2 ranges from 0 to 1 (corresponding to the amount of variability, expressed as a percentage), with 0 indicating that the relationship and the prediction power of the proposed model is not good and 1 indicating that the proposed model is a perfect fit that produces exact predictions (which is almost never the case). Good R^2 values usually come close to 1, and the closeness is a matter of the phenomenon being modeled; for example, while an R^2 value of 0.3 for a linear regression model in social science may be considered good enough, an R^2 value of 0.7 in engineering might not be considered a good fit. Improvement in the regression model can be achieved by adding more explanatory variables, taking some of the variables out of the model, or using different data transformation techniques, which would result in comparative increases in an R^2 value.

Figure 5.8 shows the process flow involved in developing regression models. As shown in the figure, the model development task is followed by model assessment. Model assessment involves assessing the fit of the model and also, because of restrictive assumptions with which linear models have to comply, it involves examining the validity of the model.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

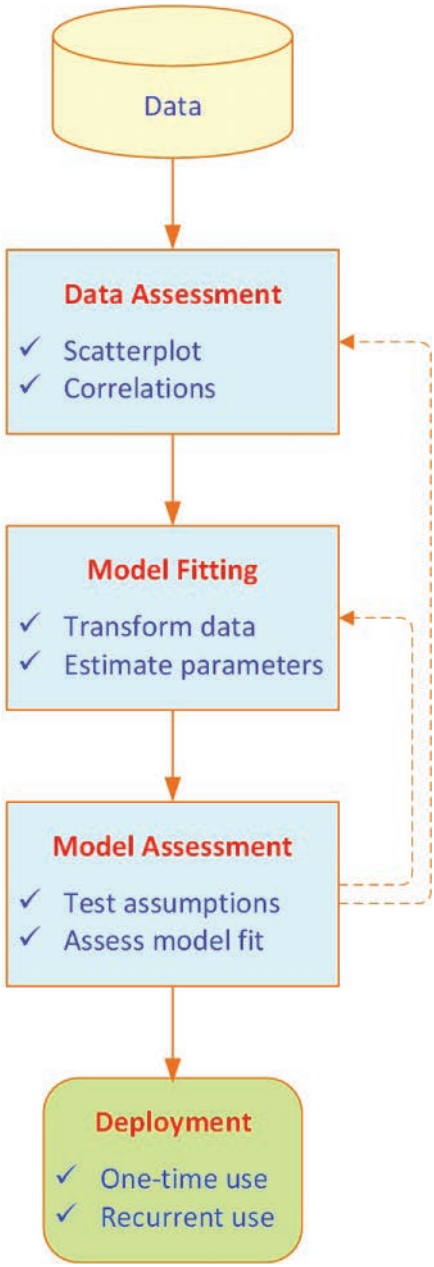


Figure 5.8 Process Flow for Developing Regression Models

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The Most Important Assumptions in Linear Regression

Even though they are still the top choice of many data analysts (both for explanatory as well as predictive modeling purposes), linear regression models suffer from several highly restrictive assumptions. The validity of a linear model depends on its ability to comply with these assumptions. These are the most common assumptions:

- **Linearity.** This assumption states that the relationship between the response variable and the explanatory variables are linear. That is, the expected value of the response variable is a straight-line function of each explanatory variable, with all other explanatory variables held fixed. Also, the slope of the line does not depend on the values of the other variables. This implies that the effects of different explanatory variables on the expected value of the response variable are additive in nature.
- **Independence (of errors).** This assumption states that the errors of the response variable are not correlated with each other. This independence of the errors is weaker than actual statistical independence, which is a stronger condition and is often not needed for linear regression analysis.
- **Normality (of errors).** This assumption states that the errors of the response variable are normally distributed. That is, they are supposed to be totally random and should not represent any nonrandom patterns.
- **Constant variance (of errors).** This assumption, also called *homoscedasticity*, states that the response variables must have the same variance in their error, regardless of the values of the explanatory variables. In practice, this assumption is invalid if the response variable varies over a wide enough range or scale.
- **Multicollinearity.** This assumption states that the explanatory variables are not correlated (i.e., do not replicate the same but provide a different perspective of the information needed for

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

the model). Multicollinearity can be triggered by having two or more perfectly correlated explanatory variables present in the model (e.g., if the same explanatory variable is mistakenly included in the model twice, one with a slight transformation of the other variable). A correlation-based data assessment usually catches this error.

Statistical techniques have been developed to identify the violation of these assumptions, and several techniques have been created to mitigate them. A data modeler needs to be aware of the existence of these assumptions and put in place a way to assess models to make sure they are compliant with the assumptions they are built on.

Logistic Regression

Logistic regression is a very popular, statistically sound, probability-based classification algorithm that employs supervised learning. It was developed in the 1940s as a complement to linear regression and linear discriminant analysis methods. It has been used extensively in numerous disciplines, including the medical and social science fields. Logistic regression is similar to linear regression in that it also aims to regress to a mathematical function that explains the relationship between the response variable and the explanatory variables, using a sample of past observations (training data). It differs from linear regression on one major point: Its output (response variable) is a class as opposed to a numeric variable. That is, whereas linear regression is used to estimate a continuous numeric variable, logistic regression is used to classify a categorical variable. Even though the original form of logistic regression was developed for a binary output variable (e.g., 1/0, yes/no, pass/fail, accept/reject), the present-day modified version is capable of predicting multiple-class output variables (i.e., multinomial logistic regression). If there is only one predictor variable and one predicted variable, the method is called *simple logistic regression*;

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

similarly, *simple linear regression* is the term for a linear regression model with only one independent variable.

In predictive analytics, logistic regression models are used to develop probabilistic models between one or more explanatory or predictor variables (which may be a mix of both continuous and categorical variables) and a class or response variable (which may be binomial/binary or multinomial/multiple-class variables). Unlike ordinary linear regression, logistic regression is used for predicting categorical (often binary) outcomes of the response variable—treating the response variable as the outcome of a Bernoulli trial. Therefore, logistic regression takes the natural logarithm of the odds of the response variable to create a continuous criterion as a transformed version of the response variable. Thus the logit transformation is referred to as the *link function* in logistic regression; even though the response variable in logistic regression is categorical or binomial, the logit is the continuous criterion on which linear regression is conducted.

Figure 5.9 shows a logistic regression function where the odds are represented on the x-axis (i.e., a linear function of the independent variables), and the probabilistic outcome is shown on the y-axis (i.e., response variable values change between 0 and 1).

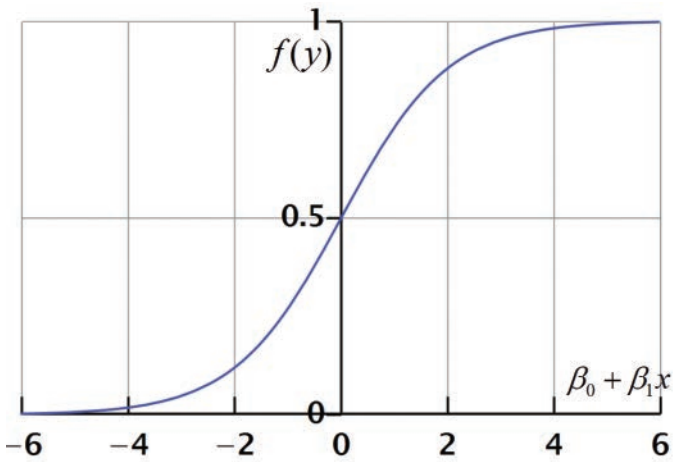


Figure 5.9 The Logistic Function

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The logistic function, which is $f(y)$ in Figure 5.9, is the core of logistic regression and can only take values between 0 and 1. The following equation is a simple mathematical representation of this function:

$$f(y) = \frac{1}{1 + e^{-(\beta_0 + \beta_0 x)}}$$

The logistic regression coefficients (the βs) are usually estimated using the maximum-likelihood estimation method. Unlike with linear regression with normally distributed residuals, it is not possible to find a closed-form expression for the coefficient values that maximizes the likelihood function, and an iterative process must be used instead. This process begins with a tentative starting solution and then revises the parameters slightly to see if the solution can be improved; it repeats this iterative revision until no improvement can be achieved or the improvements are very minimal, at which point the process is said to have completed, or converged.

Time-Series Forecasting

Sometimes the variable of interest (i.e., the response variable) may not have distinctly identifiable explanatory variables, or there may be too many of them in a highly complex relationship. In such cases, if the data is available in the desired format, a prediction model called a *time-series forecast* can be developed. A time series is a sequence of data points of the variable of interest, measured and represented at successive points in time and spaced at uniform time intervals. Examples of time series include monthly rain volumes in a geographic area, the daily closing value of the stock market indices, and daily sales totals for a grocery store. Often, time series are visualized using a line chart. Figure 5.10 shows an example of a time series of sales volumes for the years 2015 through 2019, on a quarterly basis.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.



Figure 5.10 Sample Time-Series Data on Quarterly Sales Volumes

Time-series forecasting involves using a mathematical model to predict future values of the variable of interest, based on previously observed values. Time-series plots or charts look very similar to simple linear regression scatterplots in that there are two variables: the response variable and the time variable. Beyond this similarity, there is hardly any other commonality between the two. Whereas regression analysis is often used in testing theories to see if current values of one or more explanatory variables explain (and hence predict) the response variable, time-series models are focused on extrapolating the time-varying behavior to estimate future values.

Time-series forecasting assumes that all the explanatory variables are aggregated and consumed in the response variable's time-variant behavior. Therefore, capturing time-variant behavior is a way to predict the future values of the response variable. To do that, the pattern is analyzed and decomposed into its main components: random variations, time trends, and seasonal cycles. The time-series example shown in Figure 5.10 illustrates all these distinct patterns.

The techniques used to develop time-series forecasts range from very simple (e.g., the naive forecast, which suggests that today's forecast

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

is the same as yesterday's actual) to very complex (e.g., ARIMA, which combines autoregressive and moving average patterns in data). The most popular techniques are perhaps the averaging methods that include simple averages, moving averages, weighted moving averages, and exponential smoothing. Many of these techniques also have advanced versions in which seasonality and trend can also be taken into account for better and more accurate forecasting. The accuracy of a method is usually assessed by computing its error (i.e., calculated deviation between actuals and forecasts for the past observations) via mean absolute error (MAE), mean squared error (MSE), or mean absolute percent error (MAPE). Even though they all use the same core error measure, these three assessment methods emphasize different aspects of the error, some panelizing larger errors more than the others.

Application Example: Data Mining for Complex Medical Procedures

Health care has become one of the most important issues related to quality of life in the United States and elsewhere around the world. While the demand for health care services is increasing because of the aging population, the supply side is having problems keeping up with the level and quality of service. In order to close the gap, health care systems need to significantly improve their operational effectiveness and efficiency. Effectiveness (i.e., doing the right thing, such as diagnosing and treating accurately) and efficiency (i.e., doing it the right way, such as using the least resources and time) are the two fundamental pillars on which the health care system can be revived. A promising way to improve health care is to take advantage of predictive modeling techniques along with large and feature-rich data sources (which provide true reflections of medical and health care experiences) to support accurate and timely decision making.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

According to the American Heart Association, cardiovascular disease (CVD) is the underlying cause of more than 20% of deaths in the United States. Since 1900, CVD has been the number-one killer every year except 1918, when the great flu pandemic claimed more lives. CVD kills more people than the next four leading causes of death—cancer, chronic lower respiratory disease, accidents, and diabetes mellitus—combined. More than half of all CVD deaths can be attributed to coronary diseases. Not only does CVD take a huge toll on the personal health and well-being of the population, it is also a great drain on the health care resources in the United States and elsewhere in the world. The direct and indirect costs associated with CVD for a year are estimated to exceed $500 billion.

A common surgical procedure to cure a large variant of CVD is called coronary artery bypass grafting (CABG). The cost of a CABG surgery depends on the patient and factors related to the service provider, but the average rate in the United States is between $50,000 and $100,000. As an illustrative example, Delen et al. (2012) carried out an analytics study in which they used various predictive modeling methods to predict the outcome of a CABG surgery, and they applied an information fusion–based sensitivity analysis on the trained models to better understand the importance of the prognostic factors. The main goal was to illustrate that predictive and explanatory analysis of large and feature-rich data sets provides invaluable information for making more efficient and effective decisions in health care.

Research Method

Figure 5.11 shows the model development and testing process Delen et al. used. They employed four different types of prediction models (artificial neural networks, support vector machines, and two types of decision trees: C5 and CART) and went through

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

a large number of experimental runs to calibrate the modeling parameters for each model type. Once the models were developed, Delen et al. used the models on the text data set. Finally, the trained models were exposed to a sensitivity analysis procedure to measure the contributions of the variables. Table 5.2 shows the test results for the four different types of prediction models.

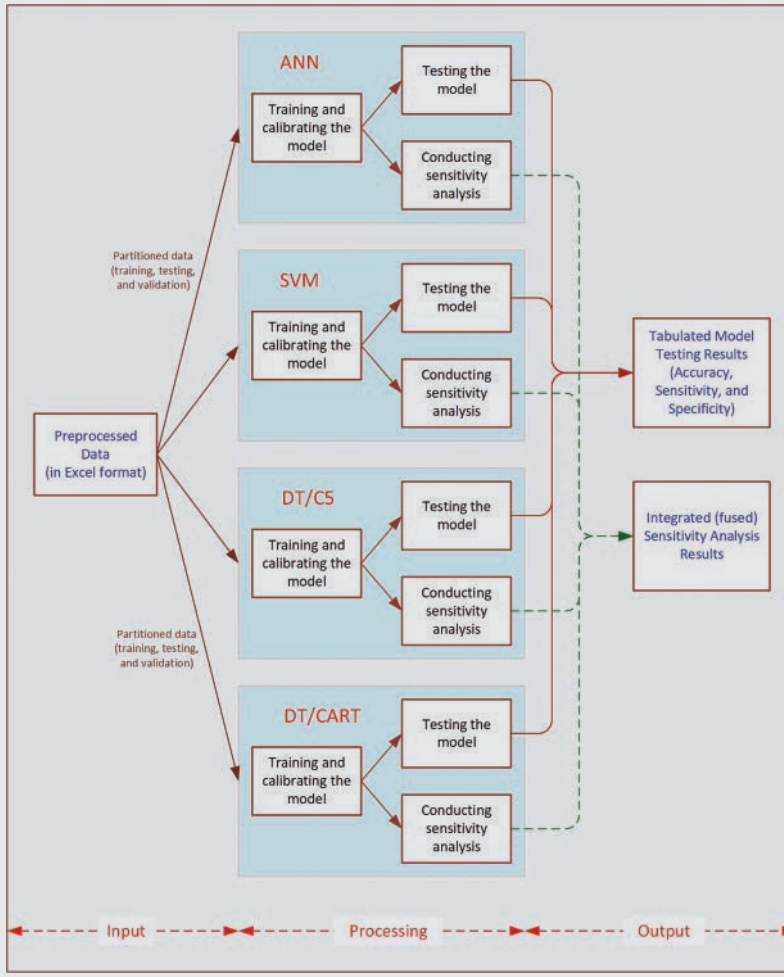


Figure 5.11 A Process Map for Training and Testing of the Four Predictive Models

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Results

In this study, Delen et al. showed the power of data mining in predicting the outcome and in analyzing the prognostic factors of complex medical procedures such as CABG surgery. They showed that using a number of prediction methods (as opposed to only one) in a competitive experimental setting has the potential to produce better predictive as well as explanatory results. Among the four methods they used, SVMs produced the best results, with prediction accuracy of 88% on the test data sample. The information fusion–based sensitivity analysis revealed the ranked importance of the independent variables. Some of the top variables identified in this analysis overlapped with the most important variables identified in previously conducted clinical and biological studies; this partial overlap confirms the validity and effectiveness of the proposed data mining methodology.

Table 5.2 Prediction Accuracy Results for All Four Model Types, Based on the Test Data Set

Model Type*		Confusion Matrices**		Accuracy†	Sensitivity†	Specificity†
		Pos (1)	Neg (0)			
ANN	Pos (1)	749	230	74.72%	76.51%	72.93%
	Neg (0)	265	714			
SVM	Pos (1)	876	103	87.74%	89.48%	86.01%
	Neg (0)	137	842			
C5	Pos (1)	876	103	79.62%	80.29%	78.96%
	Neg (0)	137	842			
CART	Pos (1)	660	319	71.15%	67.42%	74.87%
	Neg (0)	246	733			

* ANN: artificial neural network; SVM: support vector machine; C5: a popular decision tree algorithm; CART: classification and regression trees.

** Prediction results for the test data samples are shown in a confusion matrix, where the rows represent the actuals and columns represent the predicted cases.

† Accuracy, sensitivity, and specificity are the three performance measures that were used in comparing the four prediction models.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

From a managerial standpoint, clinical decision support systems that use the outcomes of data mining studies (such as the ones presented in this case study) are not meant to replace health care managers and/or medical professionals. Rather, they are meant to support them in making accurate and timely decisions to optimally allocate resources in order to increase the quantity and quality of medical services. We still have a long way to go before we will see these decision aids being used extensively in health care practices. Among others, there are behavioral, ethical, and political reasons for the resistance to adopting these tools. Perhaps need and government incentives for better health care systems will expedite their adoption.

Summary

Predictive analytics employs several algorithms, some seemingly too simple and others more complex; some algorithms have originated in traditional statistics, and others have their roots in emergent trends in artificial intelligence and machine learning. The algorithms covered in this chapter include Naive Bayes, *k*-nearest neighbor, artificial neural networks(ANNs), support vector machines (SVMs), linear regression, logistic regression, and time-series forecasting.

The goal of the chapter is to provide enough information about these algorithms (without getting too deep into the theory and mathematical complexities) that you can understand and appreciate how these algorithms really work—that is, how they do what they do. Knowing the inner workings of these algorithms, at least at a conceptual level, helps analytics professionals and data scientists make informed and smarter decisions related to the preprocessing of the data, framing of the problem, and creation of the solution methodology. An analogy might be a race car driver: Understanding how an engine works makes a better driver. Although the race car driver is not

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

expected to build or repair the engine, having an understanding of the inner workings of the engine helps the driver optimize its capacity by maximizing its power and productivity.

References

Aizerman, M., E. Braverman, & L. Rozonoer. (1964). "Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning," *Automation and Remote Control*, 25: 821–837.

Collins, E., S. Ghosh, & C. L. Scofield. (1988). "An Application of a Multiple Neural Network Learning System to Emulation of Mortgage Underwriting Judgments," *IEEE International Conference on Neural Networks*, 2: 459–466.

Das, R., I. Turkoglu, & A. Sengur. (2009). "Effective Diagnosis of Heart Disease Through Neural Networks Ensembles," *Expert Systems with Applications*, 36: 7675–7680.

Delen, D., A. Oztekin, & L. Tomak. (2012). "An Analytic Approach to Better Understanding and Management of Coronary Surgeries," *Decision Support Systems*, 52(3): 698–705.

Delen, D., R. Sharda, & M. Bessonov. (2006). "Identifying Significant Predictors of Injury Severity in Traffic Accidents Using a Series of Artificial Neural Networks," *Accident Analysis and Prevention*, 38(3): 434–444.

Delen, D., & E. Sirakaya. (2006). "Determining the Efficacy of Data-Mining Methods in Predicting Gaming Ballot Outcomes," *Journal of Hospitality & Tourism Research*, 30(3): 313–332.

Fadlalla, A., & C. Lin. (2001). "An Analysis of the Applications of Neural Networks in Finance," *Interfaces*, 31(4): 112–122.

Haykin, S. S. (2009). *Neural Networks and Learning Machines*, 3rd ed. Prentice Hall.

Hill, T., T. Marquez, M. O'Connor, & M. Remus. (1994). "Artificial Neural Network Models for Forecasting and Decision Making," *International Journal of Forecasting*, 10: 5–15.

Hopfield, J. (1982). "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Science*, 79(8): 2554–2558.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Liang, T. P. (1992). "A Composite Approach to Automated Knowledge Acquisition," *Management Science*, 38(1).

Loeffelholz, B., E. Bednar, & K. W. Bauer. (2009). "Predicting NBA Games Using Neural Networks," *Journal of Quantitative Analysis in Sports*, 5(1): 1–17.

McCulloch, W. S., & W. H. Pitts. (1943). "A Logical Calculus of the Ideas Imminent in Nervous Activity," *Bulletin of Mathematical Biophysics*, 5: 115–133.

Minsky, M., & S. Papert. (1969). *Perceptrons*. MIT Press.

Piatesky-Shapiro, G. (2013). *ISR: Microsoft Success Using Neural Network for Direct Marketing*. http://kdnuggets.com/news/94/n9.txt (accessed May 2020).

Principe, J. C., N. R. Euliano, & W. C. Lefebvre. (2000). *Neural and Adaptive Systems: Fundamentals Through Simulations*. Wiley.

Sirakaya, E., D. Delen, & H.-S. Choi. (2005). "Forecasting Gaming Referenda," *Annals of Tourism Research*, 32(1): 127–149.

Tang, Z., C. de Almeida, & P. Fishwick. (1991). "Time-Series Forecasting Using Neural Networks vs. Box-Jenkins Methodology," *Simulation*, 57(5): 303–310.

Walczak, S., W. E. Pofahi, & R. J. Scorpio. (2002). "A Decision Support Tool for Allocating Hospital Bed Resources and Determining Required Acuity of Care," *Decision Support Systems*, 34(4): 445–456.

Wallace, M. P. (2008, July). "Neural Networks and Their Applications in Finance," *Business Intelligence Journal*, pp. 67–76.

Wen, U.-P., K.-M. Lan, & H.-S. Shih. (2009). "A Review of Hopfield Neural Networks for Solving Mathematical Programming Problems," *European Journal of Operational Research*, 198: 675–687.

Wilson, C. I., & L. Threapleton. (2003, May 17–22). "Application of Artificial Intelligence for Predicting Beer Flavours from Chemical Analysis," *Proceedings of the 2th European Brewery Congress*, Dublin. http://neurosolutions.com/resources/apps/beer.html (accessed May 2020).

Wilson, R., & R. Sharda. (1994). "Bankruptcy Prediction Using Neural Networks," *Decision Support Systems*, 11(5): 545–557.

Zahedi, F. (1993). *Intelligent Systems for Business: Expert Systems with Neural Networks*. Wadsworth.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

This page intentionally left blank

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

6

Advanced Topics in Predictive Modeling

Predictive analytics is still as much art as it is science. For a given business problem and a set of real-world data, developing predictive analytics and machine learning models that are capable of producing highly accurate and unbiased predictions that are interpretable and explainable by humans while computationally efficient and easily deployable is a challenging task for any data scientist. Although we have seen significant improvements in the standardization and automation of methodologies (such as the CRISP methodology described in Chapter 3, "Standardized Processes for Predictive Analytics") and creation of best/better practices for model building in recent years, the true optimization of the complete data science/business analytics process still remains as a utopia. Many, including the author of this book, believe that at best we are halfway to achieving this endeavor. In this chapter, we cover some of the relatively novel concepts and approaches proposed to pave the road toward achieving the utopia. Specifically, this chapter covers ensemble modeling, the bias–variance trade-off, how to handle imbalanced data problems, and the human interpretability of machine learning models.

Model Ensembles

Ensembles (more appropriately called model ensembles or ensemble modeling) are relatively new tools in data science and predictive modeling. With ensemble modeling, outcomes produced by two or more analytics models are combined into a compound output.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Ensembles are primarily used for prediction modeling, where the scores of two or more models are combined to produce better predictions. The predictions can be either classification-type predictions (predicting a class label) or regression-/estimation-type predictions (estimating a numeric output variable).

Although use of ensembles has been dominated by prediction-type modeling, it can also be used for other analytics tasks, such as clustering and association rule mining. That is, model ensembles can be used for supervised as well as unsupervised machine learning tasks. Traditionally, these machine learning procedures focused on identifying and building the best possible model (often the most accurate predictor on the holdout data) from a large number of alternative model types. Analysts and scientists used an elaborate experimental process that mainly relied on trial and error to improve each single model's performance (defined by some predetermined metrics, such as, prediction accuracy) to its best possible level so that the best of the models could be used/deployed for the task at hand. The ensemble approach turns this thinking around. Rather than build models and select the single best model to use/deploy, the ensemble approach builds many models and uses them all for the task they are intended to perform (e.g., prediction).

Motivation: Why Do We Need Model Ensembles?

Usually researchers and practitioners build ensembles for two main reasons: better accuracy and more stable/robust/consistent/reliable outcomes. Numerous research studies and publications over the past two decades have shown that ensembles almost always improve predictive accuracy for the given problem and rarely predict more poorly than the single models (Abbott, 2014).

Ensembles began to appear in the data mining/analytics literature in the 1990s, motivated by the limited success obtained by the earlier works on combining forecasts. By the early to mid-2000s, ensembles became popular and almost essential to winning data mining and

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

predictive modeling competitions. One of the most popular examples of ensembles winning competitions is perhaps the famous Netflix case of an open competition that solicited researchers and practitioners to predict user ratings of films based on historical ratings. Netflix offered a prize of $1 million for the team that could reduce the root means squared error (RMSE) of the then-existing Netflix internal prediction algorithm by the largest margin but no less than 10 percentage points. The winner, runner-up, and nearly all the teams at the top of the leaderboard used model ensembles in their submissions. The winning submission was the result of an ensemble containing hundreds of predictive models.

When it comes to justifying the use of ensembles, Vorhies (2016) puts it the best: If you want to win a predictive analytics competition or at least get a respectable place on the leaderboard, you have to embrace and intelligently use model ensembles. Kaggle has become the premier platform for data scientists to showcase their talents. According to Vorhies, the Kaggle competitions are like Formula 1 racing for data science. Winners edge out competitors at the fourth decimal place, and their solutions are as distinct from the competitors' solutions as Formula 1 race cars are from the cars most of us drive. The amount of time devoted and the extreme techniques used would not always be appropriate for an ordinary data science production project, but like paddle shifters and exotic suspensions, some of those improvements and advanced features find their way into the day-to-day life and practice of analytics professionals.

In addition to Kaggle competitions, reputable associations such as the Association for Computing Machinery's (ACM's) Special Interest Group (SIG) on Knowledge Discovery and Data Mining (SIGKDD) and Pacific-Asia Conference in Knowledge Discovery and Data Mining (PAKDD) regularly organize competition (often called "cups") for the community of data scientists to demonstrate their competence—sometimes for monetary rewards but most often simply for bragging rights. Some of the popular analytics companies, such as SAS Institute

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

and Teradata Corporation, organize similar competitions for (and extend a variety of relatively modest awards to) both graduate and undergraduate students in universities all over the world, usually in concert with their regular analytics conferences.

It is not just their accuracy that makes model ensembles popular and unavoidably necessary. Yes, it has been shown time and time again that ensembles can improve model accuracy, but they can also improve model robustness, stability, and reliability. These advantages of model ensembles are perhaps even more important than accuracy in situations where reliable prediction is of the essence. A model ensemble combines multiple models into a single prediction outcome (using some form of averaging) so that no single model dominates the final predicted value of the models; this reduces the likelihood of making a way-off-target prediction. Figure 6.1 graphically illustrates a model ensemble for classification-type prediction problems. Although some varieties exist, most ensemble modeling methods follow this general process. From left to right, Figure 6.1 illustrates the general tasks of data acquisition and data preparation, followed by cross-validation and model building and testing, and, finally assembling/combining the individual model outcomes and assessing the resultant predictions.

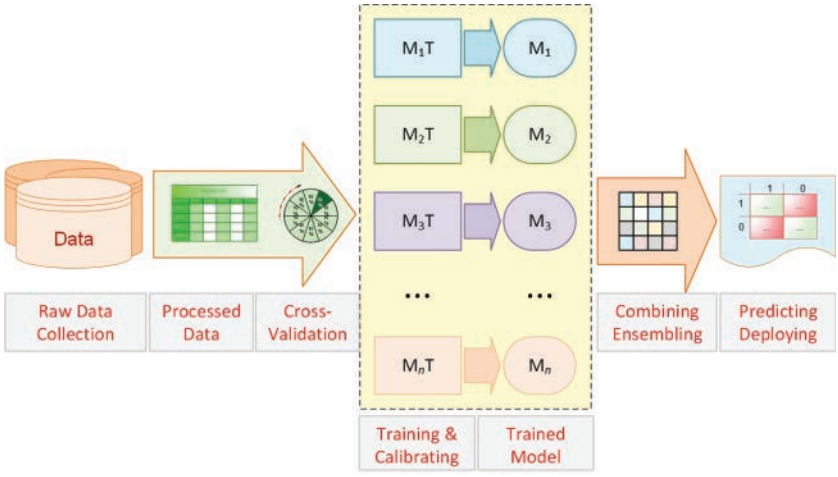


Figure 6.1 A Graphical Depiction of a Model Ensemble for Prediction Modeling

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Another way to look at ensembles is from the perspective of collective wisdom or crowdsourcing. In the popular book *The Wisdom of Crowds* (Surowiecki, 2005), the author proposes that a better decision can be made through crowdsourcing, where many (even uninformed) opinions are aggregated into a decision that is superior to the best expert's opinion. In his book, Surowiecki describes four characteristics necessary for the group opinion to work well and not degenerate into the opposite effect of the poor decisions sometimes made based on the "madness of crowds": diversity of opinion, independence, decentralization, and aggregation. The first three of these characteristics relate to how the individual decisions are made: Group members must have information different from what others in the group have and must not be affected by the others in the group. The last characteristic merely states that the decisions must be combined. These four principles/characteristics lay a foundation for building better model ensembles as well. Each predictive model has a voice in the final decision. The diversity of opinion can be measured by the correlation of the predictive values themselves; if all of the predictions are highly correlated with one another—that is, if the models nearly all agree—there is no foreseeable advantage in combining them. The decentralization characteristic can be achieved by resampling data or through case weights, where each model uses either different records from a common data set or at least uses the records with weights that are different from the other models (Abbott, 2014).

One of the prevalent concepts in statistics and predictive modeling that is highly relevant to model ensembles is the bias–variance trade-off. Before delving into the different types of model ensembles, it is necessary to review and understand the bias–variance trade-off principle (as it applies to the field of statistics or machine learning). In predictive analytics, *bias* refers to the error, and *variance* refers to the consistency (more accurately, the lack of consistency)

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

in predictive accuracy of models applied to other data sets. The best models are expected to have low bias (i.e., low error or high accuracy) and low variance (i.e., consistency of accuracy from data set to data set). Unfortunately, there is always a trade-off between these two metrics in building predictive models so that improving one results in worsening the other. You can achieve low bias on training data, but the model may suffer from high variance on holdout/validation data because the models may have been overtrained/overfit. For instance, the k-NN algorithm with $k = 1$ is an example of a low-bias model, which is perfect for a training data set, but it is susceptible to high variance on a test/validation data set. Use of cross-validation along with proper model ensembles seems to be the current best practice in handling the trade-off between bias and variance in predictive modeling.

Types of Model Ensembles

Using ensembles, or teams of predictive models working together, has been the fundamental strategy for developing accurate and robust analytics models. Although ensembles have been around for quite a while, their popularity and effectiveness have surfaced in a significant way only within the past decade, as they have continually improved in parallel with the rapidly improving software and hardware capabilities. *Model ensemble* makes many people think of decision tree ensembles such as random forest and boosted trees. However, generally speaking, model ensembles can be classified into four groups in two dimensions, as shown in Figure 6.2. The first dimension is the method type (the x-axis in Figure 6.2), where the ensembles can be grouped into bagging or boosting types. The second dimension is the model type (the y-axis in Figure 6.2), where the ensembles can be grouped into homogeneous or heterogeneous types (Abbott, 2014).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

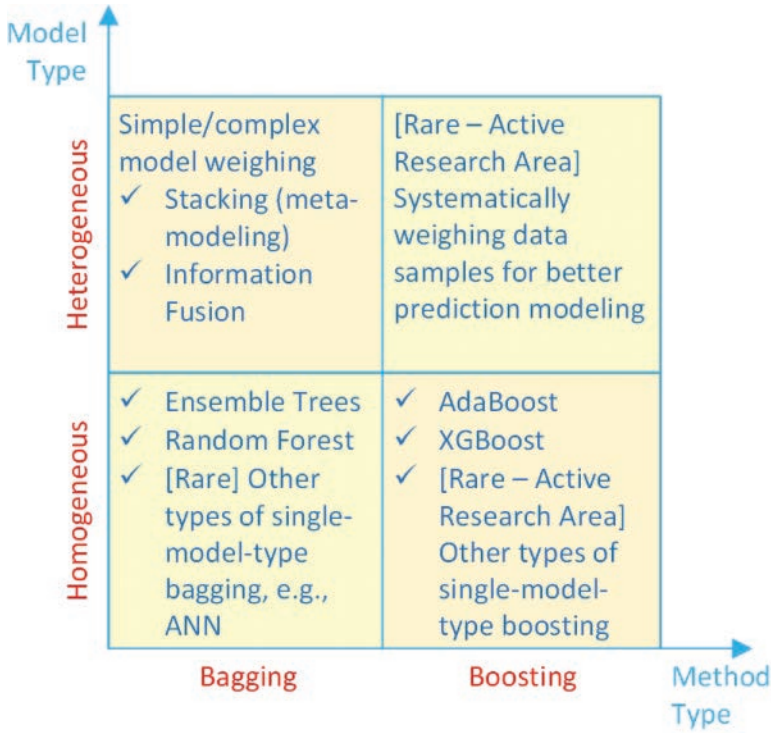


Figure 6.2 A Simple Taxonomy for Model Ensembles

As their name implies, *homogeneous* ensembles combine the outcomes of two or more models of the same type, such as decision trees. In fact, a vast majority of homogeneous ensembles are developed using a combination of decision tree structures. The two most common categories of homogeneous ensembles that use decision trees are bagging and boosting (described in detail later in this chapter). *Heterogeneous* ensembles, as you would guess, combine the outcomes of two or more different types of models, such as decision trees, artificial neural networks, logistic regression, and support vector machines. One of the key success factors in ensemble modeling is to use models that are fundamentally different from one another, that look at the data from different perspectives; this is similar to the idea of crowdsourcing. Because of the way they combine the outcomes of different

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

model types, heterogeneous ensembles are also called information fusion models (Delen & Sharda, 2010) or stacking models (both discussed later in this chapter).

Bagging

Bagging is the simplest and most common ensemble method. Leo Breiman, a very well-respected scholar in the world of statistics and analytics, first published a description of the bagging (short for bootstrap aggregating) algorithm at Berkeley in 1996 (Breiman, 1996). The idea behind begging is quite simple yet powerful: Build multiple decision trees from resampled data and combine the predicted values through averaging or voting. The resampling method Breiman used was bootstrap sampling (i.e., sampling with replacement), which creates replicates of some records in the training data. With this selection method, on average, about 37% of the records will not be included at all in the training data set (Abbott, 2014).

Although bagging was first developed for decision trees, the idea can be applied to any predictive modeling algorithm that produces outcomes with sufficient variation in the predicted values. Although rare in practice, the other predictive modeling algorithms that are potential candidates for bagging-type model ensembles include neural networks, Naive Bayes, *k*-nearest neighbor (for low values of *k*), and, to a lesser degree, logistic regression. *k*-nearest neighbor is not a good candidate for bagging if the value of *k* is already large; the algorithm votes or averages predictions, and with larger values of *k*, predictions are already very stable and have low variance.

Bagging can be used for both classification and regression/estimation prediction problems. In classification-type prediction problems, all of the participant models' outcomes (class assignments) are combined by using either a simple or complex/weighted majority voting mechanism. The class label that gets the most/highest votes becomes the aggregated/ensemble prediction for that sample/record. In a

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

regression-/estimation-type prediction problem where the output/target variable is a number, all of the participant models' outcomes (numeric estimations) are combined using either a simple or complex/weighted averaging mechanism. Figure 6.3 provides a graphical depiction of a decision tree bagging algorithm.

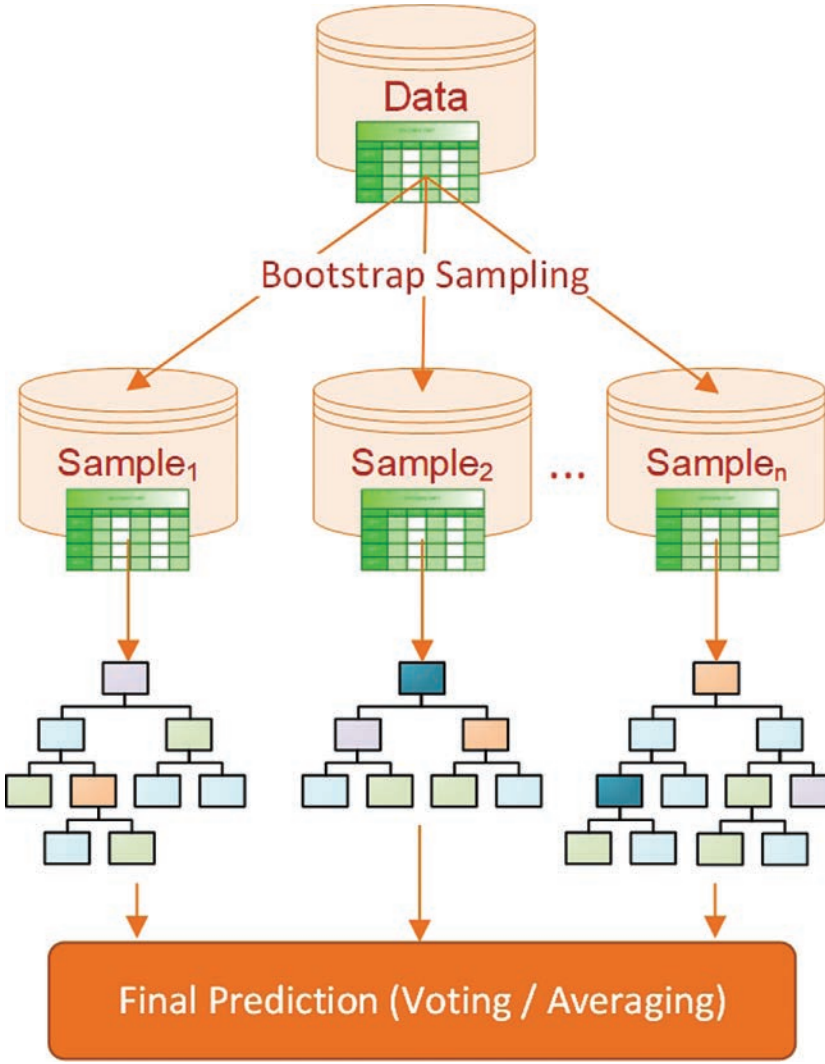


Figure 6.3 Bagging Decision Tree Ensembles

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

One of the key questions in bagging is "How many bootstrap samples, also called replicates, should be created?" According to Breiman, "my sense of it is that fewer are required when *y* [dependent variable] is numerical and more are required with an increasing number of classes [for classification-type prediction problems]." Breiman typically used 10 to 25 bootstrap replicates and found significant improvements with as few as 10 replicates. Overfitting the models is an important requirement for building good bagged ensembles. Overfitting each model leads to low bias, but the decision tree generally ends up with worse accuracy on held-out data. However, bagging is a variance-reduction technique; the averaging of predictions smooths the predictions so that they behave in a more stable (i.e., less-variant) way on new data.

As mentioned earlier, the diversity of model predictions is a key factor in creating effective ensembles. One way to measure the diversity of predictions is to examine the correlation of predicted values. If the correlations between model predictions are always very high (i.e., greater than 0.95), there is little additional predictive information each model brings to the ensemble, and therefore little improvement in accuracy is achievable. Generally, it is best to have correlations less than 0.9 at most. The correlations should be computed from the model propensities or predicted probabilities rather than from the {0,1} classification value. Bootstrap sampling in bagging is the key to introducing diversity in the models. You can think of the bootstrap sampling methodology as creating case weights for each record; some records are included multiple times in the training data (i.e., their weights are 1, 2, 3, or more), and others are not included at all (i.e., their weights are equal to 0) (Abbott, 2014).

Boosting

Boosting is another ensemble method, and it is perhaps the most common method after bagging. Yoav Freund and Robert E. Schapire

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

first introduced the boosting algorithm in separate publications in the early 1990s, and then, in a 1996 joint publication (Yoav & Schapire, 1996), they introduced the well-known boosting algorithm AdaBoost. The idea behind boosting is quite straightforward. First, you build a rather simple classification model; the model only needs to be slightly better than random chance, so for a binary classification problem, it only needs to be slightly better than a 50% correct classification. In the first pass, the records used in the algorithm have equal case weights, as would normally be done in a predictive model. The errors in the predicted values for each case are noted. The correctly classified records/cases/samples have their case weights stay the same, or perhaps reduced, and the records that are incorrectly classified have their case weights increased. Then a second simple model is built on these weighted cases (i.e., the transformed/weighted training data set). In other words, for the second model, records that were incorrectly classified are "boosted" through case weights to be considered more strongly or seriously in the construction of the new prediction model. In each iteration, the records that are incorrectly predicted (i.e., the ones that are difficult to classify) get their case weights increased so that the algorithm pays more attention to these records until, hopefully, they are finally classified correctly.

This process of boosting is often repeated tens or even hundreds of times. After the tens or hundreds of iterations, the final predictions are made based on a weighted average of the predictions from all the models. Figure 6.4 illustrates the simple process of boosting in building decision tree–type model ensembles. As shown, each tree takes the most current data set (i.e., the one with the most recently boosted case weights) to build another tree. The feedback from incorrectly predicted cases is used as an indicator to determine which cases to boost and to what extent (in direction and magnitude) for the training samples/cases.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

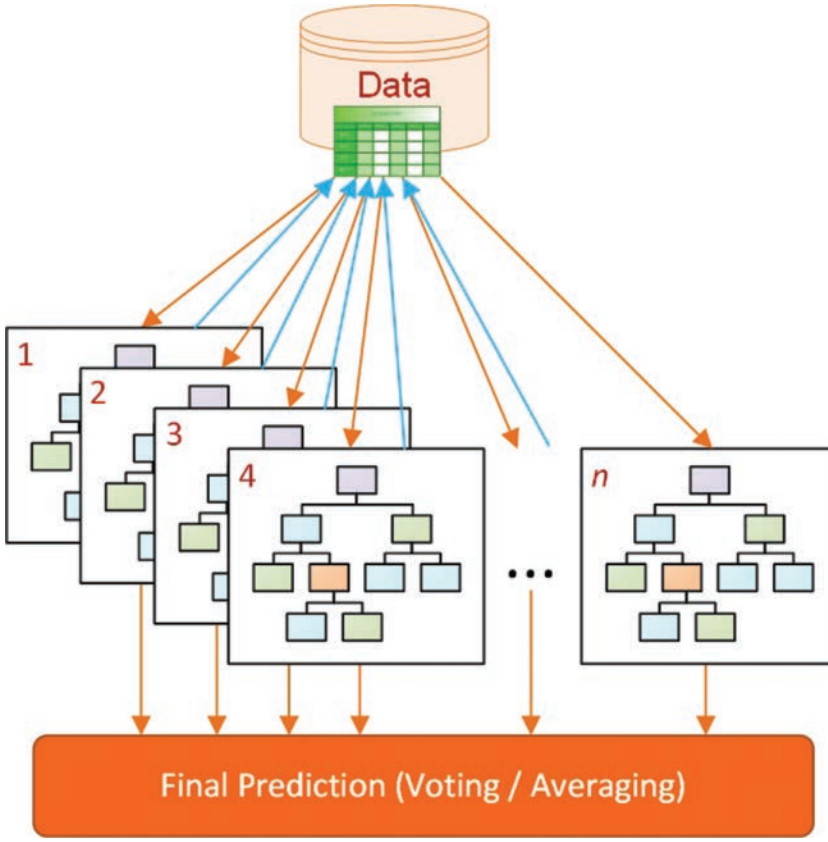


Figure 6.4 Boosting-Type Ensembles for Decision Trees

Although they look quite similar in structure and purpose, bagging and boosting employ slightly different strategies to utilize the training data set and to achieve the goal of building the best possible prediction ensemble. There are two key differences between bagging and boosting. First, bagging uses a bootstrap sample of the cases to build decision trees, whereas boosting uses the complete training data set. Second, whereas bagging creates independent, simple trees for the ensemble, boosting creates dependent trees (with each tree "learning" from the previous one to pay more attention to the incorrectly predicted cases) that collectively contribute to the final ensemble.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Boosting methods are designed to work with weak learners (i.e., simple models); the component models in a boosted ensemble are simple models with high bias and low variance. The improvement with boosting is better, as with bagging, when algorithms that are unstable predictors are used. Decision trees are most often used in boosted models. Naive Bayes is also used with boosting but produced rather marginal improvements over a single model. Empirically speaking, boosting typically produces better model accuracy than single decision trees and bagging-type ensembles.

Variants of Bagging and Boosting

Bagging and boosting were the first ensemble methods that appeared in predictive analytics software (primarily with decision tree algorithms). Since their introduction, many other approaches to building ensembles have been developed and made available, particularly in open source software (both as part of open analytics platforms like KNIME and Orange and as class libraries in R and Python). The most popular and successful (advanced) variants of bagging and boosting are random forest and stochastic gradient boosting, respectively.

Random Forests

Breiman (2000) first introduced random forest (RF) as a modification to the simple bagging algorithm. As with bagging, the RF algorithm begins with bootstrap sampled data, and one decision tree is built from each bootstrap sample. Compared to simple bagging, there is, however, an important twist to the random forest algorithm: At each split in the tree, starting from the very first split, rather than considering all input variables as candidates, only a random subset of variables is considered. Hence, in random forest, the bootstrap sampling technique applies to both random selection of cases and random selection of features (i.e., input variables).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The number of cases and the number of variables to consider, along with how many trees to construct are parameters to consider in building random forest models. Typically, the default number of variables to consider as candidates at each split point is the square root of the total number of candidate inputs. For example, if there are 100 candidate inputs for the model, a random 10 inputs are candidates for each split. This means that it is unlikely that the same inputs will be available for splits at parent and child nodes in a given tree, so the tree will be forced to find alternative ways to maximize the accuracy of subsequent splits. Therefore, there is an intentionally created two-fold diversity mechanism built into the tree construction process: random selection of cases and variables. Random forest models produce prediction outcomes that are usually more accurate than simple bagging and are often more accurate than simple boosting (i.e., AdaBoost).

Stochastic Gradient Boosting

AdaBoost is only one of many boosting algorithms currently documented in the literature. In commercial software, AdaBoost is still the most commonly used boosting technique, but dozens of boosting variants can be found in open source software packages. One interesting boosting algorithm that has recently gained popularity due to its superior performance is the stochastic gradient boosting (SGB) algorithm created by Jerry Friedman at Stanford University. Friedman (2001) also developed an advanced version of the AdaBoost algorithm that he called multiple additive regression trees (MART), which was later branded as TreeNet by Salford Systems in its software tool. Like other boosting algorithms, the MART algorithm builds successive, simple trees and combines them additively. Typically, the simple trees are more than stumps; each tree usually contains up to six terminal nodes.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

After building the first tree, errors (also called *residuals*) are computed. The second tree and all subsequent trees then use residuals as the target variable. Subsequent trees identify patterns that relate inputs to small and large errors. Poor prediction of the errors results in large errors to predict in the next tree, and good predictions of the errors result in small errors to predict in the next tree. Typically, hundreds of trees are built, and the final predictions are an additive combination of the predictions that is, interestingly, a piecewise constant model because each tree is itself a piecewise constant model. However, hundreds of trees are typically included in the ensemble (Abbott, 2014). The TreeNet algorithm has won multiple data modeling competitions since its introduction and has proven to be an accurate predictor; in addition, very little data cleaning is needed for the trees before modeling.

Stacking

Stacking (a.k.a. stacked generalization or super learner) is a heterogeneous ensemble method. Some analytics professionals believe it is the best ensemble technique, but it is also the least understood (and the most difficult to explain). Due to its two-step model training procedure, stacking can seem to be an overly complicated form of ensemble modeling. Basically, stacking creates an ensemble from a diverse group of strong learners. In the process, it interjects a metadata step called a "super learner" or "meta learner." The intermediate meta-classifiers forecast how accurately the primary classifiers have been and are used as the basis for adjustments and corrections (Vorhies, 2016). The process of stacking is figuratively illustrated in Figure 6.5.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

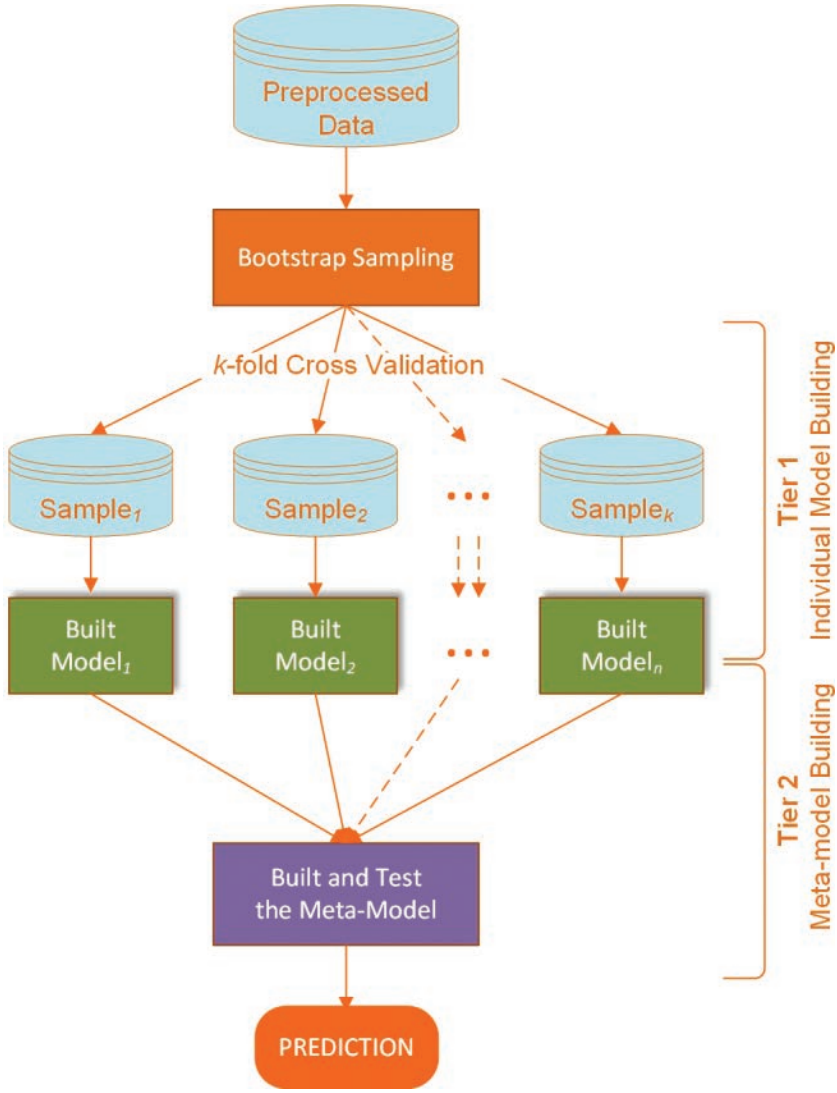


Figure 6.5 Stacking Type Model Ensembles

As shown in Figure 6.5, in constructing a stacking type model ensemble, a number of diverse strong classifiers are first trained using bootstrapped samples of the training data, creating Tier 1 classifiers (each optimized to its full potential for the best possible prediction outcomes). The outputs of the Tier 1 classifiers are then used to train a Tier 2 classifier (i.e., a meta-classifier) (Wolpert, 1992). The idea is

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

to determine whether training data have been properly learned. For example, if a particular classifier incorrectly learned a certain region of the feature space, and it consistently misclassifies instances coming from that region, then the Tier 2 classifier may be able to learn this behavior, and along with the learned behaviors of other classifiers, it can correct such improper training. Cross-validation selection is typically used for training the Tier 1 classifiers: The entire training data set is divided into k mutually exclusive subsets, and each Tier 1 classifier is trained on (a different set of) $k - 1$ subsets of the training data. Each classifier is then evaluated on the kth subset, which was not seen during training. The outputs of these classifiers on their pseudo-training blocks, along with the actual correct labels for those blocks, constitute the training data set for the Tier 2 classifier.

Information Fusion

As part of heterogeneous model ensembles, information fusion combines (fuses) the output (i.e., predictions) of different types of models, such as decision trees, artificial neural networks, logistic regression, support vector machines, Naive Bayes, k-nearest neighbor, and their variants, among other. There is a difference between stacking and information fusion in that information fusion has no "meta-modeling" or "super-learners." It simply combines the outcomes of heterogeneous strong classifiers using simple or weighted voting (for classification) or simple or weighted averaging (for regression). Therefore, it is simpler and less computationally demanding than stacking. In the process of combining the outcomes of multiple models, either simple voting (where each model contributes one vote) or a weighted combination of voting (where each model contributes based on its prediction accuracy, so that more accurate models have higher weight values) can be used. Regardless of the combination method, heterogeneous ensembles have been shown to be invaluable additions to data mining and predictive modeling projects.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Figure 6.6 graphically illustrates the process of building information fusion model ensembles.

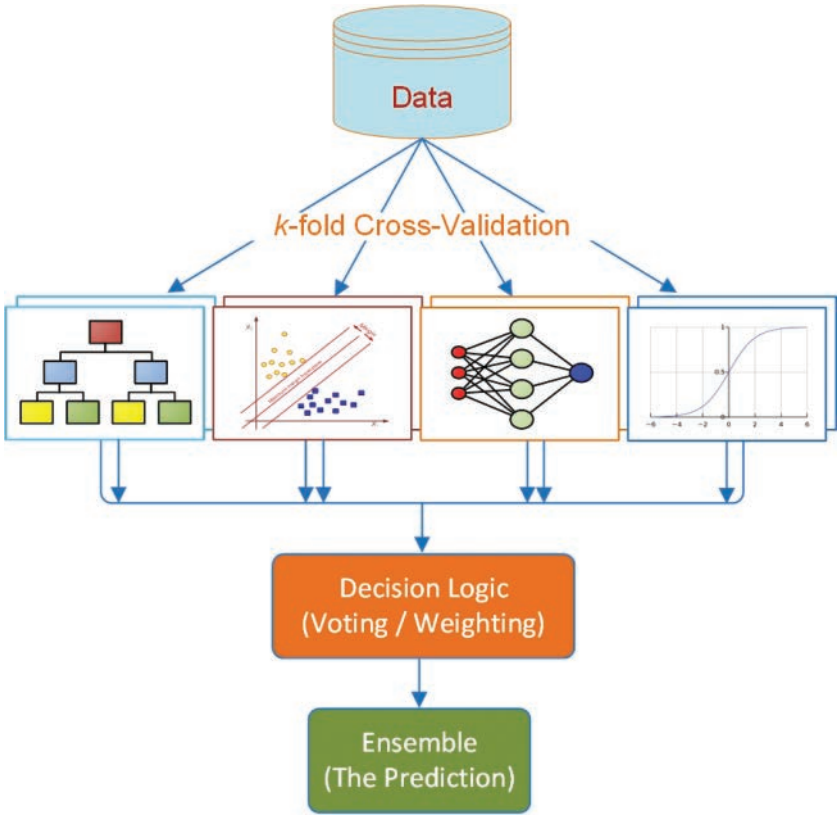


Figure 6.6 An Illustration of the Building Process for Information Fusion Model Ensembles

Ensembles Are Not Perfect!

When you are asked to build a prediction model (or any other analytics model for that matter), you are expected to develop some of the popular model ensembles along with the standard individual models. Ensembles are often more accurate and almost always more robust and reliable than individual models. Although they seem like silver bullets, model ensembles are not without shortcomings. Their most common and important shortcomings are complexity and lack of transparency.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Complexity

Model ensembles are more complex than individual models. The core data science principle Occam's razor says that simpler models are more likely to generalize better, so it is better to reduce/regularize complexity or, in other words, to simplify the models so that the inclusion of each term, coefficient, or split in a model is justified by its power of reducing the error a sufficient amount. One way to quantify the relationship between accuracy and complexity is taken from information theory in the form of information theoretic criteria, such as the Akaike information criterion (AIC), the Bayesian information criterion (BIC), and minimum description length (MDL). Data scientists have borrowed these criteria from statisticians and are using them for variable selection in predictive modeling. Information theoretic criteria require a reduction in model error to justify additional model complexity. Do model ensembles violate Occam's razor? Ensembles, after all, are much more complex than single models. According to Abbott (2014), if the accuracy of an ensemble is better on held-out data than the accuracy of a single model, then the answer is no—as long as we think of model complexity in terms of not just computational complexity but behavioral complexity. Therefore, we should not fear that adding computational complexity (more terms, splits, or weights) will necessarily increase the complexity of models because sometimes an ensemble significantly reduces the behavioral complexity.

Transparency

The interpretation of ensembles can become quite difficult. If you build a random forest ensemble containing 200 trees, how do you describe why a prediction has a particular value? You can examine each of the trees individually, though doing so is clearly not practical. For this reason, ensembles are often considered black-box models, meaning that what they do is not transparent to the modeler or domain expert. However, you can look at the split statistics (which variables

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

are more often picked to split early in those 200 trees) to artificially judge the level of contribution (a pseudo variable's importance) each variable makes to the trained model ensemble. Compared to a single decision tree, this is still difficult, and it is not a very intuitive way to interpret how a model performs. Another way to determine which inputs to the model are most important is to perform a sensitivity analysis.

In addition to the complexity and transparency issues, model ensembles can also be difficult and computationally expensive to build and deploy. Table 6.1 shows the pros and cons of model ensembles as they compared to individual models.

Table 6.1 Advantages and Shortcomings of Model Ensembles

Advantages	**Description**
Accuracy	Model ensembles usually provide more accuracy than do individuals models.
Robustness	Model ensembles tend to be more robust against outliers and noise in the data than are individual models.
Reliability (stable)	Because of the variance reduction, model ensembles tend to produce more stable, reliable, and believable results than do individual models.
Coverage	Model ensembles tend to have better coverage of the hidden complex patterns in the data sets than do individual models.
Shortcomings	**Description**
Complexity	Model ensembles are much more complex than individual models.
Computationally expensive	Compared to building individual models, building ensembles requires more time and computational power.
Lack of transparency (explainability)	Because of their complexity, it is harder to understand the inner structure of model ensembles (how they do what they do) than it is to understand the structure of individual models.
Harder to deploy	Model ensembles are much harder to deploy in an analytics-based managerial decision system than are individual models.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

In summary, model ensembles are the new frontier for predictive modelers who are interested in accuracy as they either reduce errors in the models or reduce the risk that the models will behave erratically. Evidence for this is clear from the dominance of ensembles in predictive analytics and data mining competitions: ensembles always win.

The good news for predictive modelers is that many techniques for building ensembles are built into software already. The most popular ensemble algorithms (i.e., bagging, boosting, and stacking and their variants) are available in nearly every commercial and open source software tool. Building customized ensembles is also supported in many software products, whether based on a single algorithm or a heterogeneous ensemble.

Ensembles aren't appropriate for every solution; the applicability of ensembles is determined by the modeling objectives defined during business understanding and problem definition. However, ensembles should be part of every predictive modeler's and data scientist's modeling arsenal.

Bias–Variance Trade-off in Predictive Analytics

In developing machine learning models for predictive analytics, there is always a trade-off between simplicity/parsimony and complexity, generalizability (i.e., capture of the signal) and overfitting (including the noise), predictive accuracy and explainability, overtraining/curve-fitting and undertraining, training error and validation error. The bias–variance is a rather common concept in statistics and machine learning that generally deals with the assessment of a trained predictive model's usefulness and accuracy. It is considered a crucial issue in predictive modeling, data mining, machine learning, and supervised learning. Although the best model is achieved when both

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

bias and variance are minimized, there is, as the word *trade-off* in the name implies, a balancing act. The concepts bias and variance oppose each other; they do not go in the same direction, and improving one worsens the other. Finding the optimal combination of bias and variance to achieve the best predictive model for a given problem and data combination is anything but a trivial process; it often requires consideration of best (or better) practices along with extensive experimentation and comparative analyses; even then, the obtained model cannot be confirmed as globally "optimal."

What Are Bias and Variance?

Bias is the difference between a model's prediction output and the ground truth (i.e., the real/actual values) for a given prediction problem. It is the training error. A model with high bias captures very little of the desired signal (i.e., patterns and relationships) in the training data and therefore produces an oversimplified/underfit model. As a result, such a model leads to high error values on both training and testing data.

Variance, on the other hand, is a testing error. It is the variability of a model's predictions for a given data set that characterizes the robustness of the prediction model. A model with high variance is the one that captures not only the signal (i.e., generalized patterns and relationships) in the data but also the noise and therefore produces an overly complex/overfit model. As a result, such a model performs well on training data but does very poorly on test data.

To graphically and intuitively illustrate the trade-off between bias and variance, we often use bullseye diagrams like the ones shown in Figure 6.7. Bias values are depicted on the y-axis, and variance values are depicted on the x-axis. For simplicity, Figure 6.7 is split into four quadrants (Q1 through Q4) in a two-dimensional Cartesian coordinate system along the dichotomized values of the x- and y-axes to

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

show low and high. The simple interpretations of these quadrants are as follows:

- **Q1: High bias, low variance.** The predictions are consistent but are off-target/inaccurate (representing a consistently underperforming prediction model).
- **Q2: High bias, high variance.** The predictions are neither consistent nor accurate (representing an underperforming and inconsistent prediction model).
- **Q3: Low bias, low variance.** The predictions are both consistent and accurate (representing the ideal prediction model).
- **Q4: Low bias, high variance.** The predictions are inconsistent but somewhat accurate (representing a reasonably well-performing but inconsistent prediction model).

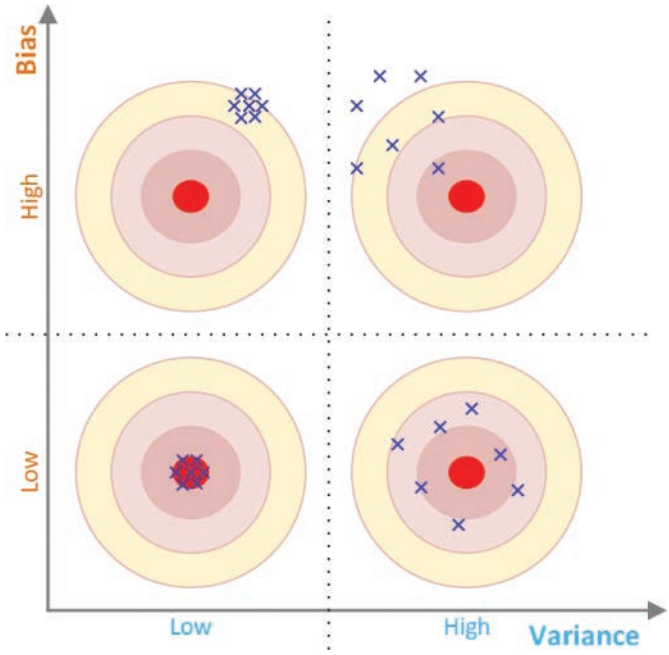


Figure 6.7 A Bullseye Diagram Illustration of the Bias–Variance Trade-off

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Overfitting Versus Underfitting

The process of training a predictive model is often characterized as "fitting" (e.g., fitting a model to a given data set). Figure 6.8 shows the three levels of fitting (also called training or learning)—overfitting, underfitting, and balanced fitting—in a two-dimensional feature space. Overfitting (also called overtraining) is the notion of making the model too specific to the training data), and by doing so, capturing not only the signal but also the noise in the data set. As a result, the overtrained model (especially in the case of highly parametrized model types, such as neural networks) perform exceptionally well (at the "too good to be true" level) on the training data and do very poorly on the test data. This phenomenon is also commonly characterized on the bias–variance trade-off continuum as low-bias/high-variance outcome.



Figure 6.8 The Three Levels of Fitting a Model to the Training Data

On the other hand, underfitting (also called undertraining) is the notion of leaving the model too generic by not completing the training (i.e., the learning process) from the training data (i.e., not capturing the signal inherent into the training data) and, therefore, not performing as well as expected on the training data (or the testing data). (Although an undertrained model may perform equally well on the test data because of its poor performance on the training data, such performance may not be nearly sufficient for any practical purposes.) Underfitting is commonly characterized on the bias–variance trade-off continuum as high-bias/low-variance outcome.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

What is desired for a predictive model is to avoid both overtraining and undertraining and, by doing so, to achieve an optimal balance in fitting the model to the available training data.

How to Determine the Optimal Level of Training

Most machine learning tools are formulated to learn from data in an iterative manner. With each iteration, a machine learning tool tries to incorporate more and more details deduced/inferred from the data in order to minimize the error. Since predictive models are at the core of supervised learning—part of the machine learning family of methods—during training/learning, they aim to minimize the error (difference or delta) between the model output and the actual output. Each iteration focuses on reducing this error by adjusting the model parameters. For instance, in artificial neural networks, the learning process aims to minimize the difference between model output (the value at the output layer) and the actual output of the data. Similarly, in decision tree induction, the learning process keeps growing the tree with new branches in order to purify the outputs at the leaf nodes (to achieve the perfect accuracy or matching).

Figure 6.9 graphically illustrates the change in the prediction error (and its reflection in the bias and variance curves/values) on the *y*-axis as the complexity of the model (and the training/learning process) progressively changes/evolves. As shown in the figure, at the left end of the *x*-axis, the simplest, least complex models—the undertrained, underlearned, underfitted models—have large error values (i.e., low prediction accuracy on the training), a high level of bias, and a low level of variance. The most complex models, at the right end of the *x*-axis—the overtrained, overlearned, overfitted models—have very low error values (i.e., low prediction accuracy on the training data), a low level of bias, and a high level of variance.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

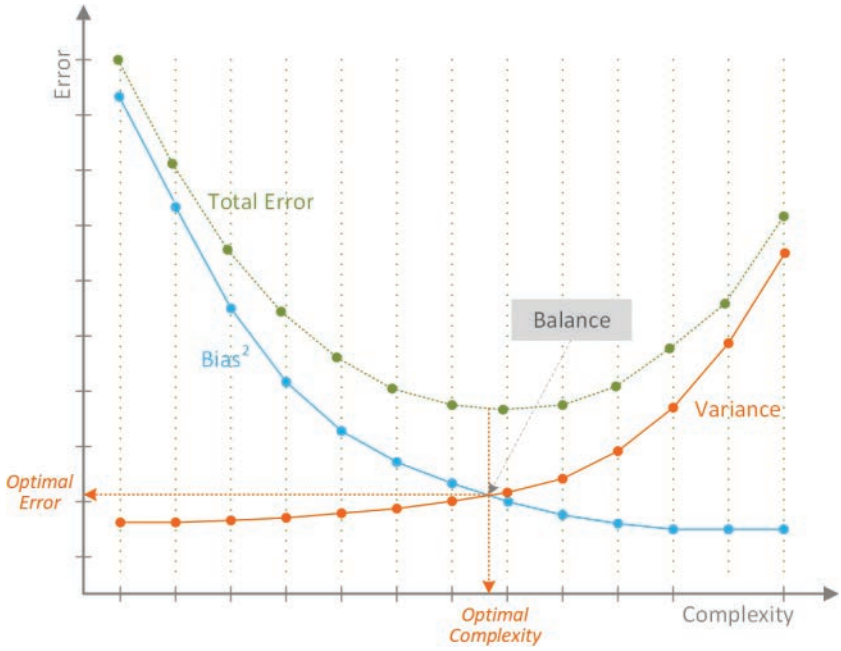


Figure 6.9 The Nonlinear Relationship Between Error/Complexity and Bias/Variance

Building a good prediction model requires finding an optimal balance between the errors related to bias and variance. The optimal balance between the bias and variance can be determined by summing the error values that represent the two curves into a total error curve (with a concave shape) and then identifying/calculating the lowest point in the curve. This lowest point can be used to find the corresponding "optimal" values in terms of complexity and the corresponding best achievable error values. The individual errors and the total error associated with each of the three curves in Figure 6.9 can be formulated mathematically as follows:

$$Error(x) = Bias^2 + Variance + Noise$$

$$Error(x) = \left(E\left[\hat{f}(x)\right] - f(x)\right)^2 + E\left[\left(\hat{f}(x) - E\left[\hat{f}(x)\right]\right)^2\right] + \sigma_\varepsilon^2$$

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Error(x) is the sum of the three error terms: Bias^2, Variance, and Noise (i.e., the irreducible error). Irreducible error, as the name implies, cannot be reduced further through more training and use of more sophistication in the model. It is considered the noise in the model that you have to live with, and it is usually due to missing (unrepresented) information in the variable space. In real-world problems, which are complex and need to be simplified via modeling assumptions, there will always be some degree of missing information (input variables), and hence, a model will never be perfectly or completely able to produce zero noise.

Imbalanced Data Problems in Predictive Analytics

Developing prediction models using imbalanced data sets is one of the most critical challenges in the field of data mining and decision analytics. A data set is *imbalanced* when the distribution of different classes in the target variable are significantly dissimilar. For instance, in the case of binary classification problems in medical testing, the data may have many more examples that belong to one class (e.g., negative examples) compared to the other class (e.g., positive examples). Usually, we call the class with fewer examples the minority class and the class with more examples the majority class. Although most of the discussion on imbalanced data problems centers around the binary classification problems, the phenomenon also applies to multiclass classification problems as well as to problems that deal with numeric target variables (i.e., regression- or estimation-type prediction problems). Decision modeling based on imbalanced data sets is quite common in real-world problems, including medical diagnosis, fraud detection, churn analysis, email spam identification, outlier detection, bankruptcy prediction, and network intrusion detection.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The issues related to imbalanced data problems are especially obvious in the development of clinical decision support systems where the cases with a rare disease would usually constitute a very small proportion of the total patient population. For instance, a study by Piri et al. (2017) involved the development of a clinical decision support system for diagnosing diabetic retinopathy patients using only a small set of diabetes test results and patient demographic information (contained within the EHR database). In this data set—a population of several hundred thousand diabetic patients—only about 5% of the patients had diabetic retinopathy, and the remaining 95% did not have the disease.

A less severely imbalanced data problem occurred in a decision support system developed for educational institutions (Thammasiri et al., 2014). The main goals of this educational decision support system were to identify, explain, and mitigate freshmen student attrition at a higher education institution. This student attrition prediction problem involved a data set that contained about four times more negative samples (freshmen students who did not drop out) than positive samples (freshmen students who did drop out). Figure 6.10 graphically illustrates the data imbalance situation dealt with while developing this system. As shown on the right side of the figure (the "model assessment" pane), the imbalanced data produced very high predictive accuracy on the majority class but performed very poorly on the minority class; balanced data produced significantly better prediction accuracy for both minority and majority classes. Although the overall accuracy achieved with an imbalanced data set seems to be scientifically higher, this is a rather deceptive perception and is often characterized as "fool's gold" in data mining literature. Keep in mind that a balanced data set allows machine learning techniques to produce reasonably high prediction accuracy for both classes.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

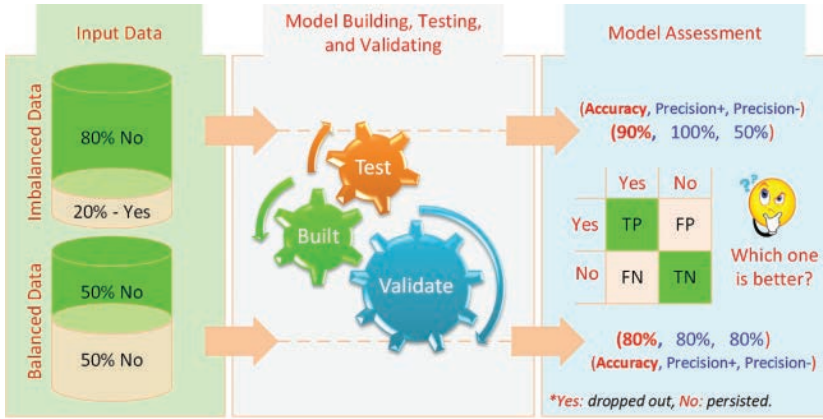


Figure 6.10 A Graphical Depiction of an Imbalanced Data Problem

But why does an imbalanced data set lead to poor prediction outcomes? Machine learning methods learn from the training data in a manner somewhat similar to the way that humans learn from their experiences. That is, in humans (and perhaps in most animals as well), memory and recall are largely determined by repeated events and experiences. Frequently observed experiences lead to more permanent and vivid memories and lead to easier and more accurate recall and recognition. In contrast, experiences that are observed less frequently are often overlooked or simply ignored, and hence they are often not accurately recalled when needed. In machine learning, a similar learning process takes place, where the method produces biased predictive models that focus on the majority class patterns while conveniently ignoring the minority class specifics.

How to Solve Imbalanced Data Problems

There are various ways to mitigate imbalanced data problems and thereby improve the performance of a given predictive model. Figure 6.11 shows a simple taxonomy of the methods used to handle imbalanced data problems. Although these methods may differ

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

in level of algorithmic sophistication, the goal of all of these methods is to make sure the prediction method (or the underlying algorithm) pays nearly equal/fair attention to patterns presented by all classes—majority, minority, and anything in between. While some of these methods try to solve imbalanced data problems by simply manipulating the distribution of the classes in the training data, others try to manipulate the learning process by adjusting the underlying algorithm or adjusting the cost of performance to favor the accurate prediction of the minority class.

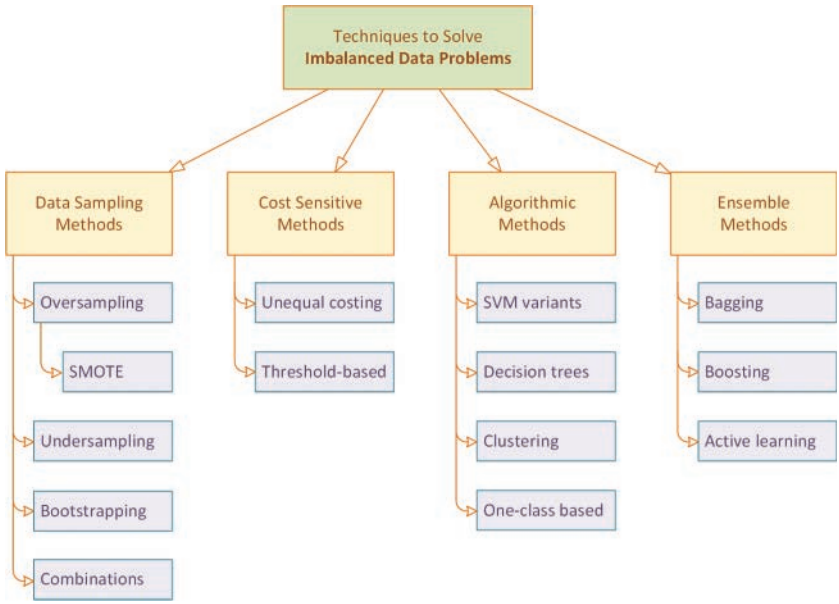


Figure 6.11 A Taxonomy of Methods Used to Handle the Imbalanced Data Problems

Data sampling methods—the most widely used class of methods in data science and machine learning because of their ease of understanding, formulation, and implementation—either increase the number of minority examples by generating additional examples (collectively called oversampling methods) or decrease the number of majority examples by removing some of them (usually called undersampling methods).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Oversampling methods increase the number of examples in the minority class in one of two ways: (1) replicating the examples in the minority class until they are equal to the number of examples in the majority class (via a simple copy and paste or bootstrapping-type random sampling) or (2) synthetically creating new examples that are not the same but are very similar to the samples in the minority class. A popular technique called SMOTE (Synthetic Minority Oversampling Technique) synthetically creates similar minority class examples by using the *k*-nearest neighbor algorithm to increase the number of examples in the minority class. SMOTE works quite well with data sets that are dominated by genuinely numeric input variables; on the other hand, it does a very poor job with data sets that are dominated by categorical/nominal variables. There have been numerous attempts to create variants of SMOTE algorithms to address the shortcomings of the original formulation.

Undersampling, on the other hand, keeps the minority class examples as is and randomly samples equal numbers of examples from the majority class. That is, some of the majority class examples are excluded from the training data set. The random selection can be with replacement (i.e., bootstrapping) or without replacement.

Another class of methods used to handle imbalanced data problems, called cost-sensitive methods, focuses on misclassification cost. Whereas data sampling methods focus on balancing the data prior to the learning/training process, cost-sensitive methods focus on adjustment to the cost of classification performance during the model building (i.e., training) progress. Specifically, this class of methods assigns different misclassification costs to various classes based on the severity of the imbalance in the data. In practice, these methods aim to either adjust the classification threshold or assign disproportional cost to classes to enhance the focus on the minority class.

Another type of method to handle the imbalanced data problem focuses on formulaic adjustments to a variety of classification algorithms. Although these adjusted classification algorithms all have the

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

same goal (i.e., to mitigate the effect of imbalanced data), the specific methods the algorithms use to achieve this goal differ significantly. The support vector machine and its variants benefit most from this type of mitigation. For example, in a recent research project, Piri et al. (2018) developed a novel synthetic informative minority oversampling (SIMO) algorithm that employs a support vector machine to enhance the learning from imbalanced data sets by creating synthetic examples of the minority class near the support vectors (i.e., the hard-to-classify regions in the multidimensional feature space).

The one-class method has also been used by the machine learning community to address imbalanced data problems. The general idea behind the one-class method is to focus on only one class label at a time. That is, the training data samples are all from a single class (e.g., all positive or all negative samples) in order to learn only the specific characteristics of that class. Because this method focuses on recognizing and predicting only one class, it is often called a recognition-based method; in contrast, discrimination-based methods are commonly used in ordinary classification, where models are developed to discern/differentiate between two or more classes. Because the main idea behind the one-class method is to make the underlying model specific to the characteristics of one class and to recognize everything else as not belonging to that class, it uses three unification/homogenization approaches: (1) density-based characterization, (2) boundary determination, and (3) reconstruction/evolution-based specification. These approaches borrow their conceptual depiction and related functionalities from, and hence can be thought of as extensions of, clustering algorithms such as *k*-means, *k*-medoids, *k*-centers, and self-organizing maps.

Some of the latest methods to handle imbalanced data problems emerged from the field of method ensembles. As opposed to using a single prediction model, this type of model uses a number of either the same type of methods (homogeneous ensembles) or different prediction methods (heterogeneous ensembles) to mitigate the negative

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

effects of the imbalanced data. Variants of both bagging and boosting ensemble methods have been proposed as solutions to imbalanced data problems. In the case of bagging, the random sampling is adjusted to favor the minority class, and in the case of boosting, the weights for the minority class samples are adjusted disproportionally. Active learning—a methodology that involves learning iteratively in a piecewise manner—is also being explored as a way to handle imbalanced data problems.

Despite the fact that there have been numerous efforts to address the imbalanced data problem in the machine learning community, the current state of the art is still limited to heuristic solutions and ad hoc methodologies. There is still an absence of universally accepted theories, methodologies, and best practices. Although most studies claim to have developed data balancing methodologies that improve the prediction accuracy of the minority class, a significant number of studies have also concluded that these methodologies worsen the prediction accuracy of the majority class and the overall classification accuracy. Ongoing research is still seeking to answer questions such as "Can there be a single universal methodology that produces better prediction results?" or "Can there be a methodology that can prescribe the best data balancing algorithm for a given machine learning method and the specifics of the data on hand?"

Explainability of Machine Learning Models for Predictive Analytics

In predictive analytics (and machine learning), there has been a trade-off between model complexity and model performance so that the more complex the machine learning models (e.g., deep neural networks, random forest, gradient boosting), the more accurate the prediction outcomes. As a model gets simpler (i.e., moving toward more parsimony), less predictive performance is observed, but at the same time, the model becomes more interpretable and transparent.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The seminal research paper "Why Should I Trust You" by Ribeiro et al. (2016) appropriately highlights the issue with machine learning models being black boxes. With the new and enhanced interest in machine learning methods—especially since the advent of model ensembles and deep learning—model interpretability has become a fast-growing field of research; today this trend is appropriately called explainable AI and human-interpretable machine learning.

As you have seen in this book, predictive analytics methods and their underlying machine learning algorithms are really good at capturing complex relationships between input and output variables (producing very accurate prediction models), but they are not nearly as good at explaining how they do what they do (i.e., model transparency/model explainability). In order to mitigate this deficiency (commonly called "black-box syndrome"), the machine learning community has proposed several methods, most of which are characterized as sensitivity analysis. Some of these methods are global (providing an explanation based on the average score of all data samples) and some are local (providing a single sample-level explanation). In the context of predictive modeling, sensitivity analysis usually refers to an exclusive experimentation process designed and executed to discover the cause-and-effect relationship between the input and output variables. Some of the variable importance methods are model/algorithm specific (i.e., applied to decision trees, neural networks, random forest), and some are model/algorithm agnostic (i.e., applied to any and every predictive model). Here are the most commonly used variable importance methods employed in machine learning and predictive modeling:

- **Developing and observing a well-trained decision tree model to see the relative discernibility of the input variables.** The closer to the root of the tree a variable is used to split, the greater is its importance or relative contribution to the prediction model.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

- **Developing and observing a rich and large random forest model and assessing the variable split statistics.** If the ratio of a given variable's selection to candidate counts (i.e., the number of times a variable is selected as the level 0 splitter divided by the number of time it was picked randomly as one of the split candidates) is larger, then its importance/relative contribution is also greater. This process can be extended to the top three layers of the trees to generate a weighted average of the split statistics and can be used as a measure of variable importance for random forest models.
- **Sensitivity analysis based on input value perturbation, where the input variables are gradually and systematically changed/perturbed one at a time and the relative change in the output is observed.** The larger the change in the output, the greater the importance of the perturbed variable. This method is often used in trained feed-forward neural network modeling where all of the input variables are numeric and standardized/normalized.
- **Sensitivity analysis based on leave-one-out methodology.** This method can be applied to any type of predictive analytics method; that is, it is predictive model agnostic. Because of its general applicability and ease of implementation, this sensitivity analysis method is used in several commercial and free/open source analytics tools as the default functionality; it is explained further later in this section.
- **Sensitivity analysis based on developing a surrogate model to assess the variable importance of a single record/sample using LIME and SHAP methodologies.** Whereas the methods listed previously are considered global interpreters, LIME and SHAP are called local because they explain the importance of the variables at the sample level (as opposed to the average of all samples). Because of the increased interest in the LIME and SHAP methods, they are explained later in this section.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Sensitivity Analysis Based on Leave-One-Out Methodology

Sensitivity analysis based on leave-one-out methodology relies on the experimental process of systematically removing input variables, one at a time, from the input variable set, developing and testing a model, and observing the impact of the absence of this variable on the predictive performance of the machine learning model. Specifically, the model is first trained and tested (often using *k*-fold cross-validation) with all input variables, and the best predictive accuracy is recorded as the baseline. Then, the same model with exactly the same settings is trained and tested with all but one input variable, and the best prediction accuracy is recorded. This process is repeated once for each variable, each time excluding a different input variable. Then the degradation from the baseline is measured for the omission of each variable, and these measures are used to create a tabular and graphical illustration of the variable importance, like the one shown in Figure 6.12.

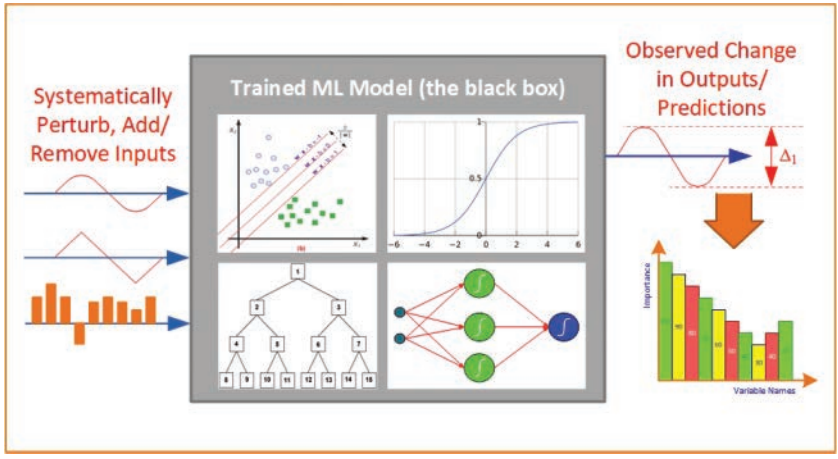


Figure 6.12 A Graphical Depiction of the Sensitivity Analysis Process

This sensitivity analysis method can be used for any predictive analytics and machine learning method. Historically, it has often been used for support vector machines, decision trees, logistic regression, and artificial neural networks. Saltelli (2002) formalized the algebraic

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

representation of this measurement process with the following equation:

$$S_i \frac{V_i}{V(F_t)} = \frac{V(E(F_t \mid X_i))}{V(F_t)}$$

In this equation, the denominator $V(F_t)$ refers to the variance in the output variable. In the numerator, $V(E(F_t|X_i))$, E is the expectation operator to call for an integral over parameter X_i; that is, inclusive of all input variables except X_i, the V, the variance operator applies a further integral over X_i. The variable contribution (i.e., importance), represented as S_i, for the ith variable is calculated as the normalized sensitivity measure. In a later study, Saltelli et al. (2004) proved that this equation is the most probable measure of model sensitivity that is capable of ranking input variables (i.e., predictors) in the order of importance for any combination of interactions, including the non-orthogonal relationships among the input variables.

Using this method of sensitivity analysis may produce slightly different importance measures for different model types for the same data set. It is possible to either select and use the variable importance measured for the most predictive model type while ignoring the ones generated by the other model types or to use some combination (i.e., ensemble) of all importance measures produced by all model types. One way to properly combine the sensitivity analysis results for all prediction model types is to use an information fusion-based methodology—a weighted average in which the weights are determined based on the prediction power of the individual model types. Particularly, it is possible to modify the preceding equation in such a way that the sensitivity measure of an input variable n is obtained based on the information combined (i.e., fused) from m number of prediction model types. The following equation represents this weighted summation function:

$$S_{n(fused)} = \sum_{i=1}^{m} \omega_i S_{in} = \omega_1 S_{1n} + \omega_2 S_{2n} + \ldots + \omega_m S_{mn}$$

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

In this equation ω_i represents the normalized contribution/weight for each prediction model, where the level of contribution/weight of a model is calculated as a function of its relative predictive power. The larger the prediction power (i.e., accuracy), the greater the value of ω.

Local Interpretability with LIME and SHAP

In linear regression models, *beta* coefficients explain the prediction for all the data points; that is, if a variable value increases by one unit, the prediction value increases by beta for every data point. This is commonly called *global interpretability*, and in causal modeling, it is called *average causal analysis* (Nayak, 2019). However, global interpretability does not specifically explain individual differences; for example, the effect of variable values for rejecting the loan application of one applicant could be different from that of another. *Local interpretability* provides an explanation for an individual data point or local subsection of the joint distribution of independent variables. LIME and SHAP are two recent and popular methods that aim to provide local explainability by building surrogate models to the black boxes. They tweak the input slightly (as is done in sensitivity analysis) and test the changes in prediction through a surrogate model representation (see Figure 6.13). Because these surrogate models still treat the machine learning models as black boxes, these methodologies are considered model agnostic.

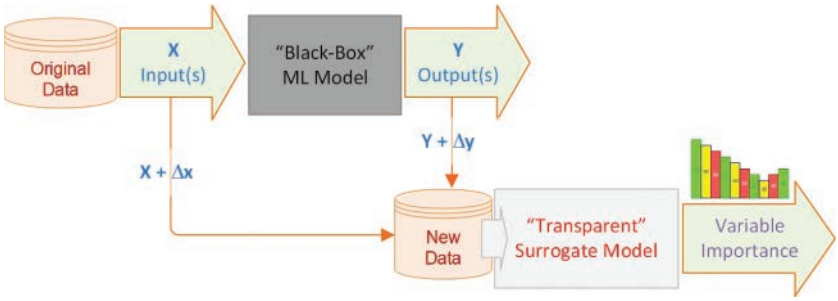


Figure 6.13 From Global Black-Box Model to Local Surrogate Model for Variable Importance

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Local Interpretable Model-Agnostic Explanations (LIME) is a relatively new variable assessment method developed to explain the prediction of any prediction model (i.e., classifier) in a human-interpretable manner by developing/learning a surrogate model locally based on the specifics of the prediction (i.e., the predicted case/sample). The main idea behind LIME is to simulate the relationships between inputs and outputs of a complex machine learning model by developing a surrogate model (which is similar but much simpler and explainable) for explaining the prediction results of a single record (i.e., a specific customer). To do so, the surrogate model utilizes a sample of synthetically generated records that closely resemble the actual predicted record. LIME is a popular technique because it is model agnostic (i.e., it works with any type of prediction model), it can generate human-interpretable outcomes, it provides an explanation at the local (single record) level, it is additive (i.e., the sum of importance values of all the variables for a data point is equal to the final prediction), and it is a computationally very efficient algorithm. For more on LIME and its algorithmic details, see Ribeiro et al. (2016).

Shapley Additive Explanations (SHAP) is another recently proposed technique for model interpretation. It is widely considered the best local model interpretation technique and is superior to LIME in several dimensions. Although SHAP employs a similar concept to LIME, it provides theoretical guarantees based on the game theory concept of Shapley values; it is capable of capturing and representing complex relationships at the local level beyond just linear relationships, and produces more accurate, robust, and reliable interpretation outcomes. SHAP can also be used effectively for variable selection: You can select a subset of variables with higher variable importance and drop other variables if you want to reduce the number of variables. Because of its consistency, the order of variable importance does not change, so less important variables can be ignored. For more details on SHAP, see Lundberg and Lee (2017).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Which Technique Is the Best?

As is the case almost always, there is no magic answer or solution to this question that applies to all problems and data sets. If you are interested in model-agnostic global explainability, perhaps leave-one-out sensitivity analysis could be used. If a model-specific interpretation is desired, then the appropriate sensitivity analysis for the specific model type could be used. If you desire local interpretation, then either LIME of SHAP can be used; of these two, SHAP is considered a better method, but it is computationally more expensive than LIME. If a quick explanation at the single record level is needed for a large and complex problem, perhaps LIME might be the method of choice.

Application Case: To Imprison or Not to Imprison—A Predictive Analytics-Based Decision Support System for Drug Courts

Many businesses, organizations, and government agencies have used analytics to learn from past experiences in order to more effectively and efficiently use their limited resources to achieve their goals and objectives. Despite all the promises of analytics, however, its multidimensional and multidisciplinary nature can sometimes disserve its proper, full-fledged application. This is particularly true for the use of predictive analytics in several social science disciplines, as these domains are traditionally dominated by descriptive analytics (i.e., causal-explanatory statistical modeling) and may not have easy access to the skills required to build predictive analytics models. A review of the extant literature shows that drug court is one such area. While many researchers have studied this initiative, its requirements, and its outcomes from a descriptive analytics perspective, there is currently a dearth of predictive analytics models that can accurately and appropriately predict who

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

would (or would not) graduate from these programs. To fill this gap and to help authorities better manage resources and improve outcomes, this study (Zolbanin, et al., 2020) aims to develop and compare several predictive analytics models (both single models and ensembles) to identify who would graduate from these treatment programs.

Ten years after President Nixon first declared a "war on drugs," President Reagan signed an executive order leading to stricter drug enforcement, stating, "We're taking down the surrender flag that has flown over so many drug efforts; we are running up a battle flag." The reinforcement of the "war on drugs" resulted in an unprecedented 10-fold surge in the number of citizens incarcerated for drug offenses during the following two decades. A skyrocketing number of drug cases inundated court dockets, overloaded the criminal justice system, and overcrowded the prisons. The drug-related caseload, aggravated by the longer processing time of these cases in contrast to those of most other felonies, imposed tremendous costs on state and federal departments of justice.

Court systems started to look for innovative ways to accelerate the inquest of drug-related cases. To determine whether analytics-driven decision support systems are the solution to the problem, the current study sets out to build and compare several predictive models that use a large sample of drug courts data across different locations to predict who is most likely to complete the treatment successfully. They believe this endeavor can reduce the costs to the criminal justice system and local communities in the course of these initiatives.

Methodology

The methodology we used in this effort included a multistep process that aimed to use predictive analytics methods in a social science context. The first step of this process, which focused on

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

understanding the problem domain and the need to conduct this study, is presented in the previous section. For the following steps of the process, we employed a structured and systematic approach to develop and evaluate a set of predictive models using a large and feature-rich real-world data set. These steps included data understanding, data preprocessing, model building, and model evaluation. Our approach also involved multiple iterations of experimentations and numerous modifications to improve individual tasks and to optimize the modeling parameters to achieve the best possible outcomes. Figure 6.14 illustrates our methodology.

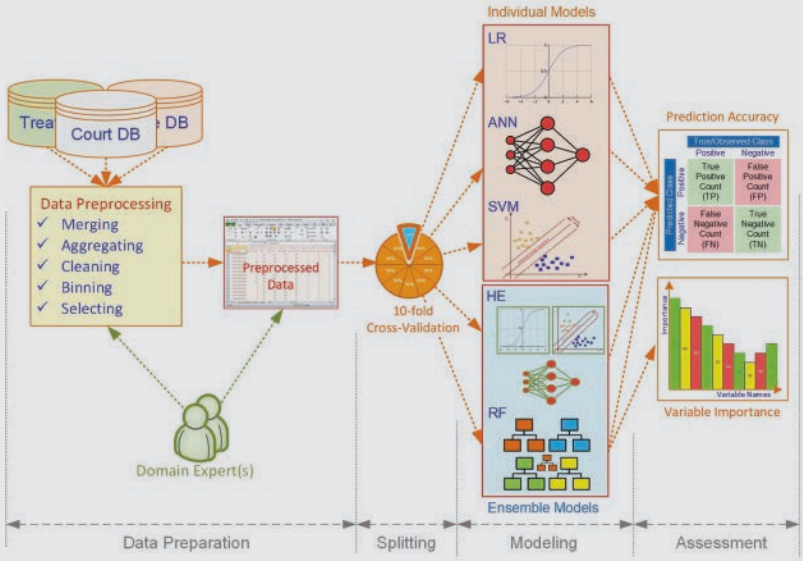


Figure 6.14 Research Methodology Workflow

Results

Table 6.2 provides a summary of the performance of the models, based on accuracy, sensitivity, specificity, and AUC. As the results show, random forest (RF) has the best classification accuracy and the greatest AUC among the models. The heterogeneous ensemble (HE) model closely follows RF, and support vector machines,

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

artificial neural networks, and logistic regression rank third to last, based on their classification performances. RF also has the highest specificity and the second-highest sensitivity. Sensitivity in the context of this study is an indicator of a model's ability in correctly predicting the outcome for successful participants. Specificity, on the other hand, determines how a model performs in predicting the end results for those who do not complete the treatment. Consequently, we can conclude that RF outperforms other models for the drug courts data set used in this study.

Table 6.2 Performance of Predictive Models Using 10-Fold Cross-Validation on the Balanced Data Set

Model Type*			Confusion Matrix		Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC
			G	*T*				
Individual Models	***ANN***	*G*	6831	1072	86.63	86.76	86.49	0.909
		T	1042	6861				
	SVM	*G*	6911	992	88.67	89.63	87.75	0.917
		T	799	7104				
	LR	*G*	6321	1582	85.13	86.16	81.85	0.859
		T	768	7135				
Ensembles	***RF***	*G*	6998	905	91.16	93.44	89.12	0.927
		T	491	7412				
	HE	*G*	6885	1018	90.61	93.66	87.96	0.916
		T	466	7437				

* ANN: artificial neural networks; SVM: support vector machines; DT: decision trees; LR: logistic regression; RF: random forest; HE: heterogeneous ensemble; AUC: area under the curve; G: graduated; T: terminated.

Although the RF model performs better than the other models in general, it falls second to the HE model in the number of false negative predictions. Similarly, the HE model has slightly better performance in terms of true negative predictions. False positive predictions represent the participants who were terminated

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

from the treatment but were mistakenly classified by the models as successful graduates. False negatives pertain to individuals who graduated but who were predicted by the models as dropouts. False positive predictions are synonymous with increased costs and opportunity losses, whereas false negatives carry social impacts. Spending resources on offenders who would recidivate at some point in time during the treatment (and would therefore be terminated from the program) means not giving an opportunity to participate in the treatment to a number of prospective offenders who might potentially be successful. Conspicuously, depriving potentially successful offenders of the treatment is against the initial objective of drug courts in reintegrating nonviolent offenders into their communities.

In summary, traditional descriptive analytics uses statistical inference and significance levels to test and evaluate the explanatory power of hypothesized underlying models or to investigate the association between variables retrospectively. Although it is a legitimate approach to understanding the relationships within the data used to build the model, descriptive analytics falls short in predicting outcomes for prospective observations. In other words, partial explanatory power does not imply predictive power, and predictive analytics is required for building empirical models that predict well. Therefore, based on the findings of this study, application of predictive analytics (rather than use of only descriptive analytics) to predict the outcomes of drug courts is well grounded.

Summary

Predictive analytics is a developing area that involves novel methods and methodologies. The idea behind these developments is to address the shortcomings of the existing techniques in order to create more accurate and more explainable models. This chapter covers

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

several recent advances in predictive analytics: model ensembles, bias–variance trade-off, imbalanced data problems, and explainable AI/machine learning. While the first three of these advances focus on building more accurate and unbiased predictive models, explainable AI/machine learning focuses on making the black-box models more transparent so that they can be used in domains where explainability is one of the required characteristics in a prediction model.

References

Abbott, D. (2014). *Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst.* John Wiley & Sons.

Breiman, L. (1996). "Bagging Predictors," *Machine Learning,* 24(2): 123–140.

Delen, D., & Sharda, R. (2010). "Predicting the Financial Success of Hollywood Movies Using an Information Fusion Approach." *Industrial Engineering Journal*, 21(1), 30-37.

Freund, Y., & Schapire, R. E. (1996). "Experiments with a New Boosting Algorithm," *Machine Learning: Proceedings of the Thirteenth International Conference*, pp. 148–156.

Friedman, J. H. (2001). "Greedy Function Approximation: A Gradient Boosting Machine," *Annals of Statistics*, pp. 1189–1232.

He, H., & Ma, Y. (Eds.). (2013). *Imbalanced Learning: Foundations, Algorithms, and Applications*. John Wiley & Sons.

Lundberg, S. M., & Lee, S. I. (2017). "A Unified Approach to Interpreting Model Predictions," *Advances in Neural Information Processing Systems*, pp. 4765–4774.

Nayak, A. (2019). *Idea Behind LIME and SHAP.* https://towardsdatascience.com/idea-behind-lime-and-shap-b603d35d34eb (accessed July 2020).

Piri, S., Delen, D., & Liu, T. (2018). "A Synthetic Informative Minority Over-Sampling (SIMO) Algorithm Leveraging Support Vector Machine to Enhance Learning from Imbalanced Datasets," *Decision Support Systems*, 106: 15–29.

Piri, S., Delen, D., Liu, T., & Zolbanin, H. M. (2017). "A Data Analytics Approach to Building a Clinical Decision Support System for Diabetic Retinopathy: Developing and Deploying a Model Ensemble," *Decision Support Systems*, 101: 12–27.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). "Why Should I Trust You? Explaining the Predictions of Any Classifier," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 1135–1144.

Saltelli, A. (2002). "Sensitivity Analysis for Importance Assessment." *Risk analysis*, 22(3), 579-590.

Saltelli, A., Tarantola, S., Campolongo, F., & Ratto, M. (2004). *Sensitivity Analysis in practice: A Guide to Assessing Scientific Models* (Vol. 1). New York: Wiley.

Sarkar, D. (2018). *The Importance of Human Interpretable Machine Learning*. https://towardsdatascience.com/human-interpretable-machine-learning-part-1-the-need-and-importance-of-model-interpretation-2ed758f5f476 (accessed July 2020).

Seni, G., & Elder, J. F. (2010). "Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions," *Synthesis Lectures on Data Mining and Knowledge Discovery,* 2(1): 1–126.

Surowiecki, J. (2006). *The Wisdom of Crowds.* Penguin Random House Publishing.

Thammasiri, D., Delen, D., Meesad, P., & Kasap, N. (2014). "A Critical Assessment of Imbalanced Class Distribution Problem: The Case of Predicting Freshmen Student Attrition," *Expert Systems with Applications,* 41(2): 321–330.

Vorhies, W. (2016). *Want to Win Competitions? Pay Attention to Your Ensembles.* Data Science Central Web Portal. www.datasciencecentral.com/profiles/blogs/want-to-win-at-kaggle-pay-attention-to-your-ensembles (accessed September 2020).

Wolpert, D. (1992). "Stacked Generalization," *Neural Networks,* 5(2): 241–260.

Zolbanin, H. M., Delen, D., Crosby, D., & Wright, D. (2020). "A Predictive Analytics-Based Decision Support System for Drug Courts." *Information Systems Frontiers*, 1–20.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

7

Text Analytics, Topic Modeling, and Sentiment Analysis

The information age that we are living in is characterized by rapid growth in the amount of data and information collected, stored, and made available in electronic format. The vast majority of business data is stored in text documents that are virtually unstructured. According to a study by Merrill Lynch and Gartner, 85% of all corporate data is captured and stored in some sort of unstructured form (McKnight, 2005). The same study also stated that this unstructured data is doubling in size every 18 months. Because knowledge is power in today's business world, and knowledge is derived from data and information, businesses that effectively and efficiently tap into their text data sources will have the necessary knowledge to make better decisions, leading to a competitive advantage over businesses that lag behind. This is where the need for text analytics and text mining fits into the big picture of today's businesses.

Even though the overarching goal for both text analytics and text mining is to turn unstructured textual data into actionable information through the application of natural language processing (NLP) and analytics, the definitions of these terms are somewhat different—at least to some experts in the field. According to many, *text analytics* is a broader concept that includes information retrieval (e.g., searching and identifying relevant documents for a given set of key terms) as well as information extraction, data mining, and web mining, whereas

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

text mining is primarily focused on discovering new and useful knowledge from textual data sources.

Figure 7.1 illustrates the relationships between text analytics and text mining, along with other related application areas. The bottom of the figure lists the main disciplines that play critical roles in the development of these increasingly popular application areas and provide a foundation for them.

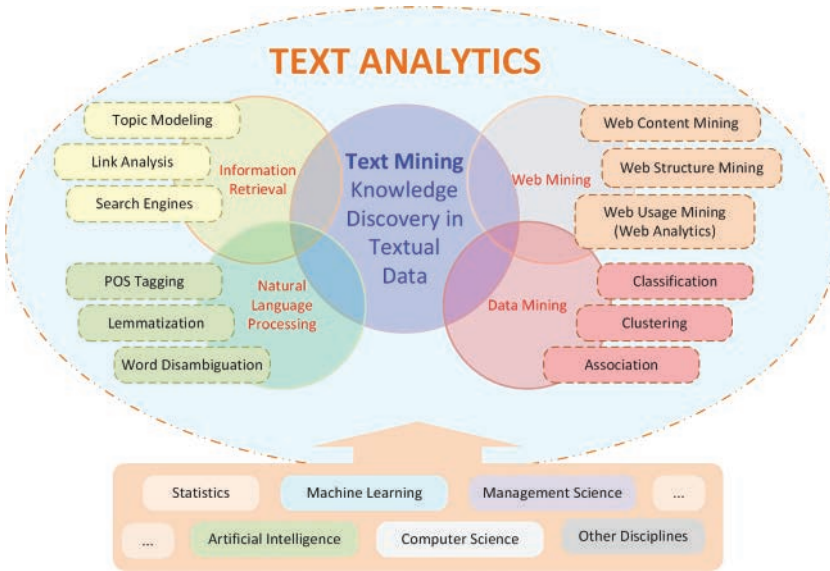


Figure 7.1 Text Mining and Enabling Disciplines

Text analytics is a newer term than text mining. With the recent emphasis on analytics, many related technical application areas (e.g., consumer analytics, completive analytics, visual analytics, social analytics) have jumped on the analytics bandwagon. While the term *text analytics* is more commonly used in the business application context, the term *text mining* is frequently used in academic research circles. Even though these terms may be defined somewhat differently at times, they are usually used interchangeably.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Text mining (also known as text data mining or knowledge discovery in textual databases) is the semiautomated process of extracting patterns (i.e., useful information and knowledge) from large amounts of unstructured data. Remember that data mining is the process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data stored in structured databases, where the data are organized in records structured by categorical, ordinal, or continuous variables. Text mining is the same as data mining in that it has the same purpose and uses the same processes, but with text mining the input to the process is a collection of unstructured (or less structured) data files, such as Word documents, PDF files, text excerpts, XML files, tweets, blog posts, customer reviews, and so on. In essence, text mining can be thought of as a process that starts with imposing structure on the text-based data sources followed by extracting relevant information and knowledge from this structured text-based data by using data mining techniques and tools.

The benefits of text mining are obvious in areas where very large amounts of textual data are being generated, such as social media (e.g., tweets, blogs, wall posts), law (e.g., court orders), academic research (e.g., research articles), marketing (e.g., online reviews), finance (e.g., quarterly reports and discussion boards), medicine (e.g., discharge summaries), biology (e.g., molecular interactions), technology (e.g., patent files), and marketing (e.g., customer comments). For example, free-form text-based interactions with customers in the form of complaints (or praise) and warranty claims can be used to objectively identify product and service characteristics that are deemed to be less than perfect and can be used as input to improve product development and service allocations. Information can be used similarly with market outreach programs and focus groups that generate large amounts of data. By not restricting product or service feedback to a codified form, customers can present, in their own words, what they think about a company's products and services.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Another area where the automated processing of unstructured text has had a lot of impact is in email and other electronic communications. Text mining can be used to classify and filter junk email, and it can also be used to automatically prioritize email based on importance level as well as generate automatic responses (Weng & Liu, 2004). The following are some of the most popular application areas of text mining:

- **Information extraction.** Text mining can identify key phrases and relationships in text by looking for predefined objects and sequences in text through pattern matching. Perhaps the most commonly used form of information extraction is entity extraction. Named entity extraction includes named entity recognition (i.e., recognition of known entity names—for people and organizations, place names, temporal expressions, and certain types of numeric expressions, using existing knowledge of the domain), co-reference resolution (i.e., detection of co-reference and anaphoric links between text entities), and relationship extraction (i.e., identification of relationships between entities).
- **Topic detection.** Text mining can be used to identify hidden/latent topics within a large collection of textual documents through some type of probabilistic machine learning algorithms.
- **Topic tracking.** Based on a user profile and documents that a user views, text mining can predict other documents that might be of interest to the user.
- **Summarization.** Text mining can be used to summarize a document to save the reader time.
- **Categorization.** Text mining can be used to identify the main themes of a document and then place the document into a predefined set of categories, based on those themes.
- **Sentiment Analysis.** Automatic identification of opinions and sentiments (i.e., negative, positive, or neutral) hidden in textual

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

content created by humans over the Internet (e.g., product reviews, blog posts, social media interactions).

- **Clustering.** Text mining can be used to group similar documents without having a predefined set of categories.
- **Concept linking.** Text mining can be used to connect related documents by identifying their shared concepts and, by doing so, help users find information that they perhaps would not find by using traditional search methods.
- **Question answering.** Text mining can be used to find the best answer to a given question through knowledge-driven pattern matching.

Text mining has its own language, with many technical terms and acronyms. Following is a list of some of the terms and concepts that are commonly used in the context of text mining:

- **Unstructured data (versus structured data).** Structured data has a predetermined format. It is usually organized into records with simple data values (i.e., categorical, ordinal, and continuous variables) and stored in databases. In contrast, unstructured data does not have a predetermined format and is stored in the form of textual documents. In essence, the structured data is for the computers to process, while the unstructured data is for humans to process and understand.
- **Corpus.** In linguistics, a corpus (plural *corpora*) is a large and structured set of texts (now usually stored and processed electronically), prepared for the purpose of conducting knowledge discovery.
- **Terms.** A term is a single-word or multiword phrase extracted directly from the corpus of a specific domain through natural language processing (NLP).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

- **Concepts.** Concepts are features generated from a collection of documents by means of manual, statistical, rule-based, or hybrid categorization methodologies. Concepts result from higher-level abstraction than do terms.
- **Stemming.** Stemming is the process of reducing inflected words to their stem (or base or root) form. For instance, *stemmer*, *stemming*, and *stemmed* are all based on the root *stem*.
- **Lemmatization.** Similar to stemming, lemmatization is also used to convert inflected words to their stem/root forms. While stemming does this syntactically, lemmatization does it semantically using a term dictionary.
- **Stop words.** Stop words (or noise words) are words that are filtered out prior to or after processing of natural language data (i.e., text). Even though there is no universally accepted list of stop words, most natural language processing tools use a list that includes articles (e.g., a, an, the), auxiliary verbs (e.g., is, are, was, were), and context-specific words that are deemed not to have differentiating value.
- **Synonyms and polysemes.** Synonyms are syntactically different words (i.e., spelled differently) with identical or at least similar meanings (e.g., movie, film, and motion picture). In contrast, polysemes, also called homonyms, are syntactically identical words (i.e., spelled exactly the same) but have different meanings (e.g., bow can mean "to bend forward," "the front of a ship," "a weapon that shoots arrows," or "a kind of tied ribbon").
- **Tokenizing.** A token is a categorized block of text in a sentence. The block of text corresponding to the token is categorized according to the function it performs. This assignment of meaning to blocks of text is known as tokenizing. A token can look like anything; it just needs to be a useful part of the structured text.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

- **Term dictionary.** A term dictionary is a collection of terms specific to a narrow field that can be used to restrict the extracted terms within a corpus.
- **Word frequency.** The number of times a word is found in a specific document is the word frequency.
- **Part-of-speech tagging.** This is the process of marking up the words in a text that correspond to a particular part of speech (e.g., nouns, verbs, adjectives, adverbs), based on each word's definition and the context in which it is used.
- **Morphology.** Morphology is a branch of the field of linguistics and a part of natural language processing that studies the internal structure of words (patterns of word formation within a language or across languages).
- **Term-by-document matrix (occurrence matrix).** This type of matrix is a common representation schema for the frequency-based relationship between terms and documents in tabular format, where terms are listed in rows, documents are listed in columns, and the frequency between the terms and documents is listed in cells as integer values.
- **Singular-value decomposition (latent semantic indexing).** This dimensionality reduction method is used to transform a term-by-document matrix to a manageable size by generating an intermediate representation of the frequencies using a matrix manipulation method similar to principal-component analysis.

Natural Language Processing

Some of the early text mining applications used a simplified representation called bag-of-words when introducing structure to a collection of text-based documents in order to classify them into two or

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

more predetermined classes or cluster them into natural groupings. In the bag-of-words model, text—such as a sentence, paragraph, or complete document—is represented as a collection of words, disregarding the grammar or the order in which the words appear. The bag-of-words model is still used in some simple document classification tools. For instance, in spam filtering, an email message can be modeled as an unordered collection of words (a bag of words) that is compared against two different predetermined bags. One bag is filled with words found in spam messages, and the other is filled with words found in legitimate emails. Although some of the words are likely to be found in both bags, the "spam" bag will contain spam-related words such as *stock market*, *Viagra*, and *buy* much more frequently than the legitimate bag, which will contain more words related to the user's friends or workplace. The level of match between a specific email's bag of words and the two bags containing the descriptors determines the characterization of the email as either spam or legitimate.

Naturally, we humans do not use words without some order or structure. We use words in sentences, which have semantic as well as syntactic structure. Thus, automated techniques (e.g., text mining) need to look for ways to go beyond the bag-of-words interpretation and incorporate more and more semantic structure into their operations. The current trend in text mining is toward including many of the advanced features that can be obtained using natural language processing.

It has been shown that the bag-of-words method may not produce good enough information content for text mining tasks (e.g., classification, clustering, association). A good example of this can be found in evidence-based medicine. A critical component of evidence-based medicine is incorporating the best available research findings into the clinical decision-making process, which involves appraisal of the information collected from the printed media for validity and relevance. Several researchers from the University of Maryland developed

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

evidence assessment models using a bag-of-words method (Lin & Demner-Fushman, 2005). They employed popular machine learning methods along with more than half a million research articles collected from MEDLINE (Medical Literature Analysis and Retrieval System). In their models, they represented each abstract as a bag of words, where each stemmed term represented a feature. Despite using popular classification methods with proven experimental design methodologies, their prediction results were not much better than simple guessing, which may indicate that the bag-of-words method is not generating a good enough representation of the research articles in this domain; hence, more advanced techniques such as natural language processing are needed.

Natural language processing (NLP) is an important component of text mining and is a subfield of artificial intelligence and computational linguistics. It studies the problem of understanding natural human language, with the view of converting depictions of human language (e.g., textual documents) into more formal representations (in the form of numeric and symbolic data) that are easier for computer programs to manipulate. The goal of NLP is to move beyond syntax-driven text manipulation (which is often called "word counting") to a true understanding and processing of natural language that considers grammatical and semantic constraints as well as the context.

The definition and scope of the word *understanding* is one of the major discussion topics in NLP. Considering that the natural human language is vague and that a true understanding of meaning requires extensive knowledge of a topic (beyond what is in the words, sentences, and paragraphs), will computers ever be able to understand natural language the same way and with the same accuracy that humans do? Probably not! NLP has come a long way from the days of simple word counting, but it has an even longer way to go to really

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

understand natural human language. The following are just a few of the challenges commonly associated with the implementation of NLP:

- **Part-of-speech tagging.** It is difficult to mark up terms in a text as corresponding to a particular part of speech (e.g., nouns, verbs, adjectives, adverbs) because the part of speech depends not only on the definition of the term but also on the context in which the term is used.
- **Text segmentation.** Some written languages, such as Chinese, Japanese, and Thai, do not have single-word boundaries. In these instances, the text-parsing task requires the identification of word boundaries, which is often difficult. Similar challenges in speech segmentation emerge when analyzing spoken language because sounds representing successive letters and words blend into each other.
- **Word sense disambiguation.** Many words have more than one meaning. Selecting the meaning that makes the most sense can only be accomplished by taking into account the context in which the word is used.
- **Syntactic ambiguity.** The grammar for natural languages is ambiguous; that is, multiple possible sentence structures often need to be considered. Choosing the most appropriate structure usually requires a fusion of semantic and contextual information.
- **Imperfect or irregular input.** Foreign or regional accents and vocal impediments in speech and typographical or grammatical errors in texts make the processing of language a difficult task.
- **Speech acts.** A sentence can often be considered a request for action. The sentence structure alone may not contain enough information to define the action. For example, "Can you pass the class?" requests a simple yes/no answer, whereas "Can you pass the salt?" is a request for a physical action to be performed.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

It is a longstanding dream of the artificial intelligence community to come up with algorithms that are capable of automatically reading and obtaining knowledge from text. By applying a learning algorithm to parsed text, researchers from Stanford University's NLP lab have developed methods that can automatically identify concepts and relationships between those concepts in text. By applying a unique procedure to large amounts of text, their algorithms automatically acquire hundreds of thousands of items of world knowledge and use them to produce significantly enhanced repositories for WordNet. WordNet is a laboriously hand-coded database of English words, their definitions, sets of synonyms, and various semantic relationships between synonym sets. It is a major resource for NLP applications, but it has proven to be very expensive to build and maintain manually. By automatically inducing knowledge into WordNet, the potential exists to make WordNet an even greater and more comprehensive resource for NLP at a fraction of the cost.

One prominent area where the benefits of NLP and WordNet are already being harvested is in customer relationship management (CRM). Broadly speaking, the goal of CRM is to maximize customer value by better understanding and effectively responding to actual and perceived customer needs. An important area of CRM, where NLP is making a significant impact, is sentiment analysis. Sentiment analysis is a technique used to detect favorable and unfavorable opinions about specific products and services, using large numbers of textual data sources, such as customer feedback from web postings. (Sentiment analysis is covered in detail later in this chapter.)

NLP has successfully been applied to a variety of domains for a variety of tasks via computer programs to automatically process natural human language that previously could be done only by humans. Following are among the most popular of these tasks:

- **Question answering.** NLP can be used to automatically answer a question posed in natural language—that is, to produce

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

a human-language answer to a human-language question. To find the answer to a question, the computer program may use either a prestructured database or a collection of natural language documents (e.g., a text corpus such as the World Wide Web).

- **Automatic summarization.** A computer program can be used to create a shortened version of a textual document that contains the most important points of the original document.
- **Natural language generation.** Systems can convert information from computer databases into readable human language.
- **Natural language understanding.** Systems can convert samples of human language into more formal representations that are easier for computer programs to manipulate.
- **Machine translation.** Systems can automatically translate one human language to another.
- **Foreign language reading.** A computer program can assist a nonnative language speaker in reading a foreign language with correct pronunciation and accents on different parts of the words.
- **Foreign language writing.** A computer program can assist a nonnative language user in writing in a foreign language.
- **Speech recognition.** Systems can convert spoken words to machine-readable input. Given a sound clip of a person speaking, such a system produces a text dictation.
- **Text-to-speech.** A computer program can automatically convert normal language text into human speech. This is also called speech synthesis.
- **Text proofing.** A computer program can read a proof copy of a text in order to detect and correct any errors.
- **Optical character recognition.** A system can automatically translate images of handwritten, type-written, or printed text (usually captured by a scanner) into machine-editable textual documents.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The success and popularity of text mining depends greatly on advancements in NLP in both generation as well as understanding of human languages. NLP enables the extraction of features from unstructured text so that a wide variety of data mining techniques can be used to extract knowledge (novel and useful patterns and relationships) from it. In that sense, simply put, text mining is a combination of NLP and data mining.

Text Mining Applications

As the amount of unstructured data collected by organizations increases, so do the value proposition and popularity of text mining tools. Many organizations are now realizing the importance of extracting knowledge from their document-based data repositories through the use of text mining tools. The following sections describe a small subset of the exemplary application categories of text mining.

Marketing Applications

Text mining can be used to increase cross-selling and upselling by analyzing the unstructured data generated by call centers. Text mining algorithms can be used to analyze text generated by call center notes as well as transcriptions of voice conversations with customers to extract novel, actionable information about customers' perceptions about a company's products and services. In addition, blogs, user reviews of products at independent websites, and discussion board postings are gold mines of customer sentiments. This rich collection of information, properly analyzed, can be used to increase satisfaction and the overall lifetime value of a customer.

Text mining has become invaluable for customer relationship management. Companies can use text mining to analyze rich sets of unstructured text data, combined with relevant structured data extracted from organizational databases, to predict customer

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

perceptions and subsequent purchasing behavior. For instance, text mining can be used to meaningfully improve the ability of a mathematical model to predict customer churn (i.e., customer attrition) so that customers identified as most likely to leave a company could be accurately identified for retention tactics.

Treating products as sets of attribute–value pairs rather than as atomic entities can potentially boost the effectiveness of many business applications, including demand forecasting, assortment optimization, product recommendations, assortment comparison across retailers and manufacturers, and product supplier selection. Ghani et al. (2006) used text mining to develop a system capable of inferring implicit and explicit attributes of products to enhance retailers' ability to analyze product databases. The proposed system allows a business to represent its products in terms of attributes and attribute values without much manual effort. The system learns these attributes by applying supervised and semi-supervised learning techniques to product descriptions found on retailers' websites.

Security Applications

One of the largest and most prominent text mining applications in the security domain is the highly classified ECHELON surveillance system. As rumor has it, ECHELON is capable of identifying the content of telephone calls, faxes, emails, and other types of data, intercepting information sent via satellites, public switched telephone networks, and microwave links.

In 2007, EUROPOL developed an integrated system capable of accessing, storing, and analyzing vast amounts of structured and unstructured data sources in order to track transnational organized crime. Called the Overall Analysis System for Intelligence Support (OASIS), this system aims to integrate the most advanced data and text mining technologies available in today's market. The system has

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

enabled EUROPOL to make significant progress in supporting its law enforcement objectives at the international level (EUROPOL, 2007).

The U.S. Federal Bureau of Investigation (FBI) and the Central Intelligence Agency (CIA), under the direction of the Department of Homeland Security, are jointly developing a supercomputer data and text mining system. The system is expected to create a gigantic data warehouse along with a variety of data and text mining modules to meet the knowledge-discovery needs of federal, state, and local law enforcement agencies. Prior to this project, the FBI and CIA each had its own separate databases, with little or no interconnection.

Another security-related application of text mining is in the area of deception detection. Applying text mining to a large set of real-world criminal (person-of-interest) statements, Fuller et al. (2008) developed prediction models to differentiate deceptive statements from truthful ones. Using a rich set of cues extracted from the textual statements, the model predicted the holdout samples with 70% accuracy, which is believed to be a significant success, considering that the cues are extracted only from textual statements (with no verbal or visual cues). Furthermore, compared to other deception-detection techniques, such as the polygraph, this method is nonintrusive and widely applicable to not only textual data but also (potentially) to transcriptions of voice recordings. A more detailed description of text-based deception detection is provided at the end of this chapter.

Biomedical Applications

Text mining holds great potential for the medical field in general and biomedicine in particular for several reasons. First, the published literature and publication outlets (especially with the advent of open source journals) in the field are expanding exponentially. Second, compared to most other fields, the medical literature is more standardized and orderly, making it a more "minable" information source.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Finally, the terminology used in this literature is relatively constant, having a fairly standardized ontology. A few exemplary studies have successfully used text mining techniques in extracting novel patterns from biomedical literature.

Experimental techniques such as DNA microarray analysis, serial analysis of gene expression (SAGE), and mass spectrometry proteomics, among others, are generating large amounts of data related to genes and proteins. As in any other experimental approach, it is necessary to analyze this vast amount of data in the context of previously known information about the biological entities under study. The literature is a particularly valuable source of information for experiment validation and interpretation. Therefore, the development of automated text mining tools to assist in such interpretation is one of the main challenges in current bioinformatics research.

Knowing the location of a protein within a cell can help elucidate its role in biological processes and determine its potential as a drug target. Numerous location-prediction systems are described in the literature; some focus on specific organisms, whereas others attempt to analyze a wide range of organisms. Shatkay et al. (2007) proposed a comprehensive system that uses several types of sequence- and text-based features to predict the locations of proteins. The main novelty of their system lies in the way in which it selects its text sources and features and integrates them with sequence-based features. Shatkay et al. (2007) tested the system on previously used data sets and on new data sets devised specifically to test the predictive power of the system. The results showed that their system consistently beat previously reported results.

Chun et al. (2006) described a system that extracts disease–gene relationships from literature accessed via MEDLINE. They constructed a dictionary for disease and gene names from six public databases and extracted relationship candidates by using dictionary matching. Because dictionary matching produces a large number of false positives, Chun et al. developed a method of machine

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

learning–based named entity recognition (NER) to filter out false recognition of disease/gene names. They found that the success of relation extraction is heavily dependent on the performance of NER filtering and that the filtering improved the precision of relation extraction by 26.7%, at the cost of a small reduction in recall.

Figure 7.2 shows a simplified illustration of a multilevel text analysis process for discovering gene–protein relationships (or protein–protein interactions) in the biomedical literature (Nakov et al., 2005). This example uses a simple sentence from biomedical text. As shown in the figure, first (at the bottom three levels), the text is tokenized using part-of-speech tagging and shallow parsing. The tokenized terms (words) are then matched (and interpreted) against the hierarchical representation of the domain ontology to derive the gene–protein relationship. Application of this method (and/or some variation of it) to the biomedical literature offers great potential for decoding the complexities in the Human Genome Project.

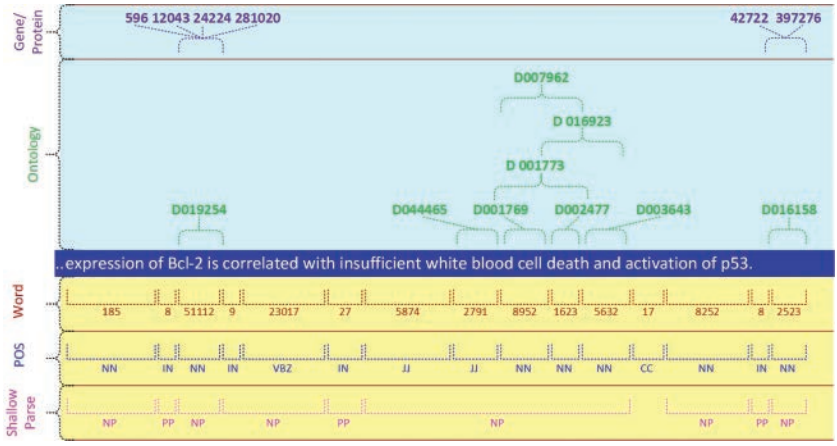


Figure 7.2 Multilevel Analysis of Text for Gene/Protein Interaction Identification

The issue of text mining is of great importance to publishers that hold large databases of information requiring indexing for better retrieval. This is particularly true in scientific disciplines, in

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

which highly specific information is often contained in written text. Initiatives have been launched, such as *Nature*'s proposal for an Open Text Mining Interface (OTMI) and the National Institutes of Health's Journal Publishing Document Type Definition (DTD), that would provide semantic cues to machines to answer specific queries contained in text without removing publisher barriers to public access.

Academic institutions have also launched text mining initiatives. For example, the National Centre for Text Mining, a collaborative effort between the Universities of Manchester and Liverpool in the United Kingdom, provides customized tools, research facilities, and advice on text mining to the academic community. With an initial focus on text mining in the biological and biomedical sciences, research has since expanded into the social sciences. In the United States, the School of Information at the University of California, Berkeley, is developing a program called BioText to assist bioscience researchers in text mining and analysis.

The Text Mining Process

In order to be successful, text mining studies should follow a sound methodology based on best practices. Remember from Chapter 3, "Standardized Processes for Predictive Analytics," that CRISP-DM is the industry standard for data mining projects; likewise, a standardized process model is needed for text mining. Even though most parts of CRISP-DM are also applicable to text mining projects, a specific process model for text mining would include much more elaborate data preprocessing activities. Figure 7.3 provides a high-level context diagram of a typical text mining process (Delen & Crossland, 2008). This context diagram presents the scope of the process, emphasizing its interfaces with the larger environment. In essence, it draws boundaries around the specific process to explicitly identify what is included in (and excluded from) the text mining process.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

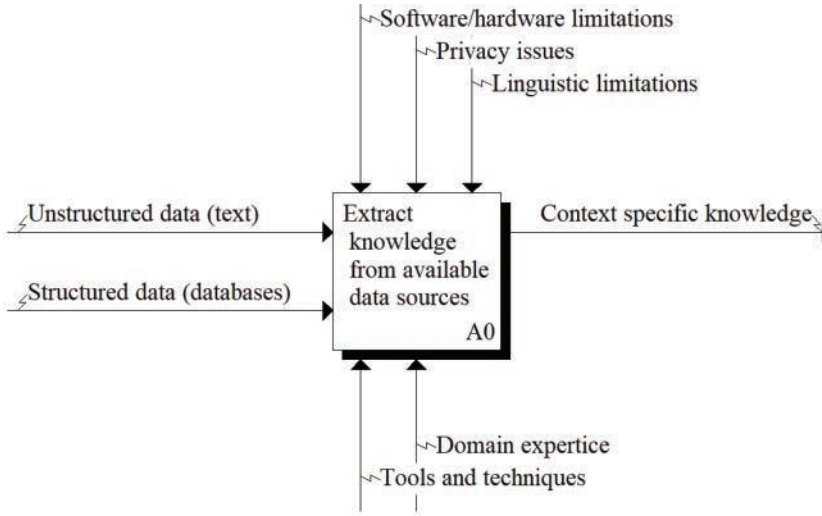


Figure 7.3 Context Diagram for the Text Mining Process

As the context diagram indicates, the input (the inward connection to the left edge of the box) into the text-based knowledge discovery process is the unstructured as well as structured data collected, stored, and made available to the process. The output (the outward extension from the right edge of the box) of the process is the context-specific knowledge that can be used for decision making. The controls, also called the constraints (the inward connection to the top edge of the box), of the process include software and hardware limitations, privacy issues, and difficulties related to processing the text that is presented in the form of natural language. The mechanisms (the inward connection to the bottom edge of the box) of the process include proper techniques, software tools, and domain expertise. The primary purpose of text mining (in the context of knowledge discovery) is to process unstructured (textual) data (along with structured data, if relevant to the problem being addressed and available) to extract meaningful and actionable patterns for better decision making.

At a very high level, the text mining process can be broken down into three consecutive tasks, each of which has specific inputs to generate certain outputs (see Figure 7.4). If, for some reason, the output of a task is not what was expected, a backward redirection to the previous task execution is necessary.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

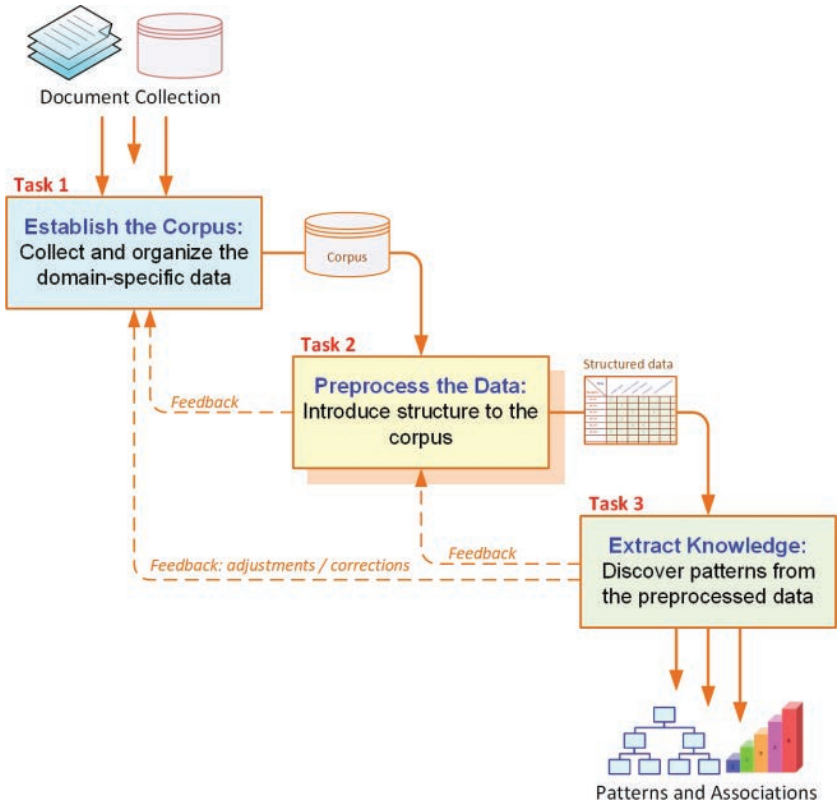


Figure 7.4 The Three-Task Text Mining Process

Task 1: Establish the Corpus

The main purpose of the first task in the text mining process is to collect all the documents related to the context (i.e., domain of interest) being studied. This collection may include textual documents, XML files, emails, web pages, and short notes. In addition to the readily available textual data, voice recordings may also be transcribed using speech-recognition algorithms and made a part of the text collection.

Once collected, the text documents are transformed and organized in a manner such that they are all in the same representational form (e.g., ASCII text files) for computer processing. The organization

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

of the documents can be as simple as a collection of digitized text excerpts stored in a file folder, or it can be a list of links to a collection of web pages in a specific domain. Many commercially available text mining software tools can accept these as input and convert them into a flat file for processing. Alternatively, the flat file can be prepared outside the text mining software and then presented as input to the text mining application.

Task 2: Preprocess the Data: Create the Term–Document Matrix

In task 2 of the text mining process, the digitized and organized documents (the corpus) are used to create the term–document matrix (TDM). In the TDM, rows represent the documents, and columns represent the terms. The relationships between the terms and documents are characterized by indices. (An index is a relational measure that can be as simple as the number of occurrences of the term in respective documents.) Figure 7.5 shows a typical example of a TDM.

Terms / Documents	market share	resource utilization	project schedule	acquisition	material requirement	...	
Article 1	1			1			
Article 2		1					
Article 3			3		1		
Article 4		1					
Article 5			2	1			
Article 6	1			1			
...							

Figure 7.5 An Example of a Term–Document Matrix

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The goal is to convert the list of organized documents (the corpus) into a TDM where the cells are filled with the most appropriate indices. The assumption is that the essence of a document can be represented with a list of the frequency of the terms used in that document. However, not all terms are important when characterizing documents. Some terms, such as articles, auxiliary verbs, and terms used in almost all the documents in the corpus, have no differentiating power and therefore should be excluded from the indexing process. This list of terms, commonly called *stop terms* or *stop words*, is specific to the domain of study and should be identified by the domain experts. On the other hand, the experts might choose a set of predetermined terms under which the documents are to be indexed (called *include terms* or a *dictionary*). In addition, synonyms (pairs of terms that are to be treated the same) and specific phrases (e.g., "Eiffel Tower") can also be provided so that the index entries are more accurate.

Another filtration that should take place to accurately create the indices is stemming, which refers to the reduction of words to their roots so that, for example, different grammatical forms or conjugations of a verb are identified and indexed as the same word. For example, stemming would ensure that *modeling* and *modeled* will be recognized as the word *model*.

The first generation of a TDM includes all the unique terms identified in the corpus (as its columns), excluding the ones in the stop term list; all the documents (as its rows); and the occurrence count of each term for each document (as its cell values). If, as is commonly the case, the corpus includes a rather large number of documents, then there is a very good chance that the TDM will have a very large number of terms. Processing such a large matrix might be time-consuming and, more importantly, might lead to extraction of inaccurate patterns. At this point, it is important to decide the following:

- What is the best way to represent the indices?
- How can we reduce the dimensionality of this matrix to a manageable size?

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

What Is the Best Way to Represent the Indices?

Once the input documents are indexed and the initial word frequencies (by document) are computed, a number of additional transformations can be performed to summarize and aggregate the extracted information. The raw term frequencies generally reflect on how salient or important a word is in each document. Specifically, words that occur with greater frequency in a document are better descriptors of the contents of that document. However, it is not reasonable to assume that the word counts themselves are proportional to their importance as descriptors of the documents. For example, if a word occurs one time in document A but three times in document B, then it is not necessarily reasonable to conclude that the word is three times as important a descriptor of document B as of document A. In order to have a more consistent TDM for further analysis, these raw indices need to be normalized. As opposed to showing the actual frequency counts, the numeric representation between terms and documents can be normalized using a number of alternative methods. The following are a few of the most commonly used normalization methods (StatSoft, 2014):

- **Log frequencies.** The raw frequencies can be transformed using the log function. This transformation "dampens" the raw frequencies and how they affect the results of subsequent analysis.
- **Binary frequencies.** An even simpler transformation than log frequencies can be used to enumerate whether a term is used in document binary frequencies. The resulting TDM matrix will contain only 1s and 0s to indicate the presence or absence of the respective words. Again, this transformation dampens the effect of the raw frequency counts on subsequent computations and analyses.
- **Inverse document frequencies.** Another issue to consider related to the indices used in further analyses is the relative

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

document frequencies (df) of different terms. For example, a term such as *guess* may occur frequently in all documents, whereas another term, such as *software*, may appear only a few times. The reason is that one might make *guesses* in various contexts, regardless of the specific topic, whereas *software* is a more semantically focused term that is only likely to occur in documents that deal with computer software. A common and very useful transformation that reflects both the specificity of words (i.e., document frequencies) and the overall frequencies of their occurrences (i.e., term frequencies) is the so-called inverse document frequency (Manning et al., 2008).

How Can We Reduce the Dimensionality of the TDM Matrix?

Because a TDM is often very large and rather sparse (with most of the cells filled with zeros), it is important to find a way to reduce the dimensionality of this matrix to a manageable size. Several options are available for managing matrix size:

- Have a domain expert go through the list of terms and eliminate those that do not make much sense for the context of the study. This is a manual, labor-intensive process.
- Eliminate terms with very few occurrences in very few documents.
- Transform the matrix by using singular value decomposition.

Singular value decomposition (*SVD*), which is closely related to principal-component analysis, reduces the overall dimensionality of the input matrix (i.e., the number of input documents by the number of extracted terms) to a lower dimensional space, where each consecutive dimension represents the largest degree of variability (between words and documents) possible (Manning et al., 2008). Ideally, an analyst might identify the two or three most salient dimensions that account for most of the variability (i.e., differences) between the

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

words and documents, thus identifying the latent semantic space that organizes the words and documents in the analysis. Once such dimensions are identified, the underlying meaning of what is contained (discussed or described) in the documents has been extracted.

Task 3: Extract the Knowledge

Using a well-structured TDM, potentially augmented with other structured data elements, novel patterns are extracted in the context of the specific problem being addressed. The main categories of knowledge extraction methods are classification, clustering, association, and trend analysis. Short descriptions of these methods follow.

Classification

Arguably the most common knowledge discovery topic in analyzing complex data sources is the classification (or categorization) of certain objects. The task is to classify a given data instance into a predetermined set of categories (or classes). As it applies to the domain of text mining, the task is known as *text categorization*, where for a given set of categories (subjects, topics, or concepts) and a collection of text documents, the goal is to find the correct topic (subject or concept) for each document, using models developed with a training data set that includes both the documents and actual document categories. Today, automated text classification is applied in a variety of contexts, including automatic or semiautomatic (interactive) indexing of text, spam filtering, web page categorization under hierarchical catalogs, automatic generation of metadata, and detection of genre.

Clustering

Clustering is an unsupervised process whereby objects are classified into "natural" groups called *clusters*. In categorization, a collection of preclassified training examples is used to develop a model

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

based on the descriptive features of the classes in order to classify a new unlabeled example; on the other hand, in clustering, the problem is to group an unlabeled collection of objects (e.g., documents, customer comments, web pages) into meaningful clusters without any prior knowledge.

Clustering is useful in a wide range of applications, from document retrieval to better web content searching. In fact, one of the prominent applications of clustering is the analysis and navigation of very large text collections, such as web pages. The basic underlying assumption is that relevant documents tend to be more similar to each other than to irrelevant ones. If this assumption holds, the clustering of documents based on the similarity of their content improves search effectiveness.

Association

As discussed in Chapter 4, "Data and Methods for Predictive Analytics," the main idea in generating association rules (or solving market-basket problems) is to identify the sets that frequently go together. In text mining, *associations* specifically refers to the direct relationships between concepts (terms) or sets of concepts. The concept set association rule, relating two frequent concept sets A and C, can be quantified by the two basic measures support and confidence. In this case, *confidence* is the percentage of documents that include all the concepts in C within the same subset of those documents that include all the concepts in A. *Support* is the percentage (or number) of documents that include all the concepts in A and C. For instance, in a document collection, the concept *software implementation failure* may appear most often in association with *enterprise resource planning* and *customer relationship management*, with significant support (4%) and confidence (55%), meaning that 4% of the documents had all three concepts represented together in the same document, and of the documents that included *software implementation failure*, 55% of them also included *enterprise resource planning* and

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

customer relationship management. Mahgoub et al. (2008) used text mining with association rules to analyze published literature (news and academic articles posted on the Web) to chart the outbreak and progress of bird flu. The idea was to automatically identify the associations among the geographic areas, spread across species, and countermeasures (treatments).

Trend Analysis

Recent methods of trend analysis in text mining have been based on the notion that the various types of concept distributions are functions of document collections; that is, different collections lead to different concept distributions for the same set of concepts. It is therefore possible to compare two distributions that are otherwise identical except that they are from different subcollections. One notable direction of this type of analyses is having two collections from the same source (e.g., from the same set of academic journals) but from different points in time. Delen and Crossland (2008) applied trend analysis to a large number of academic articles published in the three highest-rated academic journals to identify the evolution of key concepts in the field of information systems. The following section describes this text mining application.

Application Example: Text Mining of Research Literature

Researchers conducting searches and reviews of relevant literature face an increasingly complex and voluminous task. In extending the body of relevant knowledge, it has always been important to work hard to gather, organize, analyze, and assimilate existing information from the literature, particularly from one's home discipline. With the increasing abundance of potentially significant research being reported in related fields, and even in what are traditionally deemed to be nonrelated fields of study, the researcher's task is ever more daunting, if a thorough job is desired.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

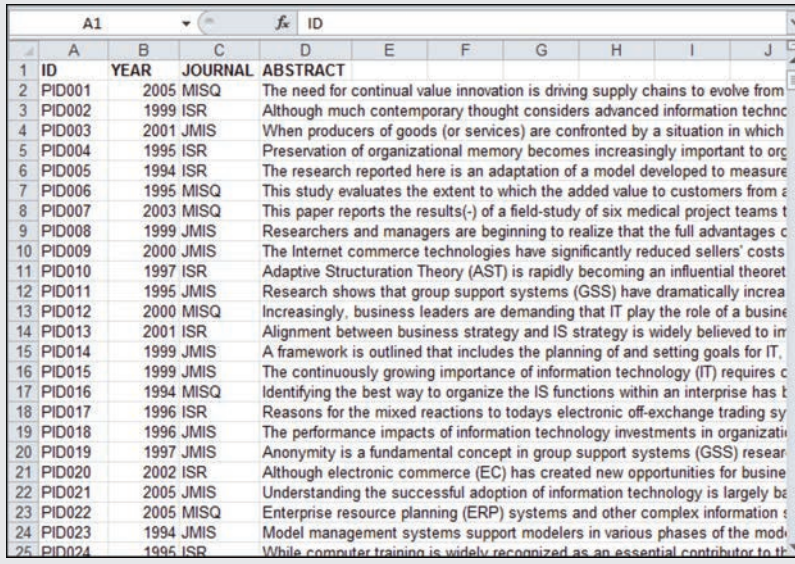
In new streams of research, the researcher's task may be even more tedious and complex. Trying to ferret out relevant work that others have reported may be difficult, at best, and it may be nearly impossible if traditional, largely manual reviews of published literature are required. Even with a legion of dedicated graduate students or helpful colleagues, trying to cover all potentially relevant published work is problematic.

Many scholarly conferences take place every year. In addition to extending the body of knowledge of the current focus of a conference, organizers often desire to offer additional mini-tracks and workshops. In many cases, these additional events are intended to introduce the attendees to significant streams of research in related fields of study and to try to identify the "next big thing" in terms of research interests and focus. Identifying reasonable candidate topics for such mini-tracks and workshops is often subjective rather than derived objectively from the existing and emerging research.

In their research, Delen and Crossland (2008) proposed a method to greatly assist and enhance the efforts of the researchers by enabling a semiautomated analysis of large volumes of published literature through the application of text mining. Using standard digital libraries and online publication search engines, the authors downloaded and collected all the available articles for the three major journals in the field of management information systems: *MIS Quarterly* (*MISQ*), *Information Systems Research* (*ISR*), and the *Journal of Management Information Systems* (*JMIS*). In order to maintain the same time interval for all three journals (for potential comparative longitudinal studies), the journal with the most recent starting date for its digital publication availability was used as the start time for this study (i.e., *JMIS* articles have been digitally available since 1994). For each article, Delen and Crossland extracted the title, abstract, author list, published keywords, volume, issue number, and year of publication. They then loaded all the article

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

data into a simple database file. Also included in the combined data set was a field that designated the journal type of each article for likely discriminatory analysis. Editorial notes, research notes, and executive overviews were omitted from the collection. Figure 7.6 shows how the data was presented.



	A	B	C	D
1	ID	YEAR	JOURNAL	ABSTRACT
2	PID001	2005	MISQ	The need for continual value innovation is driving supply chains to evolve from
3	PID002	1999	ISR	Although much contemporary thought considers advanced information techno
4	PID003	2001	JMIS	When producers of goods (or services) are confronted by a situation in which
5	PID004	1995	ISR	Preservation of organizational memory becomes increasingly important to org
6	PID005	1994	ISR	The research reported here is an adaptation of a model developed to measure
7	PID006	1995	MISQ	This study evaluates the extent to which the added value to customers from a
8	PID007	2003	MISQ	This paper reports the results(-) of a field-study of six medical project teams t
9	PID008	1999	JMIS	Researchers and managers are beginning to realize that the full advantages o
10	PID009	2000	JMIS	The Internet commerce technologies have significantly reduced sellers' costs
11	PID010	1997	ISR	Adaptive Structuration Theory (AST) is rapidly becoming an influential theoret
12	PID011	1995	JMIS	Research shows that group support systems (GSS) have dramatically increa
13	PID012	2000	MISQ	Increasingly, business leaders are demanding that IT play the role of a busine
14	PID013	2001	ISR	Alignment between business strategy and IS strategy is widely believed to im
15	PID014	1999	JMIS	A framework is outlined that includes the planning of and setting goals for IT,
16	PID015	1999	JMIS	The continuously growing importance of information technology (IT) requires o
17	PID016	1994	MISQ	Identifying the best way to organize the IS functions within an interprise has b
18	PID017	1996	ISR	Reasons for the mixed reactions to todays electronic off-exchange trading sy
19	PID018	1996	JMIS	The performance impacts of information technology investments in organizati
20	PID019	1997	JMIS	Anonymity is a fundamental concept in group support systems (GSS) resear
21	PID020	2002	ISR	Although electronic commerce (EC) has created new opportunities for busine
22	PID021	2005	JMIS	Understanding the successful adoption of information technology is largely ba
23	PID022	2005	MISQ	Enterprise resource planning (ERP) systems and other complex information s
24	PID023	1994	JMIS	Model management systems support modelers in various phases of the mod
25	PID024	1995	ISR	While computer training is widely recognized as an essential contributor to th

Figure 7.6 A Sample of the Textual Data Set

In the analysis phase, Delen and Crossland chose to use only the abstract of an article as the source of information extraction. They chose not to include the keywords listed with the publications for two main reasons:

- Under normal circumstances, the abstract would already include the listed keywords, and therefore inclusion of the listed keywords for the analysis would mean repeating the same information and potentially giving it unmerited weight.
- The listed keywords might be terms that authors would like their article to be associated with (as opposed to what is really contained in the article), therefore potentially introducing unquantifiable bias to the analysis of the content.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The first exploratory study looked at the longitudinal perspective of the three journals (i.e., evolution of research topics over time). In order to conduct a longitudinal study, Delen and Crossland divided the 12-year period from 1994 to 2005 into four 3-year periods for each of the three journals. This framework led to 12 text mining experiments with 12 mutually exclusive data sets. At this point, for each of the 12 data sets, Delen and Crossland used text mining to extract the most descriptive terms from these collections of articles represented by their abstracts. The results were tabulated and examined for time-varying changes in the terms published in these three journals.

As a second exploration, using the complete data set (including all three journals and all four periods), Delen and Crossland conducted a clustering analysis. Clustering is arguably the most commonly used text mining technique. Clustering was used in this study to identify the natural groupings of the articles (by putting them into separate clusters) and then to list the most descriptive terms that characterized those clusters. The authors used singular-value decomposition to reduce the dimensionality of the term–document matrix and then an expectation-maximization algorithm to create the clusters. They conducted several experiments to identify the optimal number of clusters, which turned out to be nine. After the construction of the nine clusters, Delen and Crossland analyzed the content of those clusters from two perspectives: representation of the journal type (see Figure 7.7) and representation of time (see Figure 7.8). The idea was to explore the potential differences and/or commonalities among the three journals and potential changes in the emphasis on those clusters; that is, they wanted to answer questions such as "Are there clusters that represent different research themes specific to a single journal?" and "Is there a time-varying characterization of those clusters?" They discovered and discussed several interesting patterns using tabular and graphical representation of their findings. (For further information, see Delen & Crossland, 2008.)

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

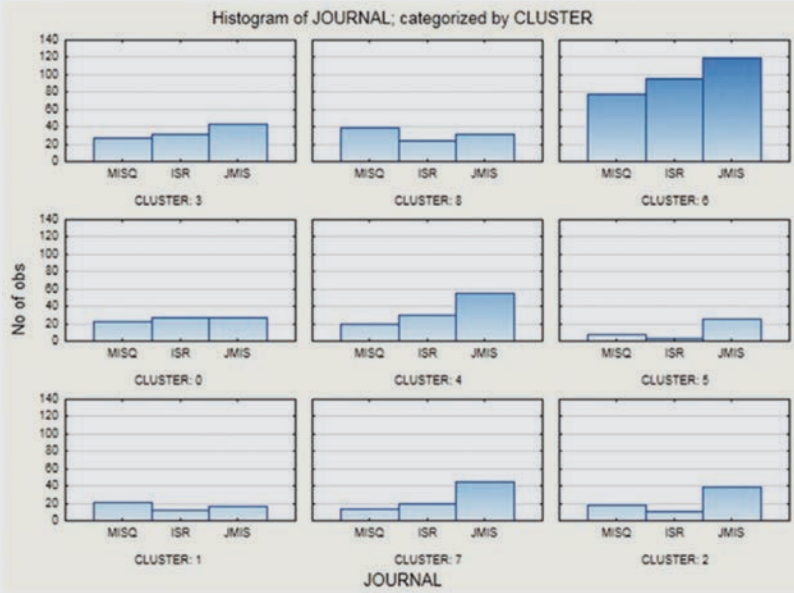


Figure 7.7 Distribution of the Number of Articles for Each Journal over the Nine Clusters

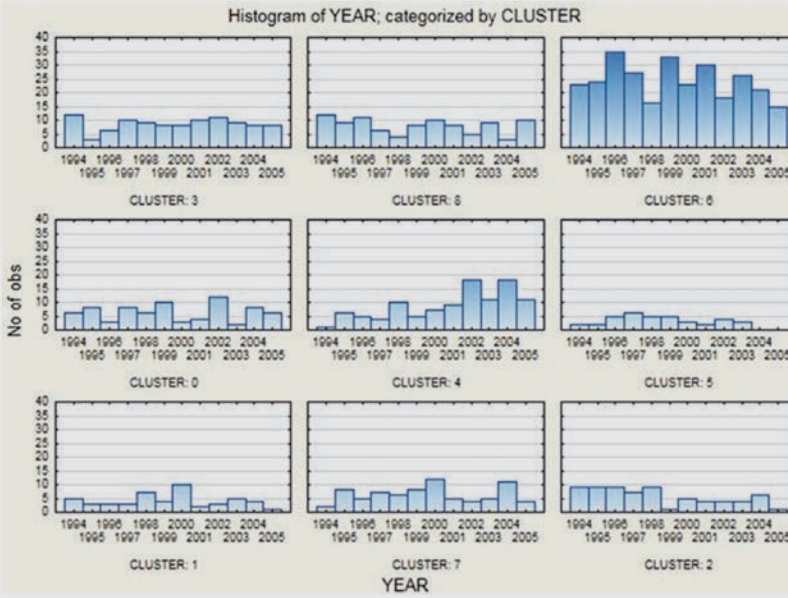


Figure 7.8 Time-Dependent Changes in the Number of Articles Published in Each Cluster

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Text Mining Tools

As more and more organizations are realizing the value of text mining, the number of software tools offered by software companies and nonprofits is also increasing. The following sections describe some of the popular text mining tools, which we classify as commercial software tools and free (and/or open source) software tools.

Commercial Software Tools

The following are some of the most popular software tools used for text mining. Note that many of these companies offer demonstration versions of their products on their websites:

- NVivo offers text analysis and visualization tools for qualitative, text-based data-driven research studies.
- IBM offers SPSS Modeler and Data and Text Analytics Toolkit.
- Megaputer Text Analyst offers semantic analysis of free-form text, summarization, clustering, navigation, and natural language retrieval with search-dynamic refocusing.
- SAS Text Miner provides a rich suite of text processing and analysis tools.
- JMP, a SAS software product, provides an easy-to-use set of text processing capabilities.
- RapidMiner has a graphically appealing user interface and is one of the popular software tools for data mining and text mining.
- KXEN (now part of SAP) Text Coder is a text analytics solution for automatically preparing and transforming unstructured text attributes into a structured representation for use in KXEN Analytic Framework.
- The Statistica Text Mining (now part of TIBCO) provides a suite of intuitive text mining functionally with exceptional visualization capabilities.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

- VantagePoint provides a variety of interactive graphical views and analysis tools with powerful capabilities to discover knowledge from text databases.
- The WordStat analysis module from Provalis Research analyzes textual information such as responses to open-ended questions and interviews.
- Clarabridge text mining software provides an end-to-end solution that enables customer experience professionals to transform customer feedback for marketing, service, and product improvements.

Free Software Tools

Free software tools, some of which are open source, are available from a number of nonprofit organizations:

- KINIME, one of the most popular free and open source analytics tool, provides a highly capable and rather comprehensive text processing extension.
- Open Calais is an open source toolkit for including semantic functionality within a blog, a content management system, a website, or an application.
- GATE is a leading open source toolkit for text mining. It has a free open source framework and graphical development environment.
- LingPipe is a suite of Java libraries for linguistic analysis of human language.
- S-EM (Spy-EM) is a text classification system that learns from positive and unlabeled examples.
- Vivisimo/Clusty is a web search and text-clustering engine.

Often, several software tools can be used synergistically to best apply text mining to complex problems and data sets.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Topic Modeling

Topic modeling, also known as *topic detection*, is the use of a suite of probabilistic machine learning algorithms to discover and annotate large archives of documents with thematic information. Topic modeling algorithms are statistical methods that analyze the words of the original texts to discover the themes that run through them, how those themes are connected to each other, and how they change over time. Two common assumptions underlie all topic modeling methods: Each document consists of a mix of topics, and each topic consists of a collection of words/terms, and the topics are "hidden" or "latent" constructs in between documents and words. The goal of topic modeling is to discover the latent variables (i.e., topics) that shape the meaning/semantics in a document collection.

There have been numerous attempts to accurately identify inherent topics in a collection of documents. Perhaps the earliest and most rudimentary attempt at topic detection was the use of clustering to group documents based on word frequencies in the documents (in the form of a term–document matrix) using a traditional clustering technique, such as hierarchical or *k*-means clustering algorithms. This method provided mathematical means to group documents (i.e., put them into predetermined number of clusters) based on how frequently certain words/terms appeared in them and then label the clusters with arbitrary topic names based on the most frequently used terms.

The next popular topic modeling approach was called *latent semantic analysis* (*LSA*), or *latent semantic indexing* (*LSI*). LSA and LSI are used interchangeably; whereas the machine learning/text mining community prefers the term LSA, the information retrieval community likes LSI. The core idea behind LSA is to take the numericized representation of a given corpus (i.e., a term–document matrix) and disperse/factor it into its component matrices. Specifically, in the process of conducting LSA, the first step is to generate

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

the term–document matrix. Given m documents and n words in the vocabulary, we construct an m × n matrix A, in which each row represents a document and each column represents a word. In the simplest version of LSA, each entry can simply be a raw count of the number of times the jth word appeared in the ith document. In practice, however, raw counts do not work particularly well because they overstate the importance of counts and do not consider the significance of unique words in the document. Instead, the common practice today is to use term frequency–inverse document frequency (tf-idf) for a better representation of the indices. The simple formula for tf-idf is as follows:

$$\underbrace{[tf\text{-}idf]_{i,j}}_{\text{Score for document } i \text{ and term } j} = \overbrace{[tf]_{i,j}}^{\text{Number of occurrences of term } j \text{ in document } i} \times log \frac{N}{\underbrace{[df]_j}_{\text{Number of document containing the term } j}}$$

Total number of documents: N

Once we have the term–document matrix A containing the indices generated using tf-idf, we can start thinking about identifying the latent *topics*. Term–document matrix A usually is a very sparse, noisy, and redundant matrix across its dimensions. Therefore, to find the latent topics that capture the relationships among the terms and documents, we need to perform some type of dimensionality reduction on A. In LSA, this dimensionality reduction is performed using singular value decomposition (SVD). SVD is a linear algebra technique used to factorize the matrix (A) into the product of three separate matrices: a term–concept matrix (U), a singular value matrix (S), and a concept–document matrix (V) in the form $A = U \times S \times V$, where S

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

is a diagonal matrix of the singular values of *A*. Then, as a final step, we group the documents into latent topics by taking the cosine of the angle between two vectors, where the resultant values close to 1 represent very similar documents, while values close to 0 represent very dissimilar documents.

One of the shortcomings of the earlier topic modeling methods is that they assumed the existence of an explicit relationship between the topics and documents. For example, they assumed that a document belongs to only one topic, but in reality, a document can belong to multiple topics to varying degrees, and each word/term can belong to each topic to varying degrees. This shortcoming is addressed in the latent Dirichlet allocation, which is discussed next.

Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is arguably the most popular and perhaps most effective topic detection technique of this decade. Although more contemporary attempts have been made to better detect/identify latent topics in textual data—some leveraging deep learning (e.g., RNN/LSTM-type neural network architectures, which are covered in Chapter 6, "Advanced Topics in Predictive Modeling") and others using Word Embedding/Word2Vec types of approaches—LDA is still the most commonly used approach. LDA uses Dirichlet priors/distributions for the document-to-topic and topic-to-term associations/allocations. Dirichlet distribution, the core function of LDA, is a family of continuous multivariate probability distributions parameterized by a vector of positive reals. It is, in fact, a multivariate generalization of the beta distribution, and hence it is also called *multivariate beta distribution*.

LDA is a generative probabilistic model that uses an unsupervised learning process: Given a set of training data, it aims to identify the underlying distribution by generating samples from the same distribution. At the highest level, LDA portrays a three-level

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

hierarchical probabilistic allocation model, where each document is modeled as a weighted mixture of the underlying set of topics, and each topic, in turn, is modeled as a weighted mixture of the underlying set of terms. Figure 7.9 illustrates the hierarchical structure of LDA.

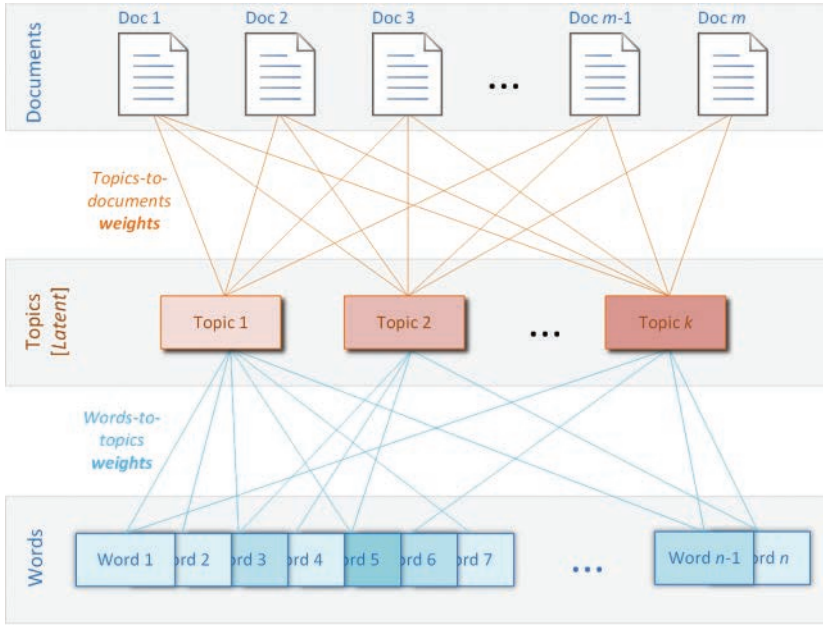
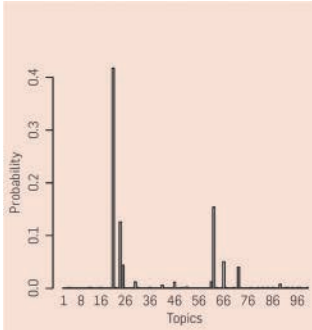


Figure 7.9 A Graphical Depiction of the Components of LDA: Documents, Words, and Latent Topics

The algorithmic details of LDA can be found in Blei et al. (2003), which is considered the seminal work for the LDA algorithm. In order to further explain the internal structure and superior applicability of LDA, the lead author on the seminal work, David M. Blei, a professor at University of California, Berkeley, presented several illustrative examples in his *Communications of the ACM* review article (Blei, 2012). In one of those examples, he fit a 100-topic LDA model to 17,000 articles from the journal *Science*. Figure 7.10 shows the results

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

of this study, where the inferred topic proportions for the example are presented on the left, and the top 15 frequent (i.e., weighted) words from the most prevalent four latent topics are given on the right.



"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

Figure 7.10 An Example of LDA Output for a Large Collection of Scientific Literature

Sentiment Analysis

We humans are social beings. We are adept at using a variety of means to exchange ideas. We often consult financial discussion forums before making investment decisions, we ask our friends for their opinions on newly opened restaurants and newly released movies, and we conduct Internet searches and read consumer reviews and expert reports before making a big purchase such as a house, a car, or an appliance. We rely on others' opinions to make better decisions, especially in areas where we don't have a lot of knowledge or experience. Thanks to the growing availability and popularity of opinion-rich Internet resources such as social media outlets (e.g., Twitter, Facebook), online review sites, and personal blogs, it is now easier than ever before to find opinions of others—thousands of them—on everything from the latest gadgets to political and public figures. Even

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

though not everybody expresses opinions on the Internet, due to fast-growing numbers and capabilities of social communication channels, the numbers are increasing exponentially.

Sentiment is a difficult word to exactly define. It is often linked to or confused with other terms, such as *belief*, *view*, *opinion*, and *conviction*. *Sentiment* suggests a subtle opinion reflective of one's feelings (Mejova, 2009). Sentiment has some unique properties that sets it apart from other concepts that we might want to identify in text. Often we want to categorize text by topic, which may involve dealing with whole taxonomies of topics. Sentiment classification, on the other hand, usually deals with two classes (positive versus negative), a range of polarity (e.g., star ratings for movies), or even a range in strength of opinion (Pang & Lee, 2008). These classes span many topics, users, and documents. Although dealing with only a few classes may seem like an easier task than standard text analysis, this is far from the reality.

As a field of research, sentiment analysis is closely related to computational linguistics, natural language processing, and text mining. Sentiment analysis is also often referred to as *opinion mining*, *subjectivity analysis*, and *appraisal extraction*, and it has some connections to *affective computing* (i.e., computer recognition and expression of emotion). The sudden upsurge of interest and activity in the area of sentiment analysis, which deals with the automatic extraction of opinions, feelings, and subjectivity in text, is creating opportunities and threats for business and individuals alike. The ones who embrace and take advantage of it will greatly benefit from it. Every opinion (good or bad) put on the Internet by an individual or a company will be attributed back to the originator and will be retrieved and mined by others (often automatically by computer programs) for a variety of purposes.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Sentiment analysis tries to answer the question "What do people feel about a certain topic?" by digging into the opinions of many, using a variety of automated tools. Bringing together researchers and practitioners in business, computer science, computational linguistics, data mining, text mining, psychology, and even sociology, sentiment analysis aims to expand traditional fact-based text analysis to new frontiers to realize opinion-oriented information systems. In a business setting, especially in marketing and customer relationship management, sentiment analysis seeks to detect favorable and unfavorable opinions about specific products and/or services, using a large number of textual data sources (e.g., customer feedback in the form of web postings, tweets, blogs).

Sentiment that appears in text comes in two flavors: explicit, where the text directly expresses an opinion ("It's a wonderful day"), and implicit, where the text implies an opinion ("The handle breaks too easily"). Most of the earlier work done in sentiment analysis focused on the first kind of sentiment, since it is was easier to analyze. Current trends are to implement analytical methods to consider both implicit and explicit sentiments. Sentiment polarity is a particular feature of text that sentiment analysis primarily focuses on. It is usually dichotomized into two areas—positive and negative—but polarity can also be thought of as a range. A document containing several opinionated statements would have a mixed polarity overall, which is different from not having a polarity at all (i.e., being objective) (Mejova, 2009).

Sentiment Analysis Applications

Traditional sentiment analysis methods were survey based or focus group centered; they were costly and time-consuming and, hence, were driven from a small sample of participants. In contrast, the new face of text analytics–based sentiment analysis is a limit breaker. Current solutions automate a very large-scale data collection,

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

filtering, classification, and clustering methods via natural language processing and data mining technologies that handle both factual and subjective information. Sentiment analysis is perhaps the most popular application of text analytics, tapping into data sources such as tweets, Facebook posts, online communities, discussion boards, blogs, product reviews, call center logs and recordings, product rating sites, chat rooms, price comparison portals, search engine logs, and newsgroups. The following sections discuss applications of sentiment analysis to illustrate the power and the widespread coverage of this technology.

Voice of the Customer (VOC)

VOC is an integral part of an analytic CRM and customer experience management systems. As the enabler of VOC, sentiment analysis can access a company's product and service reviews (either continuously or periodically) to better understand and better manage customer complaints and praises. For instance, a motion picture advertising/marketing company might detect negative sentiments toward a movie that is about to open in theaters (based on its trailers) and quickly change the composition of the trailers and its advertising strategy (on all media outlets) to mitigate the negative impact. Similarly, a software company may detect negative buzz regarding the bugs found in its newly released product early enough to release patches and quick fixes to alleviate the situation.

Often, the focus of VOC is individual customers and their service- and support-related needs, wants, and issues. VOC draws data from the full set of customer touchpoints, including emails, surveys, call center notes/recordings, and social media postings, and it matches customer voices to transactions (e.g., inquiries, purchases, returns) and individual customer profiles captured in enterprise operational systems. VOC, mostly driven by sentiment analysis, is a key element of customer experience management initiatives, where the goal is to create an intimate relationship with the customer.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Voice of the Market (VOM)

VOM is about understanding aggregate opinions and trends. It's about knowing what stakeholders—customers, potential customers, influencers, others—are saying about your, and your competitors', products and services. A well-done VOM analysis helps companies with competitive intelligence and product development and positioning.

Voice of the Employee (VOE)

Traditionally, VOE has been limited to employee satisfaction surveys. Text analytics in general and sentiment analysis specifically are huge enablers of assessing VOE. Using rich, opinionated textual data is an effective and efficient way to listen to what employees are saying. As we all know, happy employees empower customer experience efforts and improve customer satisfaction.

Brand Management

Brand management focuses on listening to social media, where anyone (e.g., past/current/prospective customers, industry experts, other authorities) can post opinions that can damage or boost a reputation. A number of relatively newly launched startup companies offer analytics-driven brand management services. Brand management is product and company focused rather than customer focused. It attempts to shape perceptions rather than to manage experiences, using sentiment analysis techniques.

Financial Markets

Predicting the future values of individual (or groups of) stocks has been an interesting and seemingly unsolvable problem. What makes a stock (or a group of stocks) move up or down is anything but an exact science. Many believe that the stock market is mostly sentiment

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

driven, making it anything but rational (especially for short-term stock movements). Therefore, use of sentiment analysis in financial markets has gained significant popularity. Automated analysis of market sentiments using social media, news, blogs, and discussion groups seems to be a proper way to compute market movements. If done correctly, sentiment analysis can identify short-term stock movements based on the buzz in the market, potentially impacting liquidity and trading.

Politics

As we all know, opinions matter a great deal in politics. Because political discussions are dominated by quotations, sarcasm, and complex references to persons, organizations, and ideas, politics is one of the most difficult—and potentially fruitful—areas for sentiment analysis. By analyzing the sentiment on election forums, it may be possible to predict who is more likely to win or lose. Sentiment analysis can help understand what voters are thinking and can clarify a candidate's position on issues. Sentiment analysis can help political organizations, campaigns, and news analysts better understand which issues and positions matter the most to voters. The technology was successfully applied by both parties to every campaign since the 2008 U.S. presidential election.

Government Intelligence

Government intelligence is another application of sentiment analysis. For example, it has been suggested that one could monitor sources for increases in hostile or negative communications. Sentiment analysis can allow the automatic analysis of the opinions that people submit about pending policy or government-regulation proposals. Furthermore, monitoring communications for spikes in negative sentiment may be of use to agencies like the Department of Homeland Security.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Other Interesting Areas

Sentiments of customers can be used to better design ecommerce sites (e.g., product suggestions, upsell/cross-sell advertising), better place advertisements (e.g., placing dynamic advertisement of products and services that consider the sentiment on the page the user is browsing), and mine opinion- or review-oriented search engines (i.e., opinion-aggregation websites, as alternatives to sites like Epinions, to summarize user reviews). Sentiment analysis can help with email filtration by categorizing and prioritizing incoming emails; for example, it can detect strongly negative or flaming emails and forward them to a proper folder. It can also help with citation analysis by determining whether an author is citing a piece of work as supporting evidence or as research that he or she dismisses.

The Sentiment Analysis Process

Because of the complexity of the problem of sentiment analysis (e.g., underlying concepts, expressions in text, the context in which text is expressed), there is not a readily available standardized process for conducting sentiment analysis. However, based on the published work in the field of sentiment analysis so far (both on research methods and range of applications), a multistep, simple, logical process, as shown in Figure 7.11, seems to be an appropriate methodology for sentiment analysis. These logical steps are iterative—that is, feedback, correction, and iteration are all part of the discovery process—and experimental in nature, and once completed and combined, they are capable of producing desired insight about the opinions in the text collection.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

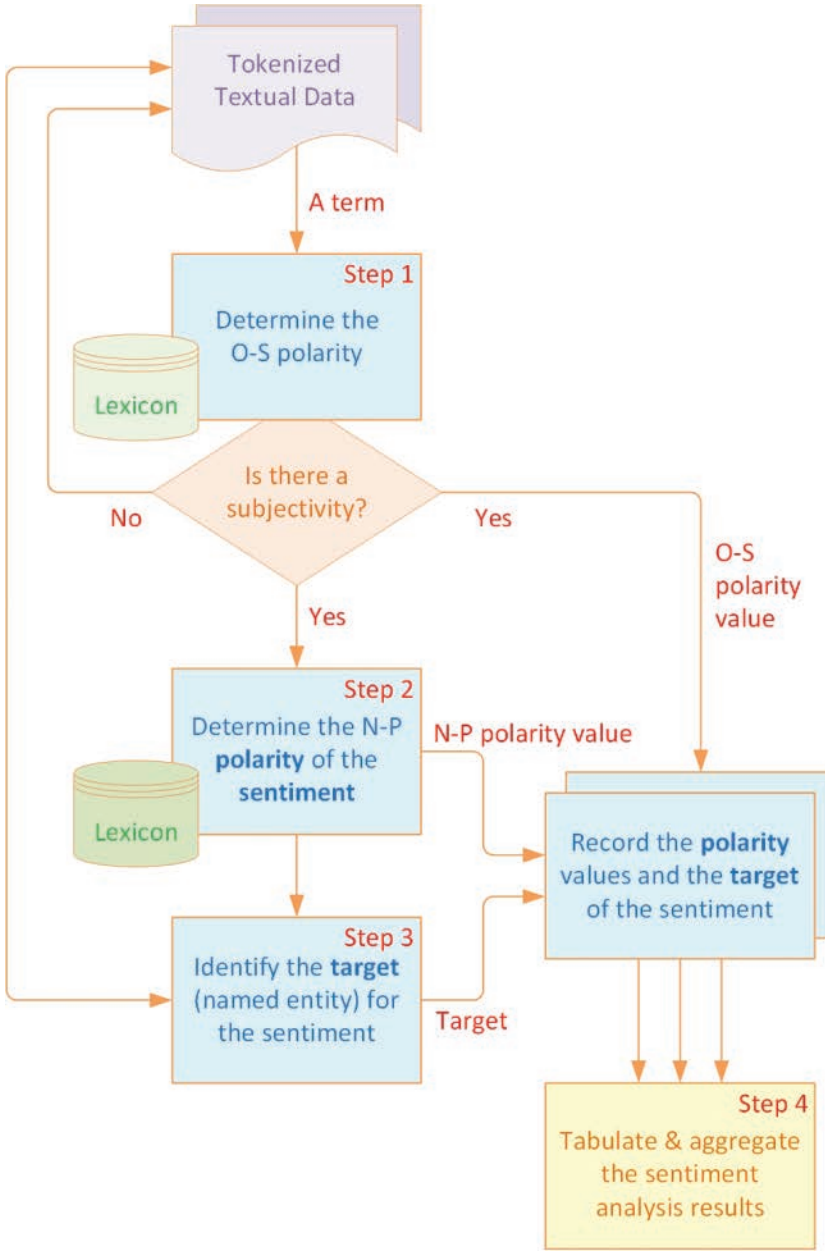


Figure 7.11 A Multistep Process for Sentiment Analysis

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Step 1: Sentiment Detection

After the retrieval and preparation of the text documents, the first main task in sensitivity analysis is the detection of objectivity. Here the goal is to differentiate between a fact and an opinion, which may be viewed as classification of text as objective or subjective. This may also be characterized as calculation of objectivity–subjectivity (O-S) polarity, which may be represented with a numeric value ranging from 0 to 1. If the objectivity value is close to 1, then there is no opinion to mine (i.e., it is a fact), and therefore the process goes back and grabs the next text data to analyze. Usually opinion detection is based on the examination of adjectives in text. For example, the polarity of "what a wonderful work" can be determined relatively easily by looking at the adjective.

Step 2: N-P Polarity Classification

The second main task is polarity classification. Given an opinionated piece of text, the goal is to classify the opinion as falling under one of two opposing sentiment polarities or locate its position on the continuum between these two polarities (Pang & Lee, 2008). When viewed as a binary feature, polarity classification is the binary classification task of labeling an opinionated document as expressing either an overall positive opinion or an overall negative opinion (e.g., thumbs up/thumbs down). In addition to the identification of N-P (negative–positive) polarity, one should also be interested in identifying the strength of the sentiment; as opposed to just positive, it may be expressed as mildly, moderately, strongly, or very strongly positive. A lot of research has been done on product or movie reviews where the definitions of "positive" and "negative" are quite clear. Other tasks, such as classifying news as "good" or "bad," present some difficulty. For instance, an article may contain negative news without explicitly using any subjective words or terms. Furthermore, these classes usually appear intermixed when a document expresses both

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

positive and negative sentiments. Then the task can be to identify the main (or dominating) sentiment of the document. Still, for lengthy texts, the tasks of classification may need to be done at several levels: term, phrase, sentence, and perhaps document level. In such cases, it is common to use the outputs of one level as the inputs for the next higher layer.

Step 3: Target Identification

The goal of this step is to accurately identify the target of the expressed sentiment (e.g., a person, a product, an event). The difficulty of this task depends largely on the domain of the analysis. Even though it is usually easy to accurately identify the target for product or movie reviews, because the review is directly connected to the target, it may be quite challenging in other domains. For instance, lengthy, general-purpose text such as web pages, news articles, and blogs do not always have predefined topics that they are assigned to, and they often mention many objects, any of which may be deduced as the target. Sometimes there is more than one target in a sentiment sentence, as is the case in comparative texts. A subjective comparative sentence orders objects in terms of preferences, such as "This laptop computer is better than my desktop PC." Such sentences can be identified using comparative adjectives and adverbs (e.g., *more*, *less*, *better*, *longer*), superlative adjectives (e.g., *most*, *least*, *best*) and other words, such as *same*, *differ*, *win*, and *prefer*. Once the sentences have been retrieved, the objects can be put in an order that is most representative of their merits.

Step 4: Collection and Aggregation

Once the sentiments of all text data points in the document are identified and calculated, in this step, they are aggregated and converted to a single sentiment measure for the whole document. This

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

aggregation may be as simple as summing up the polarities and strengths of all texts or as complex as using semantic aggregation techniques from natural language processing to come up with the ultimate sentiment.

The Methods for Polarity Identification

As mentioned in the previous section, polarity identification can be made at the word, term, sentence, or document level. The most granular level for polarity identification is the word level. After the polarity identification is made at the word level, it can be aggregated to the next higher level, and then the next…until the level of aggregation desired from the sentiment analysis is reached. There seem to be two dominant techniques used for identification of polarity at the word/term level, each of which has advantages and disadvantages:

- Using a lexicon as a reference library—either developed manually or automatically by an individual for a specific task or developed by an institution for general use
- Using a collection of training documents as the source of knowledge about the polarity of terms in a specific domain (i.e., inducing predictive models from opinionated textual documents)

Application Example: Text-Based Deception Detection

Driven by advancements in web-based information technologies and increasing globalization, computer-mediated communication continues to filter into everyday life, bringing with it new venues for deception. The volume of text-based chat, instant messaging, text messaging, and text generated by online communities of practice is increasing rapidly. Even email continues to grow in use. With the massive growth of text-based communication, the potential for people to deceive others through computer-mediated

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

communication has also grown, and such deception can have disastrous results.

Unfortunately, in general, humans tend to perform poorly at deception-detection tasks. This phenomenon is exacerbated in text-based communications. A large part of the research on deception detection (also known as *credibility assessment*) has involved face-to-face meetings and interviews. However, with the growth of text-based communication, text-based deception-detection techniques are essential.

Techniques for successfully detecting deception (i.e., lies) have wide applicability. Law enforcement can use decision support tools and techniques to investigate crimes, conduct security screening in airports, and monitor communications of suspected terrorists. Human resources professionals might use deception-detection tools to screen applicants. These tools and techniques also have the potential to screen emails to uncover fraud or other wrongdoings committed by corporate officers. Although some people believe that they can readily identify people not being truthful, a summary of deception research showed that, on average, people are only 54% accurate in making veracity determinations (Bond & DePaulo, 2006). This figure may actually be worse when humans try to detect deception in text.

Using a combination of text mining and data mining techniques, Fuller et al. (2008) analyzed person-of-interest statements completed by people involved in crimes on military bases. In such statements, suspects and witnesses are required to write their recollection of the event in their own words. Military law enforcement personnel searched archival data for statements that they could conclusively identify as being truthful or deceptive. These decisions were made on the basis of corroborating evidence and case resolution. Once the statements were labeled as truthful or deceptive, the law enforcement personnel removed identifying

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

information and gave the statements to the research team. In total, 371 usable statements were received for analysis. The text-based deception-detection method used by Fuller et al. (2008) was based on a process known as *message feature mining*, which relies on elements of data and text mining techniques. A simplified depiction of the process is provided in Figure 7.12.

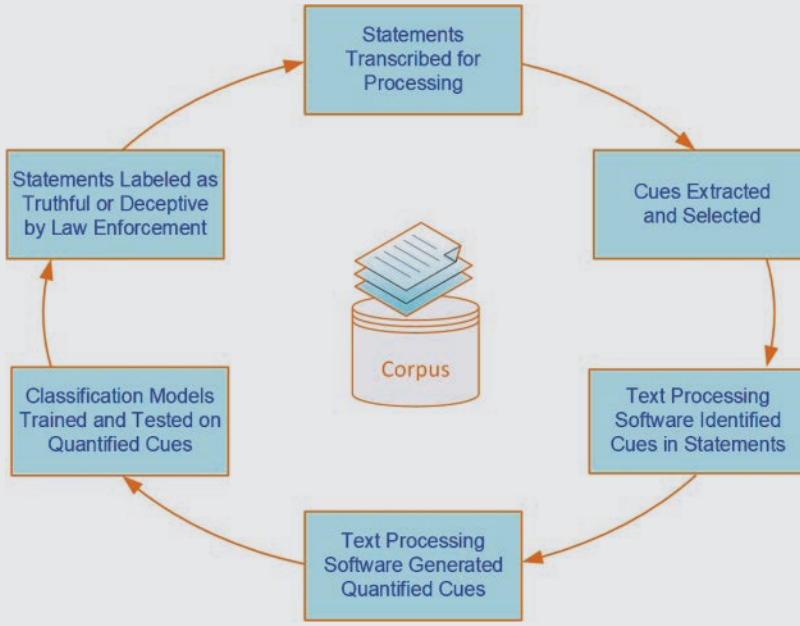


Figure 7.12 Text-Based Deception-Detection Process

First, the researchers prepared the data for processing. The original handwritten statements had to be transcribed into a word processing file. Second, features (i.e., cues) were identified. The researchers identified 31 features representing categories or types of language that are relatively independent of the text content and that can be readily analyzed by automated means. For example, first-person pronouns such as *I* or *me* can be identified without analysis of the surrounding text. Table 7.1 lists the categories used in this study and examples of cues.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Table 7.1 Categories and Examples of Linguistic Features Used in Deception Detection

Number	Category	Examples of Cues
1	Quantity	Verb count, noun-phrase count, etc.
2	Complexity	Average number of clauses, average sentence length, etc.
3	Uncertainty	Modifiers, modal verbs, etc.
4	Nonimmediacy	Passive voice, objectification, etc.
5	Expressivity	Emotiveness
6	Diversity	Lexical diversity, redundancy, etc.
7	Informality	Typographical error ratio
8	Specificity	Spatiotemporal information, perceptual information, etc.
9	Affect	Positive affect, negative affect, etc.

The features were extracted from the textual statements and input into a flat file for further processing. Using several feature-selection methods along with 10-fold cross-validation, the researchers compared the prediction accuracy of three popular data mining methods. Their results indicated that neural network models performed the best, with 73.46% prediction accuracy on test data samples; decision trees performed second best, with 71.60% accuracy; and logistic regression was last, with 67.28% accuracy.

The results indicate that automated text-based deception detection has the potential to aid those who must try to detect lies in text and can be successfully applied to real-world data. The accuracy of these techniques exceeded the accuracy of most other deception-detection techniques, even though it was limited to textual cues.

Summary

This chapter is dedicated to text analytics, which has become progressively more popular in business analytics and data science, thanks to the tremendous amount of textual content created over the Internet

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

via Web 2.0 and 3.0 as well as the social media outlets. Text analytics goes by a variety of different names, including text mining, text processing, and knowledge discovery in textual data, among others. Regardless of the exact name used, the text analytics process involves turning unstructured textual data into actionable information through the application of natural language processing (NLP) and predictive analytics.

In addition to explaining the specific language of text analytics, the text analytics process, and a variety of text analytics applications in business, this chapter covers two hugely popular areas of practice in text mining: topic modeling and sentiment analysis. Whereas topic modeling attempts to identify the latent topics in large quantities of text documents, sentiment analysis tries to automatically identify the emotions and opinions embedded in user-generated textual content.

References

Blei, D. M. (2012). "Probabilistic Topic Models," *Communications of the ACM*, 55(4): 77–84.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, 3: 993–1022.

Bond C. F., & B. M. DePaulo. (2006). "Accuracy of Deception Judgments," *Personality and Social Psychology Reports*, 10(3): 214–234.

Chun, H. W., Y. Tsuruoka, J. D. Kim, R. Shiba, N. Nagata, T. Hishiki, & J. Tsujii. (2006, January). "Extraction of Gene-Disease Relations from Medline Using Domain Dictionaries and Machine Learning," *Pacific Symposium on Biocomputing*, 11: 4–15.

Delen, D., & M. Crossland. (2008). "Seeding the Survey and Analysis of Research Literature with Text Mining," *Expert Systems with Applications*, 34(3): 1707–1720.

Etzioni, O. (1996). "The World Wide Web: Quagmire or Gold Mine?" *Communications of the ACM*, 39(11): 65–68.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

EUROPOL. (2007). *EUROPOL Work Program 2007*, http://lastradainternational.org/lsidocs/42%20Europol%20Work%20Programme%202007.pdf (accessed September 2020).

Feldman, R., & J. Sanger. (2007). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. ABS Ventures.

Fuller, C. M., D. Biros, & D. Delen. (2008). "Exploration of Feature Selection and Advanced Classification Models for High-Stakes Deception Detection," *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS)*. IEEE Press.

Ghani, R., K. Probst, Y. Liu, M. Krema, & A. Fano. (2006). "Text Mining for Product Attribute Extraction," *SIGKDD Explorations*, 8(1): 41–48.

Grimes, S. (2011, February 17). "Seven Breakthrough Sentiment Analysis Scenarios," *InformationWeek*.

Kanayama, H., & T. Nasukawa. (2006). *Fully Automatic Lexicon Expanding for Domain-Oriented Sentiment Analysis, EMNLP: Empirical Methods in Natural Language Processing*, https://www.aclweb.org/anthology/W06-1642.pdf (accessed September 2020).

Lin, J., & D. Demner-Fushman. (2005). "'Bag of Words' Is Not Enough for Strength of Evidence Classification," *AMIA Annual Symposium Proceedings*, pp. 1031–1032.

Mahgoub, H., D. Rösner, N. Ismail, & F. Torkey. (2008). "A Text Mining Technique Using Association Rules Extraction," *International Journal of Computational Intelligence*, 4(1): 21–28.

Manning, C. D., P. Raghavan & H. Schutze. (2008). *Introduction to Information Retrieval*. MIT Press.

Masand, B. M., M. Spiliopoulou, J. Srivastava, & O. R. Zaïane. (2002). "Web Mining for Usage Patterns and Profiles," *SIGKDD Explorations*, 4(2): 125–132.

McKnight, W. (2005, January 1). "Text Data Mining in Business Intelligence," *Information Management Magazine*.

Mejova, Y. (2009). *Sentiment Analysis: An Overview*, https://www.academia.edu/291678/Sentiment_Analysis_An_Overview (accessed September 2020).

Miller, T. W. (2005). *Data and Text Mining: A Business Applications Approach*. Prentice Hall.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Nakov, P., A. Schwartz, B. Wolf, & M. A. Hearst. (2005). "Supporting Annotation Layers for Natural Language Processing," *Proceedings of the ACL*, pp. 65–68.

Pang, B., & L. Lee. (2008). *Opinion Mining and Sentiment Analysis*, http://www.cs.cornell.edu/home/llee/omsa/omsa.pdf (accessed September 2020).

Peterson, E. T. (2008). *The Voice of Customer: Qualitative Data as a Critical Input to Web Site Optimization*, http://foreseeresults.com/Form_Epeterson_WebAnalytics.html (accessed May 2009).

Shatkay, H., A. Hoglund, S. Brady, T. Blum, et al. (2007). "SherLoc: High-Accuracy Prediction of Protein Subcellular Localization by Integrating Text and Protein Sequence Data," *Bioinformatics*, 23(11): 1410–1417.

StatSoft. (2014). *Statistica Data and Text Miner User Manual*. StatSoft, Inc.

Weng, S. S., & C. K. Liu. (2004) "Using Text Classification and Multiple Concepts to Answer E-mails," *Expert Systems with Applications*, 26(4): 529–543.

Zhou, Y., E. Reid, J. Qin, H. Chen, & G. Lai. (2005). "U.S. Domestic Extremist Groups on the Web: Link and Content Analysis," *IEEE Intelligent Systems*, 20(5): 44–51.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

8

Big Data for Predictive Analytics

Using data to understand customers or clients and business operations to sustain (and foster) growth and profitability is an increasingly challenging task for today's enterprises. As more and more data becomes available in various forms and fashion, timely processing of the data with traditional means becomes impractical. This phenomenon is called *big data*, and it is receiving substantial press coverage and drawing increasing interest from both business users and IT professionals—so much so, in fact, that the term *big data* is becoming an overhyped and overused marketing buzzword.

Big data means different things to people with different backgrounds and interests. Initially, the term *big data* was used to describe the massive volumes of data analyzed by huge organizations like Google or research science projects at NASA. But for most businesses, it's a relative term: *Big* depends on an organization's size. The point is more about finding new value within and outside conventional data sources. Pushing the boundaries of data analytics uncovers new insights and opportunities, and *big* depends on where you start and how you proceed. Consider this popular description of big data: Big data exceeds the reach of commonly used hardware environments and/or capabilities of software tools to capture, manage, and process it within a tolerable time span for its user population. Big data has become a popular term to describe the exponential growth,

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

availability, and use of information, both structured and unstructured. Much has been written on the big data trend and how big data can serve as the basis for innovation, differentiation, and growth.

Where Does Big Data Come From?

A simple answer is "everywhere." Sources of data that were at one time ignored due to technical limitations are now treated as gold mines. Big data may come from any number of sources, including blogs, RFID tags, GPS, sensor networks, social networks, Internet-based text documents, Internet search indices, detailed call records, astronomy, atmospheric science, biology, genomics, nuclear physics, biochemical experiments, medical records, scientific research, military surveillance, photography archives, video archives, and large-scale ecommerce practices.

Figure 8.1 illustrates the sources of big data in a three-level diagram. The traditional data sources—mainly business transactions—are illustrated as the first echelon, where the volume, variety, and velocity of the data are moderate to low. The next echelon is the data generated by the Internet and social media. This human-generated data is perhaps the most complicated and potentially most valuable for understanding collective ideas and perceptions of people. The volume, variety, and velocity of data in this echelon are moderate to high. The topmost echelon is machine-generated data. Due to the automation of data collection systems on many fronts and the Internet of Things (which connects everything to everything else), organizations are now able to collect data at volumes and richness that were unimaginable just a few years ago. All three echelons of data sources create a wealth of information that can significantly improve an organization's capability of solving complex problems and taking advantage of opportunities—if recognized and leveraged properly.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

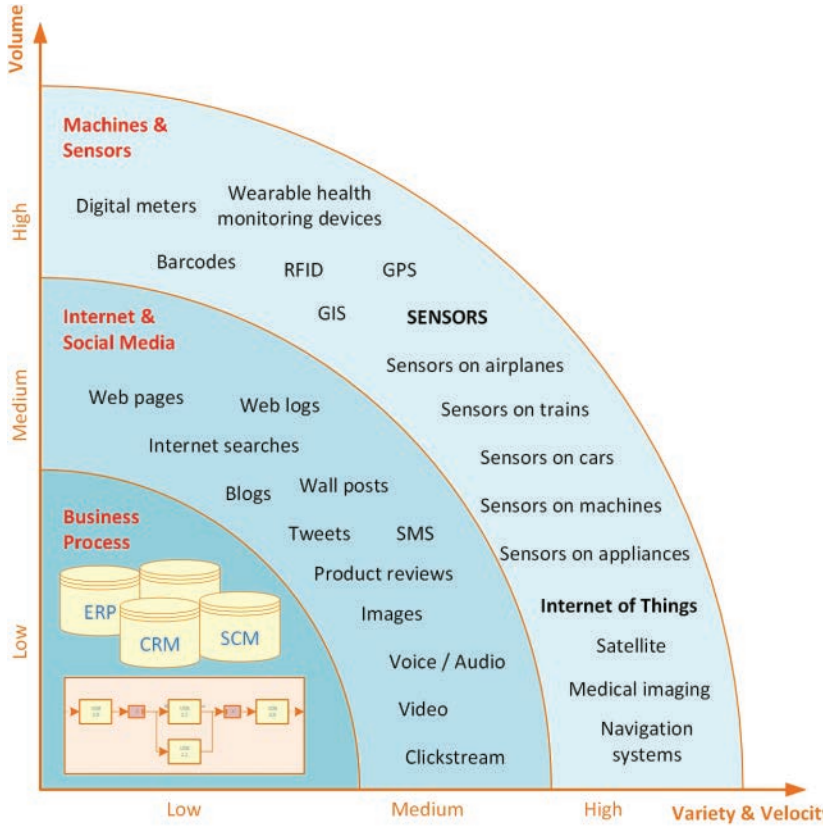


Figure 8.1 The Wide Range of Sources for Big Data

Big data is not new. What is new is that the definition and the structure of big data constantly change. Companies have been storing and analyzing large volumes of data since the advent of the data warehouses in the early 1990s. While terabytes used to be synonymous with big data warehouses, now it's petabytes, and the rate of growth in data volumes continues to escalate as organizations seek to store and analyze greater levels of transaction details, as well as web- and machine-generated data, to gain a better understanding of customer behavior and business drivers. Many academics and industry analysts and leaders alike think that "big data" is a misnomer. What it says and what it means are not exactly the same. That is, big data is not just

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

"big." The sheer volume of the data is only one of many characteristics that are often associated with big data; others include variety, velocity, veracity, variability, and value proposition.

The *V*s That Define Big Data

Big data is typically defined by three *V*s: volume, variety, and velocity. In addition to these three, we see some of the leading big data solution providers adding other *V*s, such as veracity (IBM), variability (SAS), and value proposition (literally almost everyone—academics and industry).

Volume

Volume is obviously the most common trait of big data. Many factors have contributed to the exponential increase in data volume, such as transaction-based data stored through the years, text data constantly streaming in from social media, increasing amounts of sensor data being collected, automatically generated RFID and GPS data, and so on. In the past, excessive data volume created storage issues, both technical and financial. But with today's advanced technologies coupled with decreasing storage costs, these issues are no longer significant; instead, other issues emerge, including how to determine relevance amid the large volumes of data and how to create value from data that is deemed to be relevant.

As mentioned earlier, *big* is a relative term. It changes over time and is perceived differently by different organizations. With the staggering increase in data volume, even the naming of the next big data echelon has been challenging. The highest unit of data used to be the petabyte (PB), but now we speak of the zettabyte (ZB), which is a trillion gigabytes (GB) or a billion terabytes (TB). As the volume of data increases, we are having a hard time keeping up with universally accepted naming for the next level. Table 8.1 provides an overview of the size and naming of modern data volumes (Sharda et al., 2014).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Table 8.1 Naming for the Increasing Volumes of Data

Unit Name	Symbol	Value (bytes)
Kilobyte	kB	10^3
Megabyte	MB	10^6
Gigabyte	GB	10^9
Terabyte	TB	10^{12}
Petabyte	PB	10^{15}
Exabyte	EB	10^{18}
Zettabyte	ZB	10^{21}
Yottabyte	YB	10^{24}
Brontobyte	BB	10^{27}
Gegobyte	GeB	10^{30}

Consider that an exabyte of data is created on the Internet each day, which equates to 250 million DVDs' worth of information. And the idea of even larger amounts of data—a zettabyte—isn't too far off when it comes to the amount of information traversing the Web in a year. In fact, academics and industry experts estimate that we will see more than 2 zettabytes of data traffic annually over the Internet by 2025—and soon enough, we might start talking about even bigger volumes. Some big data scientists allege that the NSA and FBI have a yottabyte of data on people. To put this measurement in perspective, a yottabyte is the amount of storage on 250 trillion DVDs. A brontobyte, which is not an official SI unit but is apparently recognized by some people in the measurement community, is a 1 followed by 27 zeros. Size of such magnitude can be used to describe the type of sensor data that we will get from the Internet of Things in the next decade—or sooner. A gegobyte is 10^{30}. With respect to where the big data comes from, consider the following (Higginbotham, 2012):

- The CERN Large Hadron Collider generates 1 petabyte of data per second.
- Sensors from a Boeing jet engine create 20 terabytes of data every hour.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

- Facebook databases take in 500 terabytes of new data per day.
- On YouTube, 72 hours of video are uploaded per minute, translating to 1 terabyte every four minutes.
- The proposed Square Kilometre Array telescope (the proposed biggest telescope in the world) will generate an exabyte of data per day.

Here is some recent historical perspective: In 2009, the world had about 0.8 ZB of data; in 2010, it exceeded the 1 ZB mark; at the end of 2011, the number was 1.8 ZB. IBM has estimated that six or seven years from now, we will have more than 100 ZB. This number is astonishing, and so are the challenges and opportunities that come with it.

Variety

Data today comes in a variety of formats—ranging from traditional databases to hierarchical data stores created by end users and OLAP systems; to text documents, email, and XML; to meter-collected, sensor-captured data; to video, audio, and stock ticker data. By some estimates, 80% to 85% of all organizations' data is in some sort of unstructured or semi-structured format (i.e., a format that is not suitable for traditional database schemas). But there is no denying its value, and hence it must be included in the analyses to support decision making.

Velocity

According to Gartner, a well-known and highly respected technology consultancy company, *velocity* refers to both how fast data is being produced and how fast the data must be processed (i.e., captured, stored, and analyzed) to meet the need or demand. RFID tags, automated sensors, GPS devices, and smart meters are driving an increasing need to deal with torrents of data in near real time. Velocity

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

is perhaps the most overlooked characteristic of big data. Reacting quickly enough to deal with velocity is a challenge for most organizations. For time-sensitive environments, the opportunity cost clock of the data starts ticking the moment the data is created. As time passes, the value proposition of the data degrades, and it eventually becomes worthless. Whether the subject matter is the health of a patient, the well-being of a traffic system, or the health of an investment portfolio, accessing the data and reacting faster to the circumstances will always create more advantageous outcomes.

In the big data storm that we are witnessing today, almost everyone is fixated on at-rest analytics, using optimized software and hardware systems to mine large quantities of variant data sources. Although this is critically important and highly valuable, another class of analytics that is often overlooked is driven by the velocity nature of big data; it is called *data stream analytics* or *in-motion analytics*. If done correctly, data stream analytics can be as valuable as—and in some business environments more valuable than—at-rest analytics. We cover this topic in more detail later in this chapter.

Veracity

IBM is using the term *veracity* as the fourth *V* to describe big data. It refers to conformity to facts—to the accuracy, quality, truthfulness, or trustworthiness of data. Tools and techniques are often used to handle big data's veracity by transforming the data into trustworthy insights.

Variability

Data flows can be highly inconsistent, with periodic peaks. Inconsistency in data flow makes it very difficult to properly and cost-effectively develop data infrastructures. If the resources are put in place to handle the peak times, for the rest of the time, they will be

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

significantly underutilized. One popular way to handle the problem of variability is to use pooled resources, based on the infrastructure-as-a-service business model. Cloud computing, service-oriented architecture, and massively parallel processing make variability a manageable issue for not only large but also small to medium-sized businesses.

Value Proposition

The excitement around big data is about its value proposition. A preconceived notion about big data is that it contains (or has a greater potential to contain) more patterns and interesting anomalies than "small" data. Thus, by analyzing large and feature-rich data, organizations can gain greater business value than they could otherwise gain. While users can detect the patterns in small data sets by using simple statistical and machine learning methods or ad hoc query and reporting tools, big data means "big" analytics. Big analytics means greater insight and better decision making—something that every organization needs.

Since the exact definition of big data is still a matter of ongoing discussion in academic and industrial circles, it is likely that more characteristics (perhaps more *V*s) will be added to this list. Regardless of what happens, the importance and value proposition of big data are here to stay.

Fundamental Concepts of Big Data

Big data by itself, regardless of the size, type, or speed of the data, is worthless unless business users are able to use it to deliver value to their organizations. That's where "big" analytics comes into the picture. Although organizations have for a long time run reports and dashboards against data warehouses, many have not opened these repositories to in-depth and on-demand explorations. This is partly because analysis tools have been too complex for the average user and

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

partly because the repositories often have not contained all the data that power users' needs. But this has begun to change in a dramatic fashion, thanks to the new big data analytics paradigm.

Along with the value proposition, big data has also brought about big challenges for organizations. The traditional means for capturing, storing, and analyzing data are not capable of dealing with big data effectively and efficiently. Therefore, new breeds of technologies need to be developed (or purchased, hired, or outsourced) to take on the big data challenge. Before making such an investment, organizations should justify the means. If any of the following statements are true, then you need to seriously consider embarking on a big data journey:

- You can't process the amount of data that you want to process because of the limitations posed by your current platform or environment.
- You want to involve new/contemporary data sources (e.g., social media, RFID, sensory, web, GPS, textual data) in your analytics platform, but you can't because it does not comply with the data storage schema-defined rows and columns without sacrificing fidelity or the richness of the new data.
- You need to (or want to) integrate data as quickly as possible to be current in your analysis.
- You want to work with a schema-on-demand (as opposed to the predetermined schema used in RDBMSs) data storage paradigm because the nature of the new data may not be known, or there may not be enough time to determine it and develop a schema for it.
- The data is arriving so fast at your organization's doorstep that your traditional analytics platform cannot handle it.

As is the case with any other large IT investment, the success in big data analytics depends on a number of factors. Figure 8.2 illustrates the most critical success factors (Watson, 2012).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.



Figure 8.2 Critical Success Factors for Big Data Analytics

The following are among the most critical of these success factors for big data analytics:

- **A clear business need (alignment with the vision and the strategy).** Business investments ought to be made for the good of the business, not for the sake of mere technology advancements. Therefore, the main driver for big data analytics should be the needs of the business, at any level: strategic, tactical, or operational.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

- **Strong, committed sponsorship (i.e., executive champions).** It is a well-known fact that if you don't have strong, committed executive sponsorship, it is difficult (or even impossible) to succeed. If the scope is a single or a few analytical applications, the sponsorship can be at the department level. However, if the target is enterprise-wide organizational transformation, which is often the case for big data initiatives, sponsorship needs to be at the highest levels, and it needs to be organization-wide.
- **Alignment between the business and IT strategy.** It is essential to make sure that analytics work is always supporting the business strategy—and not the other way around. Analytics should play an enabling role in execution of the business strategy.
- **A fact-based decision-making culture.** In a fact-based decision-making culture, the numbers—rather than intuition, gut feeling, or supposition—drive decision making. There is also a culture of experimentation to see what works and what doesn't. To create a fact-based decision-making culture, the senior management needs to
  - Recognize that some people can't or won't adjust
  - Be a vocal supporter
  - Stress that outdated methods must be discontinued
  - Ask to see what analytics went into decisions
  - Link incentives and compensation to desired behaviors
- **A strong data infrastructure.** Data warehouses have provided the data infrastructure for analytics. This infrastructure is changing and being enhanced in the big data era with new technologies. Success requires marrying the old with the new for a holistic infrastructure that works synergistically.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

As the size and complexity of data increase, the need for more efficient analytical systems is also increasing. In order to keep up with the computational needs of big data, a number of new and innovative computational techniques and platforms are being developed. These techniques, collectively called *high-performance computing*, include the following:

- *In-memory analytics* solves complex problems in near real time with highly accurate insights by allowing analytical computations and big data to be processed in-memory and distributed across a dedicated set of nodes.
- *In-database analytics* speeds insights and enables better data governance by performing data integration and analytic functions inside the database so you don't have to move or convert data repeatedly.
- *Grid computing* promotes efficiency, lower cost, and better performance by processing jobs in a shared, centrally managed pool of IT resources.
- *Appliances* bring together hardware and software in a physical unit that is not only fast but also can be scaled on an as-needed basis.

Computational requirement is just a small part of the list of challenges that big data imposes on today's enterprises. Following is a list of challenges that business executives have found to have significant impacts on successful implementation of big data analytics. When considering big data projects and architecture, being mindful of these challenges will make the journey to analytics competency a less stressful one:

- **Data volume.** It's important to be able to capture, store, and process the huge volume of data at an acceptable speed so that the latest information is available to decision makers when they need it.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

- **Data integration.** It's important to be able to combine data that is not similar in structure or source and to do so quickly and at reasonable cost.
- **Processing capabilities.** It's important to be able to process data quickly, as it is captured. Traditional ways of collecting and then processing data may not work. In many situations, data needs to be analyzed as soon as it is captured to leverage the most value. (This process, called *stream analytics*, is covered later in this chapter.)
- **Data governance.** It's important to be able to keep up with the security, privacy, ownership, and quality issues of big data. As the volume, variety (format and source), and velocity of data change, so should the capabilities of governance practices.
- **Skills availability.** Big data is being harnessed with new tools and is being looked at in different ways. There is a shortage of people (often called data scientists, discussed later in this chapter) who have the skills to do the job.
- **Solution cost.** Since big data has opened up a world of possible business improvements, there is a great deal of experimentation and discovery taking place to determine the patterns that matter and the insights that turn to value. To ensure a positive ROI on a big data project, therefore, it is crucial to reduce the cost of the solutions used to find that value.

The challenges are real, but so is the value proposition of big data analytics. Anything that business analytics leaders can do to help prove the value of new data sources to the business will move the organization beyond experimenting and exploring big data to adapting and embracing it as a differentiator. There is nothing wrong with exploration, but ultimately the value of big data comes from putting insights into action.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The Business Problems That Big Data Analytics Addresses

The top business problems addressed by big data overall are process efficiency, cost reduction, and enhanced customer experience. However, different priorities emerge in different industries. Process efficiency and cost reduction are common business problems that can be addressed by analyzing big data, and these are among the top-ranked problems that can be addressed with big data analytics for sectors such as manufacturing, government, energy and utilities, communications and media, transport, and health care. Enhanced customer experience may be at the top of the list of problems addressed by insurance companies and retailers. Risk management usually is at the top of the list for companies in banking and education.

The following are some of the top problems that can be addressed by using big data analytics:

- Process efficiency and cost reduction
- Brand management
- Revenue maximization, cross-selling, and upselling
- Enhanced customer experience
- Churn identification and customer recruiting
- Improved customer service
- Identification of new products and market opportunities
- Risk management
- Regulatory compliance
- Enhanced security capabilities

Big Data Technologies

A number of technologies are involved in processing and analyzing big data, and most of them have some common characteristics.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Primarily, they take advantage of commodity hardware to enable scale-out and parallel-processing techniques; they employ nonrelational data storage capabilities in order to process unstructured and semi-structured data; and they apply advanced analytics and data visualization technology to big data to convey insights to end users.

The following sections examine three big data technologies that stand out as being likely to transform the business analytics and data management markets: MapReduce, Hadoop, and NoSQL.

MapReduce

MapReduce is a technology popularized by Google that distributes the processing of very large multistructured data files across a large cluster of machines. High performance is achieved by breaking the processing into small units of work that can be run in parallel across the hundreds—and potentially thousands—of nodes in the cluster. In their seminal paper on MapReduce, Dean and Ghemawat (2004) said:

> MapReduce is a programming model and an associated implementation for processing and generating large data sets. Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system.

The key point to note from this quote is that MapReduce is a programming model, not a programming language. That is, it is designed to be used by programmers rather than by business users.

To see how MapReduce works, let's look at an example. Figure 8.3 shows an example in which the input to the MapReduce process is a set of colored squares. The objective is to count the number of squares of each color. The programmer in this example is

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

responsible for coding the map and reduce programs; the remainder of the processing is handled by the software system implementing the MapReduce programming model.

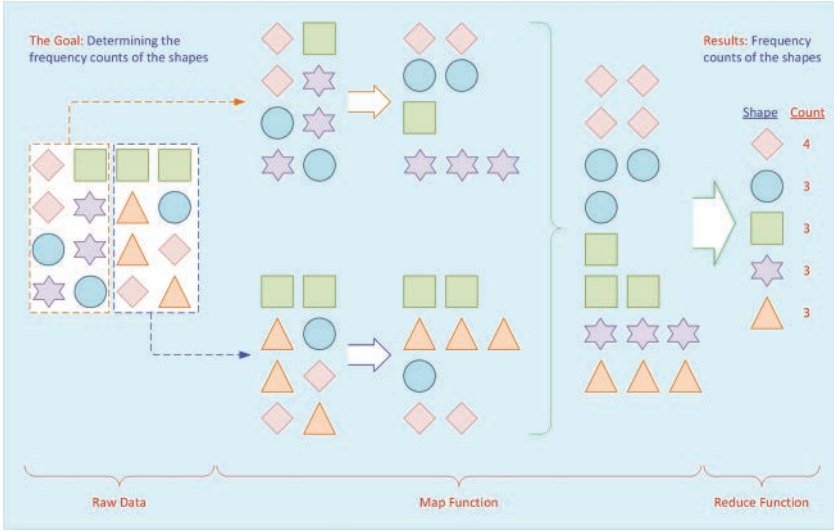


Figure 8.3 A Graphical Depiction of the MapReduce Process

The MapReduce system first reads the input file and splits it into multiple pieces. In this example, there are two splits, but in a real-life scenario, the number of splits would typically be much higher. These splits are then processed by multiple map programs running in parallel on the nodes of the cluster. The role of each map program in this case is to group the data in a split by color. The MapReduce system then takes the output from each map program and merges (i.e., shuffles and sorts) the results for input to the reduce program, which calculates the sum of the number of squares of each color. In this example, only one copy of the reduce program is used, but there may be more in practice. To optimize performance, programmers can provide their own shuffle and sort program, and they can also deploy a combiner that combines local map output files to reduce the number of output files that have to be remotely accessed across the cluster by the shuffling and sorting step.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Why Use MapReduce?

MapReduce aids organizations in processing and analyzing large volumes of multistructured data. Application examples include indexing and search, graph analysis, text analysis, machine learning, and data transformation. These types of applications are often difficult to implement using the standard SQL employed by relational DBMSs.

The procedural nature of MapReduce makes it easy for skilled programmers to understand. It also has the advantage of not requiring developers to be concerned with implementing parallel computing; the system transparently handles that.

Although MapReduce is designed for programmers, nonprogrammers can exploit the value of prebuilt MapReduce applications and function libraries. Both commercial and open source MapReduce libraries are available that provide a wide range of analytic capabilities. Apache Mahout, for example, is an open source machine learning library of algorithms for clustering, classification, and batch-based collaborative filtering that are implemented using MapReduce.

Hadoop



Hadoop is an open source framework for processing, storing, and analyzing massive amounts of distributed, unstructured data. Originally created by Doug Cutting at Yahoo!, Hadoop was inspired by MapReduce, a user-defined function developed by Google in the early 2000s for indexing the Web. Hadoop was designed to handle petabytes and exabytes of data, distributed over multiple nodes in parallel.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Hadoop clusters run on inexpensive commodity hardware, so projects can scale out without breaking the bank. Hadoop is now a project of the Apache Software Foundation, where hundreds of contributors continuously improve the core technology.

How Hadoop Works

Fundamentally, rather than bang away at one huge block of data with a single machine, Hadoop breaks up big data into multiple parts so all the parts can be processed and analyzed at the same time. A client accesses unstructured and semi-structured data from sources including log files, social media feeds, and internal data stores. It breaks up the data into parts, which are then loaded into a file system made up of multiple nodes running on commodity hardware. The default file store in Hadoop is the Hadoop Distributed File System (HDFS). File systems such as HDFS are adept at storing large volumes of unstructured and semi-structured data because they do not require data to be organized into relational rows and columns. Each part is replicated multiple times and loaded into the file system so that if a node fails, another node has a copy of the data contained on the failed node. A name node acts as facilitator, communicating back to the client information such as which nodes are available, where in the cluster certain data resides, and which nodes have failed.

After the data is loaded into a cluster, it is ready to be analyzed via the MapReduce framework. The client submits a map job—usually a query written in Java—to one of the nodes in the cluster, known as the job tracker. The job tracker refers to the name node to determine which data it needs to access to complete the job and where in the cluster that data is located. Once this is determined, the job tracker submits the query to the relevant nodes. Rather than bring all the data back into a central location for processing, processing occurs at each node simultaneously, or in parallel. This is an essential characteristic of Hadoop.

When each node has finished processing its given job, it stores the results. The client initiates a reduce job through the job tracker,

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

and the results of the map phase stored locally on individual nodes are aggregated to determine the "answer" to the original query and are then loaded onto another node in the cluster. The client accesses these results, which can then be loaded into one of a number of analytic environments for analysis. The MapReduce phase is complete at this point.

After the MapReduce phase, the processed data is ready for further analysis by data scientists and others with advanced data analytics skills. Data scientists can manipulate and analyze the data, using a number of tools for a number of uses, including to search for hidden insights and patterns or to use as the foundation to build user-facing analytic applications. The data can also be modeled and transferred from Hadoop clusters into existing relational databases, data warehouses, and other traditional IT systems for further analysis and/or to support transactional processing.

Hadoop Technical Components

A Hadoop "stack" is made up of a number of components, including the following:

- **Hadoop Distributed File System (HDFS).** This is the default storage layer in any given Hadoop cluster.
- **Name node.** This is the node in a Hadoop cluster that provides the client information on where in the cluster particular data is stored and whether any nodes fail.
- **Secondary node.** A backup to the name node, the secondary node periodically replicates and stores data from the name node in case it fails.
- **Job tracker.** This is the node in a Hadoop cluster that initiates and coordinates MapReduce jobs or the processing of the data.
- **Slave nodes.** The grunts of any Hadoop cluster, slave nodes store data and take direction to process it from the job tracker.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

In addition, the Hadoop ecosystem is made up of a number of complementary subprojects. NoSQL data stores such as Cassandra and HBase are used to store the results of MapReduce jobs in Hadoop. In addition to Java, some MapReduce jobs and other Hadoop functions are written in Pig, an open source language designed specifically for Hadoop. Hive is an open source data warehouse originally developed by Facebook that allows for analytic modeling in Hadoop. A rich set of Hadoop-related projects and supporting tools/platforms can be found in Sharda et al. (2014).

Hadoop Pros and Cons

The main benefit of Hadoop is that it allows enterprises to process and analyze large volumes of (previously inaccessible) unstructured and semi-structured data in a cost- and time-effective manner. Because Hadoop clusters can scale to petabytes and even exabytes of data, enterprises no longer need to rely on sample data sets but can process and analyze *all* relevant data. Data scientists can apply an iterative approach to analysis, continually refining and testing queries to uncover previously unknown insights. Also, getting started with Hadoop doesn't cost much. Developers can download the Apache Hadoop distribution for free and begin experimenting with Hadoop in less than a day.

The downside to Hadoop and its myriad components is that they are immature and still developing. As with any other young, raw technology, implementing and managing Hadoop clusters and performing advanced analytics on large volumes of unstructured data require significant expertise, skill, and training. Unfortunately, there is currently a dearth of Hadoop developers and data scientists available, which means it's impractical for many enterprises to maintain and take advantage of complex Hadoop clusters. Further, as Hadoop's myriad components are improved by the community and as new components are created, there is a risk of forking (as there is with any immature open source technology/approach). Finally, Hadoop is a batch-oriented framework, which means it does not support real-time data processing and analysis.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The good news is that some of the brightest minds in IT are contributing to the Apache Hadoop project, and a new generation of Hadoop developers and data scientists is coming of age. As a result, the technology is advancing rapidly, becoming both more powerful and easier to implement and manage. An ecosystem of vendors—both Hadoop-focused startups like Cloudera and well-worn IT stalwarts like IBM and Microsoft—are working to offer commercial, enterprise-ready Hadoop distributions, tools, and services to make deploying and managing the technology a practical reality for traditional enterprises. Other bleeding-edge startups are working to perfect NoSQL (which stands for Not Only SQL and is discussed later in this chapter) data stores capable of delivering near-real-time insights in conjunction with Hadoop.

A Few Demystifying Facts About Hadoop

Although Hadoop and related technologies, such as MapReduce and Hive, have been around for over a decade now, many people still have some misconceptions about them. The following list of 10 facts is intended to clarify what Hadoop is and does relative to business intelligence (BI), as well as in which business and technology situations Hadoop-based BI, data warehousing, and analytics can be useful (Russom, 2013):

- **Fact 1: Hadoop is not a single product.** People new to big data usually think that Hadoop is the key product that makes things happen in the new age of data science. In actuality, Hadoop is not a single product; it is an ecosystem. It consists of several open-source products (developed under the umbrella of Apache Hadoop Foundation) working like an engine behind the scenes to convert big data into invaluable knowledge needed to make smarter and faster decisions. The library of products from Apache Hadoop Foundation includes MapReduce, the Hadoop Distributed File System (HDFS), Hive, HBase, Pig, Sqoop, Oozie, Hue, Zookeeper, and Flume, among others. These

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

products can be combined in a variety of configurations specific to a given business analytics problem and related data sources.

- **Fact 2: Hadoop is not just from Apache; it is a community-based ecosystem.** The Hadoop library of solutions includes several products from Apache Foundation, but it also includes many products from a long list of vendors functioning in the big data arena. As the popularity of Hadoop evolves, more and more communities and vendors are joining the movement to make it as comprehensive and versatile as possible.
- **Fact 3: Hadoop is an open source community project.** Hadoop is available free to anyone as an open source software library. It can be retrieved from the Apache website, www.apache.org. Several startup companies provide packaged solutions to companies that build on the Hadoop library and extend its features to the specific needs and wants of the clients.
- **Fact 4: Hadoop and MapReduce are two complementary products.** Google developed MapReduce before the Hadoop distributed file system was invented. Therefore, the early versions of MapReduce did not rely on HDFS but instead worked (and can still work) with other non-HDFS data storage technologies, including some of the most common DBMSs.
- **Fact 5: Hadoop HDFS is a file management system, not a relational database management system.** Hadoop primarily deals with files (not tables and records) in a distributed environment. Therefore, its level of data granularity is at the file level, and hence, it does not have some of the common functions associated with DBMSs, including SQL queries, relational data tables, intentional indexing for fast retrieval, rapid access to indexed data, and so on. In return, however, HDFS is capable of doing things at the file level that a DBMS cannot do.
- **Fact 6: Hive looks like SQL but is not standard SQL.** Hive is a variant of standard SQL that is used to manipulate data in DBMSs. To a data analyst familiar with SQL, learning to code in Hive is a relatively quick and straightforward process. Many

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

in the field of big data believe and hope that with each iteration Hive will get closer to the syntax of SQL, and at some point in the near future, standard SQL will be readily used to handle data in the Hadoop family of products.

- **Fact 7: Hadoop is not a replacement for a data warehouse.** Historically, data warehouses have been designed to handle an enterprise's structured, often relational, data. With the advent of big data, data warehouses have been criticized for their inability to handle unstructured data. The Hadoop family of products aims to complement (not replace, at least not yet) data warehouses by handling unstructured data types that data warehouses cannot handle.
- **Fact 8: Hadoop empowers analytics.** Although Hadoop has been used widely and polarized by Internet-based companies, it can enable any type of analytics, not just web analytics (e.g., analyzing web logs and other Internet-based data). For example, one area where Hadoop makes a strong case is in the analysis of IoT data that is mainly generated by machines and sensors in industries such as transportation, energy, retail, manufacturing (e.g., predictive maintenance), telecommunications, and network security.
- **Fact 9: MapReduce is not just analytics.** Although there is a close relationship between MapReduce and analytics, the relationship is not exclusive. While MapReduce is a general-purpose execution engine (which is capable of handling a variety of complex tasks that involve parallel programming, network communication, and fault tolerance), it is not specific to only analytics applications; instead, it can be used for any kind of computational task.
- **Fact 10: Hadoop is not just about volume but also diversity of data.** HDFS is designed to handle not only very large and distributed files but also files of different types. The process is rather simple; any type and size of data can be stored (centrally or distributed) and managed using a rather trivial and straightforward process built in the Hadoop HDFS.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

NoSQL

A related new style of database called NoSQL (which stands for Not Only SQL) has emerged to, like Hadoop, process large volumes of multistructured data. However, whereas Hadoop is adept at supporting large-scale, batch-style historical analysis, NoSQL databases are aimed, for the most part (though there are some important exceptions), at serving up discrete data stored among large volumes of multistructured data to end-user and automated big data applications. This capability is sorely lacking from relational database technology, which simply can't maintain needed application performance levels at a big data scale.

In some cases, NoSQL and Hadoop work in conjunction. HBase, for example, is a popular NoSQL database modeled after Google BigTable that is often deployed on top of HDFS to provide low-latency, quick lookups in Hadoop. The downside of most NoSQL databases today is that they have traded ACID (atomicity, consistency, isolation, durability) compliance for performance and scalability. Many also lack mature management and monitoring tools. Both of these shortcomings are in the process of being overcome by both the open source NoSQL communities and a handful of vendors that are attempting to commercialize the various NoSQL databases. NoSQL databases currently available include HBase, Cassandra, MongoDB, Accumulo, Riak, CouchDB, and DynamoDB, among others.

Data Scientists

Data scientist is a role or a job frequently associated with big data or data science. In a very short time, it has become one of the most sought-out roles in the marketplace. In an article published in a 2012 issue of *Harvard Business Review*, Thomas H. Davenport and D. J. Patil called data scientist "the sexiest job of the 21st century." In their article, they specified data scientists' most basic, universal skill as the

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

ability to write code (in the latest big data languages and platforms). Although this may be less true in the near future, when many more people will have the title "data scientist" on their business cards, at this time it seems to be the most fundamental skill required of data scientists. A more enduring skill will be the need for data scientists to communicate in language that all their stakeholders understand—and to demonstrate the special skills involved in storytelling with data, whether verbally, visually, or—ideally—both (Davenport & Patil, 2012).

Data scientists use a combination of business and technical skills to *investigate* big data, looking for ways to improve current business analytics practices (from descriptive to predictive and prescriptive) and hence to improve decisions for new business opportunities. One of the biggest differences between a data scientist and a BI user—such as a business analyst—is that a data scientist investigates and looks for new possibilities, whereas a BI user analyzes existing business situations and operations.

One of the dominant traits expected of data scientists is an intense curiosity—a desire to go beneath the surface of a problem, find the questions at its heart, and distill them into a very clear set of hypotheses that can be tested. This often entails the associative thinking that characterizes the most creative scientists in any field. For example, one data scientist studying a fraud problem realized that it was analogous to a type of DNA sequencing problem (Davenport & Patil, 2012). By bringing together those disparate worlds, he and his team were able to craft a solution that dramatically reduced fraud losses.

Where Do Data Scientists Come From?

Although there is still disagreement about the use of *science* in the name, data science is becoming a less controversial issue. Traditionally, scientists use tools made by other scientists, or they make them if they don't exist, as a means to expand knowledge. That is also

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

what data scientists do. Experimental physicists, for example, have to design equipment, gather data, conduct multiple experiments to discover knowledge, and communicate their results. Even though they may not be wearing white coats and may not be living in a sterile lab environment, data scientists have a similar role to experimental physicists in that they use creative tools and techniques to turn data into actionable information for others to use for better decision making.

There is no consensus on what educational background a data scientist has to have. The usual suspects, like MS or PhD in computer science, MIS, industrial engineering, or, recently, analytics, may be necessary, but none of them are sufficient to call someone a data scientist. One of the most sought-out characteristics of a data scientist is expertise in both technical and business application domains. In that sense, it somewhat resembles the professional engineer (PE) or project management professional (PMP) roles, where experience is valued as much as (if not more so than) technical skills and educational background. It would not be surprising to see widespread certification programs specifically designed for data scientists in the next few years.

The field of data science is still being defined, and many of its practices are still experimental and far from being standardized. Companies are therefore quite sensitive about the experience dimension of data scientists. As the profession matures and practices are standardized, experience will be less of an issue in the definition of a data scientist. Today, companies are looking for people who have extensive experience working with complex data and have had good luck recruiting those with educational and work backgrounds in the physical or social sciences. Some of the best and brightest data scientists have been PhDs in esoteric fields like ecology and systems biology (Davenport & Patil, 2012). Even though there is no consensus on where data scientists come from, there is a common understanding of what skills and qualities they are expected to possess. Figure 8.4 shows a high-level graphical illustration of skills that a data scientist needs.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

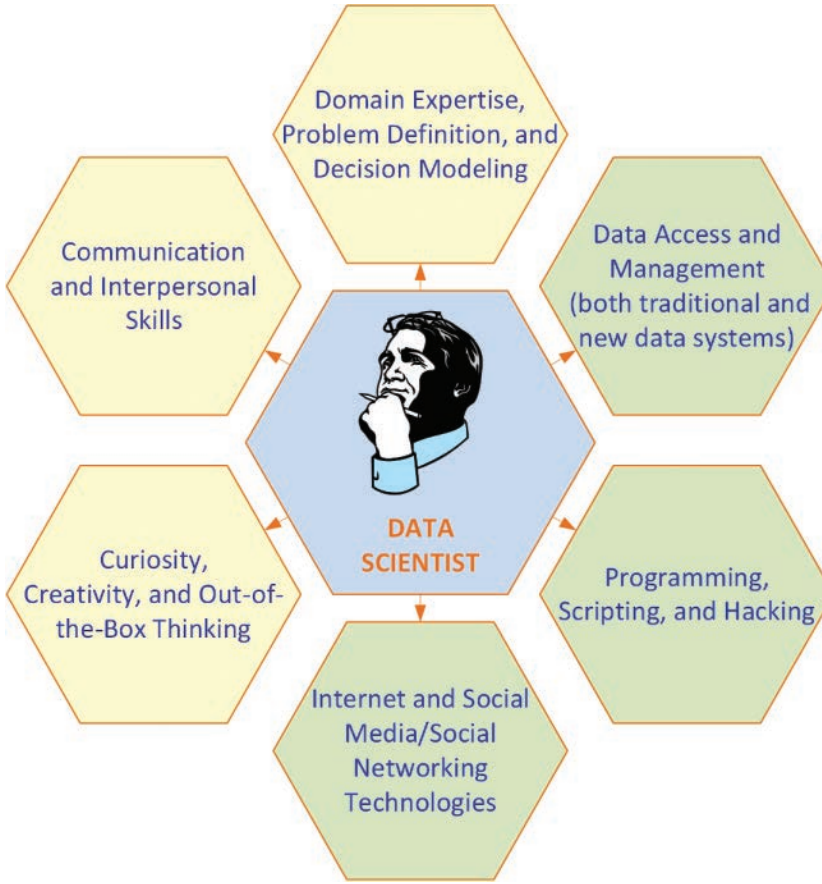


Figure 8.4 Skills That a Data Scientist Needs

Data scientists need to have soft skills such as creativity, curiosity, communication/interpersonal, domain expertise, problem definition, and managerial skills (shown on the left side of Figure 8.4). They also need sound technical skills such as data manipulation, programming/hacking/scripting, and Internet and social media/networking technologies (shown on the right side of the figure).

Big Data and Stream Analytics

Along with volume and variety, as discussed earlier in this chapter, one of the key characteristics defining big data is velocity, which

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

refers to the speed at which data is created and streamed into the analytics environment. Organizations are looking for new means to process streaming data as it comes in to react quickly and accurately to problems and opportunities to please customers and gain competitive advantage. In situations where data streams in rapidly and continuously, traditional analytics approaches that work with previously accumulated data (i.e., data at rest) often either arrive at the wrong decisions because they use too much out-of-context data, or they arrive at the correct decisions but too late for the decisions to be of any use to an organization. Therefore, it is critical for a number of business situations to analyze data soon after it is created and/or as soon as it streamed into the analytics system.

A presumption that the vast majority of modern businesses are living by today is that it is important to record every piece of data because it might contain valuable information now or sometime in the near future. However, as the number of data sources increases, the store-everything approach becomes harder and harder to do and, in some cases, is even infeasible. In fact, despite the technological advances, current total storage capacity lags far behind the digital information being generated in the world. Moreover, in the constantly changing business environment, real-time detection of meaningful changes in data as well as of complex pattern variations within a given short time window are essential in order to come up with actions that better fit with the new environment. These facts become the main triggers for a paradigm called *stream analytics*. The stream analytics paradigm came about as an answer to challenges such as the unbounded flows of data that cannot be permanently stored in order to be subsequently analyzed, in a timely and efficient manner, and complex pattern variations that need to be detected and acted upon as soon as they happen.

Stream analytics (also called *data-in-motion analytics* and *real-time data analytics*) commonly refers to the analytic process of extracting actionable information from continuously streaming data. A *stream* can be defined as a continuous sequence of data elements

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

(Zikopoulos et al., 2013). The data elements in a stream are often called *tuples*. In a relational database sense, a tuple is similar to a row of data (i.e., a record, an object, or an instance). However in the context of semi-structured or unstructured data, a tuple is an abstraction that represents a package of data, which can be characterized as a set of attributes for a given object. If a tuple by itself is not sufficiently informative for analysis, a correlation or another collective relationship among tuples is needed, and then a window of data that includes a set of tuples is used. A window of data is a finite number/sequence of tuples, and the window is continuously updated as new data becomes available. The size of a window is determined based on the system being analyzed. Stream analytics is becoming increasingly popular because of two things: First, time-to-action has become a decreasing value, and second, we have the technological means to capture and process the data while it is being created.

Some of the most important applications of stream analytics have been developed in the energy industry, specifically for smart grid (electric power supply chain) systems. The new smart grids are capable of real-time creation and processing of multiple streams of data in order to determine optimal power distribution to fulfill real customer needs; they can also generate accurate short-term predictions aimed at covering unexpected demand and renewable energy generation peaks.

Figure 8.5 illustrates a generic use case for streaming analytics in the energy industry (a typical smart grid application). The goal is to accurately predict electricity demand and production in real time by using streaming data that is coming from smart meters, production system sensors, and meteorological models. The ability to predict near-future consumption and production trends and detect anomalies in real time can be used to optimize supply decisions (e.g., how much to produce, what sources of production to use, how to optimally adjust production capacities) as well as to adjust smart meters to regulate consumption and favorable energy pricing.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

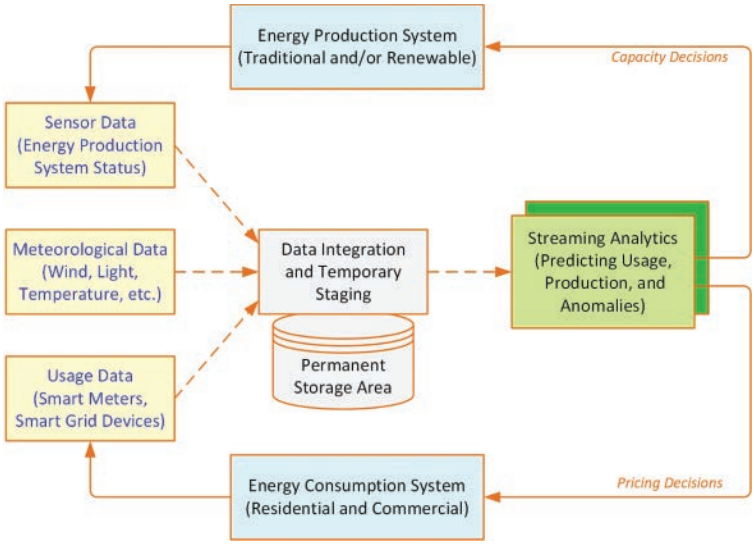


Figure 8.5 A Use Case of Streaming Analytics in the Energy Industry

Data Stream Mining

Data stream mining, as an enabling technology for stream analytics, is the process of extracting novel patterns and knowledge structures from continuous, rapid data records. As we have seen in this book, traditional data mining methods require data to be collected and organized in a proper file format and then processed in a recursive manner to learn the underlying patterns. In contrast, a data stream is a continuous flow of an ordered sequence of instances that in many applications of data stream mining can be read/processed only once or a small number of times, using limited computing and storage capabilities. Examples of data streams include sensor data, computer network traffic, phone conversations, ATM transactions, web searches, and financial data. Data stream mining can be considered a subfield of data mining, machine learning, and knowledge discovery.

In many data stream mining applications, the goal is to predict the class or value of new instances in the data stream, given some knowledge about the class membership or values of previous instances in the data stream. Specialized machine learning techniques (mostly derivative of traditional machine learning techniques) can be used

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

to learn this prediction task from labeled examples in an automated fashion. An example of such a prediction method has been developed by Delen et al. (2005), who gradually built and refined a decision tree model by using one subset of the data at a time.

Application Example: Big Data for Political Campaigns

One of the application areas where big data and analytics promise to make a big difference is the field of politics. Experiences from recent presidential elections have illustrated the power of big data and analytics to acquire and energize millions of volunteers (in the form of a modern-era grassroots movement) to not only raise hundreds of millions of dollars for the election campaign but to optimally organize and mobilize potential voters to get out and vote in large numbers. Clearly, starting with the 2008 U.S. presidential elections, creative use of big data and analytics have made a mark on the political arena with the intent to improve the chances of winning the elections. Figure 8.6 illustrates the analytical process of converting a wide variety of data to the ingredients for winning an election.

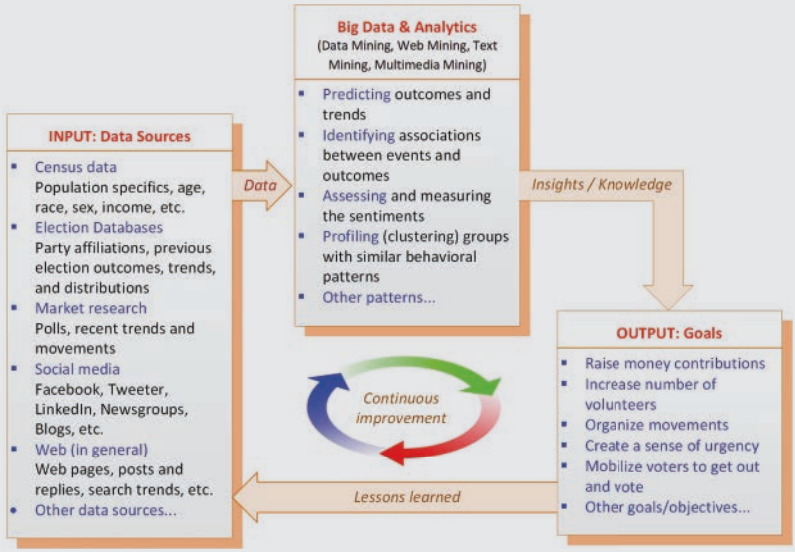


Figure 8.6 Leveraging Big Data and Analytics for Political Campaigns

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

As Figure 8.6 illustrates, data is the source of information; the richer and deeper it is, the better and more relevant the insights. The main characteristics of big data—volume, variety, and velocity (the three *V*s)—readily apply to the kind of data that is used for political campaigns. In addition to the structured data (e.g., detailed records of previous campaigns, census data, market research and poll data), vast volumes and verity of social media (e.g., tweets at Twitter, Facebook wall posts, blog posts) and web data (e.g., web pages, news articles, newsgroups) are used to learn more about voters and obtain deeper insights to enforce or change them. Often, the search and browsing histories of individuals are captured and made available to customers (political analysts), who can use such data for better insight and behavioral targeting. If handled correctly, big data and analytics can provide invaluable information to manage political campaigns better than ever before.

From predicting the election outcomes to targeting potential voters and donors, big data and analytics have a lot to offer to modern election campaigns. In fact, it has changed the way presidential election campaigns are run. In the 2008 and 2012 presidential elections, the major political parties (Republican and Democrat) employed social media and data-driven analytics for more effective and efficient campaigns, but as many agree, the Democrats clearly had the competitive advantage (Issenberg, 2012). Obama's 2012 data- and analytics-driven operation was far more sophisticated and more efficient than its much-heralded 2008 process, which was primarily social media driven. In the 2012 campaign, hundreds of analysts applied advanced analytics on very large and diverse data sources to pinpoint exactly who to target, for what reason, and with what message—on a continuous basis. Compared to 2008, they had more expertise, hardware, software, data (e.g., Facebook

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

and Twitter were orders of magnitude bigger in 2012 than they were in 2008), and computational resources to go over and beyond what they had accomplished previously (Shen, 2013). Before the 2012 election, a *Politico* reporter claimed that Obama had a data advantage and went on to say that the depth and breadth of the campaign's digital operation, from political and demographic data mining to voter sentiment and behavioral analysis, reached beyond anything politics had ever seen (Romano, 2012).

According to Shen (2013), the real winner of the 2012 election was analytics. While most people, including so-called political experts (who often rely on gut feelings and experiences), thought the 2012 presidential election would be very close, a number of analysts, based on their data-driven analytical models, predicted that Obama would win easily, with close to 99% certainty. For example, Nate Silver at FiveThirtyEight, a popular political blog published by *The New York Times*, predicted not only that Obama would win but by exactly how much. Simon Jackman, professor of political science at Stanford University, accurately predicted that Obama would win 332 electoral votes and that North Carolina and Indiana would be the only two states that Obama won in 2008 that would fall to Romney. Along with these successes, we have also witnessed some failures of big data analytics in predicting election outcomes. Despite the overwhelming prediction of almost every analytics expert and organization having Hillary Clinton win the 2016 U.S. presidential election, Donald Trump was the winner.

In short, big data and analytics have become a critical part of political campaigns. The use and expertise gap between the party lines may disappear, but the importance of analytical capabilities will continue to evolve in the foreseeable future.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Summary

We are living in a big data world. Some say that data is the new oil. The value of data—especially big data—is undeniable. As explained in this chapter, big data is big in volume, and it is also rich in data types (including textual, imagery, video, audio, and machine-generated IoT data), and high velocity in the way it is created and made available. This chapter discusses these big data characteristics and the opportunities and challenges they create for data scientists. The chapter also describes big data–enabling technologies, including Hadoop, MapReduce, and NoSQL, and explains the characteristics, both technical and non-technical, that collectively define the data scientist role.

References

Davenport, T. H., & D. J. Patil. (2012, October). "Data Scientist," *Harvard Business Review*, pp. 70–76.

Dean, J., & S. Ghemawat. (2004). *MapReduce: Simplified Data Processing on Large Clusters*. http://research.google.com/archive/mapreduce.html (accessed September 2020).

Delen, D., M. Kletke, & J. Kim. (2005). "A Scalable Classification Algorithm for Very Large Datasets," *Journal of Information and Knowledge Management*, 4(2): 83–94.

Higginbotham, S. (2012). *As Data Gets Bigger, What Comes After a Yottabyte?* http://gigaom.com/2012/10/30/as-data-gets-bigger-what-comes-after-a-yottabyte (accessed September 2020).

Issenberg, S. (2012, October 29). "Obama Does It Better," *Slate*. https://slate.com/news-and-politics/2012/10/obamas-secret-weapon-democrats-have-a-massive-advantage-in-targeting-and-persuading-voters.html (accessed September 2020).

Kelly, L. (2012). *Big Data: Hadoop, Business Analytics and Beyond*. wikibon.org/wiki/v/Big_Data:_Hadoop,_Business_Analytics_and_Beyond (accessed September 2020).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Romano, L. (2012, June 9). "Obama's Data Advantage," *Politico*. https://www.politico.com/story/2012/06/obamas-data-advantage-077213 (accessed September 2020).

Russom, P. (2013). "Busting 10 Myths About Hadoop," *Best of Business Intelligence*, 10: 45–46.

Samuelson, D. A. (2013, February). "Analytics: Key to Obama's Victory," *OR/MS Today*, pp. 20–24.

Scherer, M. (2012, November 7). "Inside the Secret World of the Data Crunchers Who Helped Obama Win," *Time*.

Sharda, R., D. Delen, & E. Turban. (2014). *Business Intelligence and Analytics: Systems for Decision Support*. Prentice Hall.

Shen, G. (2013, January–February). "Big Data, Analytics and Elections," *INFORMS Analytics Magazine*.

Watson, H. (2012). "The Requirements for Being an Analytics-Based Organization," *Business Intelligence Journal*, 17(2): 42–44.

Watson, H., R. Sharda, & D. Schrader. (2012). "Big Data and How to Teach It," Workshop at AMCIS 2012, Seattle.

Zikopoulos, P., D. deRoos, K. Parasuraman, T. Deutsch, J. Giles, & D. Corrigan. (2013). *Harness the Power of Big Data*. McGraw-Hill.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

This page intentionally left blank

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

9

Deep Learning and Cognitive Computing

As you will see in this chapter, artificial intelligence (AI) in general and the methods of machine learning in specific terms are evolving and advancing rapidly. The use of large digitized data sources (both structured and unstructured) from both inside and outside the organization, along with advanced computing systems (software and hardware combinations), has paved the way toward developing smart solutions to complex problems that were thought to be unsolvable just a few years ago.

Deep learning and cognitive computing—which are at the cutting edge in AI systems—are helping enterprises to make accurate and timely decisions by harnessing the rapidly expanding big data resources. As witnessed in academia and in industry, the new generation of AI systems is capable of solving problems in a significantly different manner, producing results that are much better than the results of their older counterparts. In the domain of fraud detection, for instance, traditional methods have always been marginally useful, having higher-than-desired false positive rates and causing unnecessary investigations and thereby dissatisfaction for their customers. As the good-old difficult problems, such as fraud detection, are resurfacing and being reimagined and redesigned, AI technologies such as deep learning are making them solvable with a high level of accuracy and applicability.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Introduction to Deep Learning

About a decade ago, conversing with an electronic device (in human language, intelligently) would have been inconceivable beyond sci-fi movies. Today, however, thanks to advances in AI methods and technologies, almost everyone has experienced this unthinkable phenomenon. You have probably at some point been driving and asked Siri or Google Assistant to dial a number from your phone address book or to find an address and give you specific directions. You might also have asked a virtual assistant to play music on your Google Home or Amazon Alexa device or on your TV. You might have found Facebook making eerily correct tagging suggestions for the faces of your friends in group photos you've uploaded. Maybe you haven't tried to translate a manuscript from a foreign language lately, but if you try it, you will find that it is as easy as taking a picture of the manuscript in the Google Translate mobile app and giving it a fraction of a second. These are only a few of the many ever-increasing applications of deep learning that have promised to make life easier for people.

Deep learning, as the newest and perhaps at this moment the most popular member of the AI and machine learning family, has a goal similar to the goals of the other machine learning methods that came before it: Mimic the thought process of humans, using mathematical algorithms to learn from data pretty much the same way that humans learn. So, what is really different (and advanced) in deep learning? As its name suggests, deep learning goes deeper than traditional machine learning. The performance of traditional machine learning algorithms such as decision trees, support vector machines, logistic regression, and neural networks relies heavily on the representation of the data. That is, only if we (analytics professionals or data scientists) provide those traditional machine learning algorithms with relevant and sufficient pieces of information (a.k.a. features) in proper format are they able to "learn" the patterns and thereby perform their prediction (classification or estimation), clustering, or

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

association tasks with an acceptable level of accuracy. In other words, these algorithms need humans to manually identify and derive features that are theoretically and/or logically relevant to the objectives of the problem at hand and feed these features into the algorithm in a proper format. For example, in order to use a decision tree to predict whether a given customer will return (or churn), the marketing manager needs to provide the algorithm with information such as the customer's socioeconomic characteristics—such as income, occupation, and educational level (along with demographic and historical interactions/transactions with the company). But the algorithm itself is not able to define such socioeconomic characteristics and extract such features, for instance, from survey forms completed by the customer or obtained from social media.

While such a structured, human-mediated machine learning approach has been working fine for rather abstract and formal tasks, it is extremely challenging to get it to work for some informal yet seemingly easy (to humans) tasks such as face identification or speech recognition since such tasks require a great deal of knowledge about the world (Goodfellow et al., 2016). It is not straightforward, for instance, to train a machine learning algorithm to accurately recognize the real meaning of a sentence spoken by a person just by manually providing it with a number of grammatical or semantic features. Accomplishing such a task requires deep knowledge about the world that is not easy to formalize and explicitly present. What deep learning has added to the classic machine learning methods is in fact the ability to automatically acquire the knowledge required to accomplish such informal tasks and consequently extract some advanced features that contribute to superior system performance.

To develop an intimate understanding of deep learning, it is important to understand where it fits in the big-picture AI family of methods. A simple hierarchical relationship diagram, or a taxonomy-like representation, may provide such a holistic understanding. In an attempt to create such a taxonomy, Goodfellow and his colleagues

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

(2016) categorized deep learning as part of the representation learning family of methods. Representation learning techniques account for one type of machine learning (which is also a part of AI) in which the emphasis is on learning and discovering features by the system in addition to discovering the mapping from those features to the output/target. Figure 9.1 uses a Venn diagram to illustrate the placement of deep learning within the overarching family of AI-based learning methods.

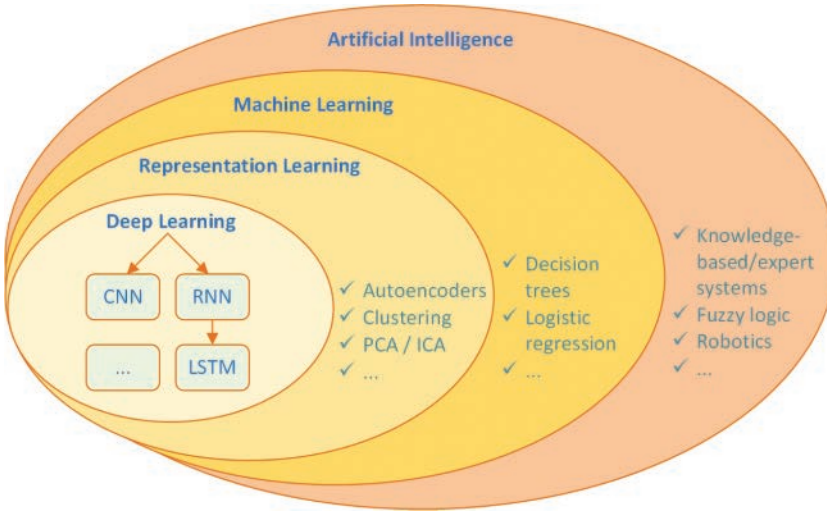


Figure 9.1 A Venn Diagram Showing the Placement of Deep Learning Within the AI-Based Learning Methods

Figure 9.2 highlights the differences in the steps/tasks that need to be performed when building a typical deep learning model versus the steps/tasks performed when building models with classic machine learning algorithms. In this figure, the shaded boxes indicate components that are able to learn directly from data. As shown in the top two workflows, knowledge-based systems and classic machine learning methods require data scientists to manually create the features (i.e., the representation) to achieve the desired output. The workflows at the bottom of the figure show that deep learning enables the

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

computer to derive some complex features from simple concepts that would require a lot of effort for a human to discover manually (or would perhaps be impossible in some problem situations); the figure then maps those advanced features to the desired output.

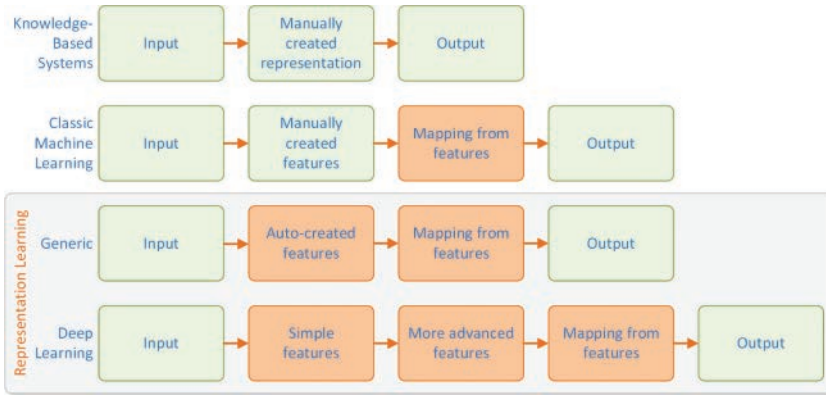


Figure 9.2 The Key Differences Between Classic Machine Learning Methods and Representation Learning/Deep Learning

From a methodological viewpoint, although deep learning is generally believed to be a new area in machine learning, its initial idea goes back to the late 1980s, just a few decades after the emergence of artificial neural networks, when LeCun et al. (1989) published an article about applying backpropagation networks for recognizing handwritten zip codes. In fact, as it is being practiced today, deep learning seems to be nothing but an extension of neural networks, with the idea that deep learning is able to deal with more complicated tasks with a higher level of sophistication by employing many layers of connected neurons along with much larger data sets to automatically characterize variables and solve the problems—but only at the expense of a great deal of computational effort. This very high computational requirement and the need for very large data sets were the two main reasons the initial idea had to wait more than two decades until some advanced computational and technological infrastructure emerged for the practical realization of deep learning. Although the

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

scale of neural networks has dramatically increased in the past decade, thanks to the advancement of related technologies, it is still estimated that having artificial deep neural networks with a comparable number of neurons and level of complexity existing in the human brain will take several more decades.

In addition to the computer infrastructures mentioned earlier, the availability of large and feature-rich digitized data sets is another key factor in the development of successful deep learning applications in recent years. Obtaining good performance from a deep learning algorithm used to be a very difficult task that required extensive skills, experience, and in-depth understanding to design task-specific networks, and therefore, not many were able to develop deep learning for practical and/or research purposes. Large training data sets, however, have greatly compensated for the lack of intimate knowledge and reduced the level of skill needed for implementing deep neural networks. Nevertheless, although the size of available data sets has exponentially increased in recent years, a great challenge, especially for supervised learning of deep networks, is now the labeling of the cases in these huge data sets. As a result, a great deal of research is ongoing, focusing on how we can take advantage of large quantities of unlabeled data for semi-supervised or unsupervised learning or how we can develop methods to label examples in bulk in a reasonable time.

The following section provides a general introduction to neural networks, which are the "shallow" intelligence from which deep learning originated. Next, the chapter introduces different types of deep learning architectures and how they work, some common applications of these deep learning architectures, and some popular computer frameworks used in implementing deep learning in practice. Since, as mentioned, the basics of deep learning are the same as those of artificial neural networks, in the following section, we provide brief coverage of the neural network architecture—the multilayer

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

perceptron (MLP) type of neural networks—and focus on mathematical principles and then look at how the various types of deep learning architectures/approaches were derived from these foundations.

Basics of "Shallow" Neural Networks

Artificial neural networks are essentially simplified abstractions of the human brain and its complex biological networks of neurons. The human brain has billions of interconnected neurons that facilitate our thinking, learning, and understanding of the world around us. Theoretically speaking, learning is nothing but the establishment and adaptation of new or existing inter-neuron connections. In artificial neural networks, however, neurons are processing units (also called processing elements [PEs]) that perform predefined mathematical operations on the numeric values coming from the input variables or from the other neuron outputs to create and push out their own outputs. Figure 9.3 shows a schematic representation of a single-input and single-output neuron—or, more accurately, the PE in an artificial neural network.

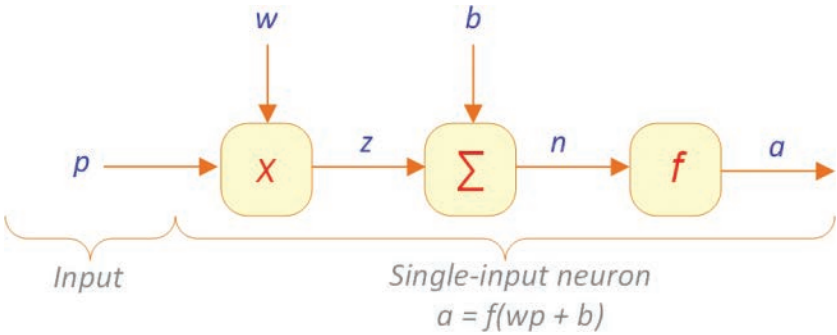


Figure 9.3 General Single-Input Artificial Neuron Representation

In this figure, p represents a numeric input. Each input goes into the neuron with an *adjustable* weight w and a bias term b. A multiplication *weight function* applies the weight to the input, and a *net input function* Σ adds the bias term to the weighted input z.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The output of the net input function (n, known as the *net input*) then goes through another function called the *transfer* (a.k.a. activation) function (shown by f) for conversion and the production of the actual output a. In other words:

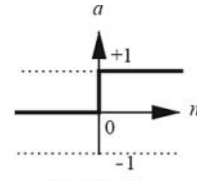
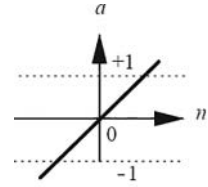
$$a = f(wp + b)$$

As a numeric example, if $w = 2$, $p = 3$, and $b = -1$, then

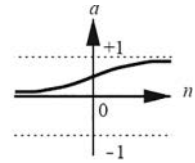
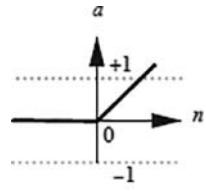
$$a = f(2 \times 3 - 1) = f(5)$$

Various types of transfer functions are commonly used in the design of neural networks. Table 9.1 shows some of the most common transfer functions and their corresponding operations. Note that, in practice, selection of proper transfer functions for a network requires broad knowledge of neural networks—characteristics of the data as well as the specific purpose for which the network is created.

Table 9.1 Common Transfer (Activation) Functions in Neural Networks

Transfer Function	Form	Operation
Hard limit	 *a = hardlim (n)*	$a = +1$ if $n > 0$ $a = 0$ if $n < 0$
Linear	 *a = purelin (n)*	$a = n$

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Transfer Function	Form	Operation
Log-sigmoid	 $a = logsig(n)$	$a = \frac{1}{1+e^{-n}}$
Positive linear (a.k.a. rectified linear, or ReLU)	 $a = poslin(n)$	$a = n$ if $n > 0$ $a = 0$ if $n < 0$

For instance, if in the previous example we had a hard limit transfer function, the actual output a would be $a = \text{hardlim}(5) = 1$. There are some guidelines for choosing the appropriate transfer function for each set of neurons in a network. These guidelines are especially robust for the neurons located at the output layer of the network. For example, if the nature of the output for a model is binary, we are advised to use *sigmoid* transfer functions at the output layer to produce an output between 0 and 1, which represents the conditional probability of $y = 1$ given x or $P(y = 1 \mid x)$. Many neural network textbooks provide and elaborate on those guidelines at different layers in a neural network with some consistency and much disagreement, suggesting that the best practices should (and usually do) come from experience.

Typically, a neuron has more than a single input. With more than one input, each individual input p_i can be shown as an element of the input vector **p**. Each of the individual input values would have its

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

own adjustable weight w_i of the weight vector **W**. Figure 9.4 shows a multiple-input neuron with R individual inputs.

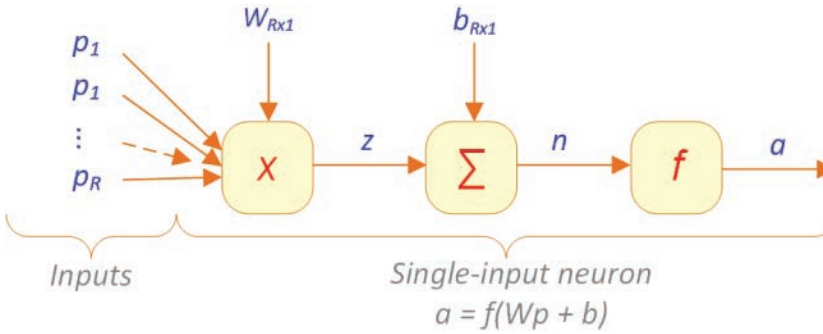


Figure 9.4 Typical Multiple-Input Neuron with R Individual Inputs

For this neuron, the net input n can be expressed as

$$n = w_{1,1}\,p_1 + w_{1,2}\,p_2 + w_{1,3}\,p_3 + \cdots + w_{1,R}\,p_R + b$$

Considering the input vector **p** as an $R \times 1$ vector and the weight vector **W** as a $1 \times R$ vector, n can be written in matrix form as

$$n = \mathbf{Wp} + b$$

where **Wp** is a scalar (i.e., 1×1 vector).

Moreover, each neural network is typically composed of multiple neurons connected to each other and structured in consecutive *layers* so that the outputs of a layer work as the inputs to the next layer. Figure 9.5 shows a typical neural network with three neurons at the input (i.e., first) layer, four neurons at the hidden (i.e., middle) layer, and a single neuron at the output (i.e., last) layer. Each of the neurons has its own weight, weighting function, bias, and transfer function and processes its own input(s), as described.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

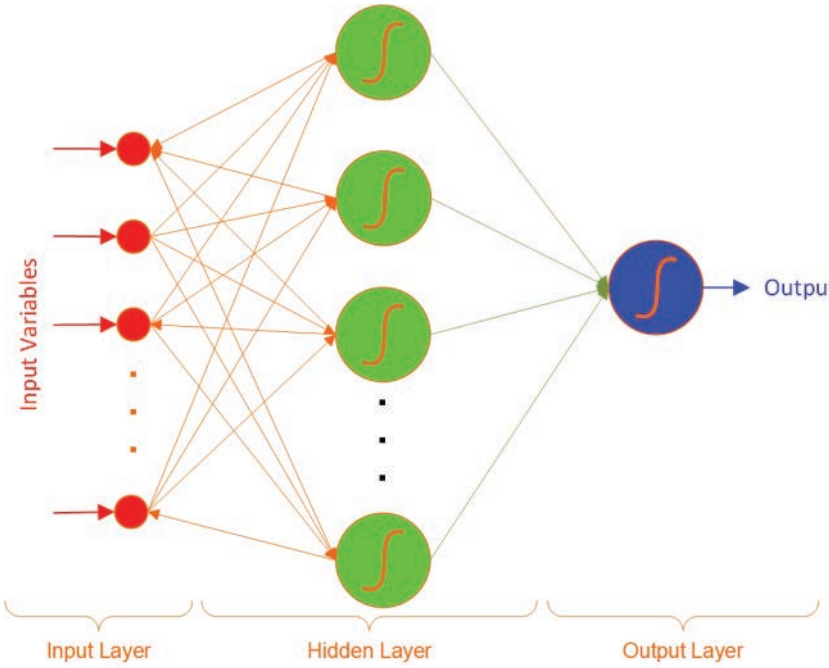


Figure 9.5 Typical Neural Network with Three Layers and Eight Neurons

While the inputs, weighting functions, and transfer functions in a given network are fixed, the values of the weights and biases are adjustable. The process of adjusting weights and biases in a neural network is commonly called *training*. In fact, in practice, a neural network cannot be used effectively for a prediction problem unless it is well trained by a sufficient number of examples with known *actual outputs* (a.k.a. *targets*). The goal of the training process is to adjust network weights and biases such that the network output for each set of inputs (i.e., each sample) is adequately close to its corresponding target value.

Elements of an Artificial Neural Network

A neural network is composed of processing elements that are organized in different ways to form the network's structure. The basic

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

processing unit in a neural network is the neuron, and a number of neurons are organized to establish a network of neurons. Neurons can be organized in a number of different ways; these various network patterns are referred to as *topologies*, or *network architectures*. One of the most popular approaches, known as the *feedforward-multi- layered perceptron*, allows all neurons to link the output in one layer to the input of the next layer, but it does not allow any feedback linkage (Haykin, 2009).

Processing Elements (PEs)

The PE of an ANN is an artificial neuron. Each neuron receives inputs, processes them, and delivers a single output, as shown in Figure 9.5. The input can be raw input data or the output of other processing elements. The output can be the final result (e.g., 1 means yes, 0 means no), or it can be input to other neurons.

Network Structure

Each ANN is composed of a collection of neurons that are grouped into layers. A typical structure is shown in Figure 9.6, where the abbreviations have the following meanings:

PE	Processing element (an artificial representation of a biological neuron)
X_i	Inputs to a PE
Y	Output generated by a PE
Σ	Summation function
f	Activation/transfer function

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

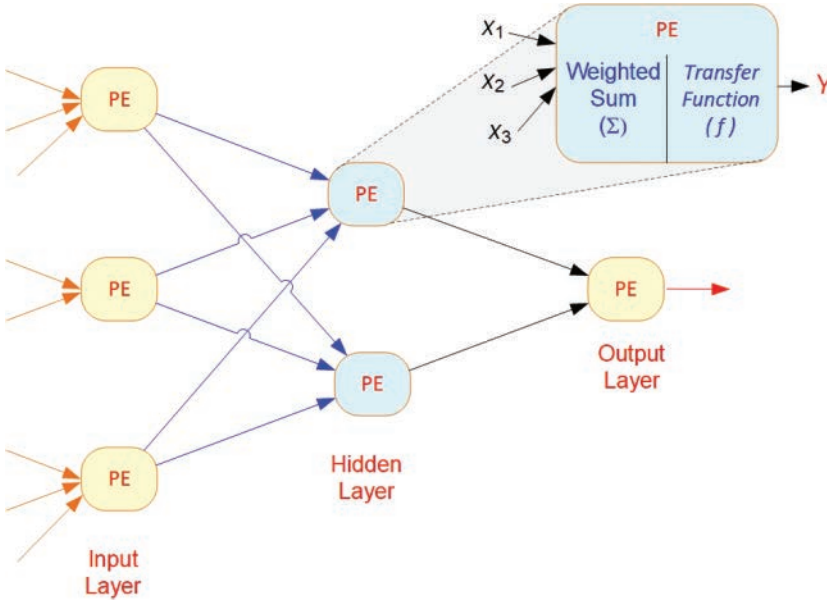


Figure 9.6 Neural Network with One Hidden Layer

Note the three layers in the figure: input, intermediate (called the *hidden layer*), and output. A hidden layer is a layer of neurons that takes inputs from the previous layer and converts those inputs into outputs for further processing. Several hidden layers can be placed between the input and output layers, although it is common to use only one hidden layer. In that case, the hidden layer simply converts inputs into a nonlinear combination and passes the transformed inputs to the output layer. The most common interpretation of the hidden layer is as a feature-extraction mechanism; that is, the hidden layer converts the original inputs in the problem into a higher-level combination of such inputs.

In an artificial neural network, when information is processed, many of the processing elements perform their computations at the same time. This parallel processing resembles the way the human brain works, and it differs from the serial processing of conventional computing.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Input

Each input corresponds to a single attribute. For example, if the problem is to decide on approval or disapproval of a loan, attributes could include the applicant's income level, age, and home ownership status. The numeric value, or the numeric representation of a nonnumeric value, of an attribute is the input to the network. Several types of data, such as text, picture, and voice, can be used as inputs. Preprocessing may be needed to convert the data into meaningful inputs from symbolic/nonnumeric data or to numeric/scale data.

Outputs

The output of a network contains the solution to a problem. For example, in the case of a loan application, the output can be "yes" or "no." The ANN assigns numeric values to the output, which may then need to be converted into categorical output using a threshold value so that the results would be 1 for "yes" and 0 for "no."

Connection Weights

Connection weights are the key elements of an ANN. They express the relative strength (or mathematical value) of the input data or the many connections that transfer data from layer to layer. In other words, weights express the relative importance of each input to a processing element and, ultimately, to the output. Weights are crucial in that they store learned patterns of information. It is through repeated adjustments of weights that a network learns.

Summation Function

The summation function computes the weighted sums of all input elements entering each processing element. A summation function multiplies each input value by its weight and totals the values

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

for a weighted sum. The formula for n inputs (represented with X) in one processing element is shown in Figure 9.7a, and for several processing elements, the summation function formulas are shown in Figure 9.7b.

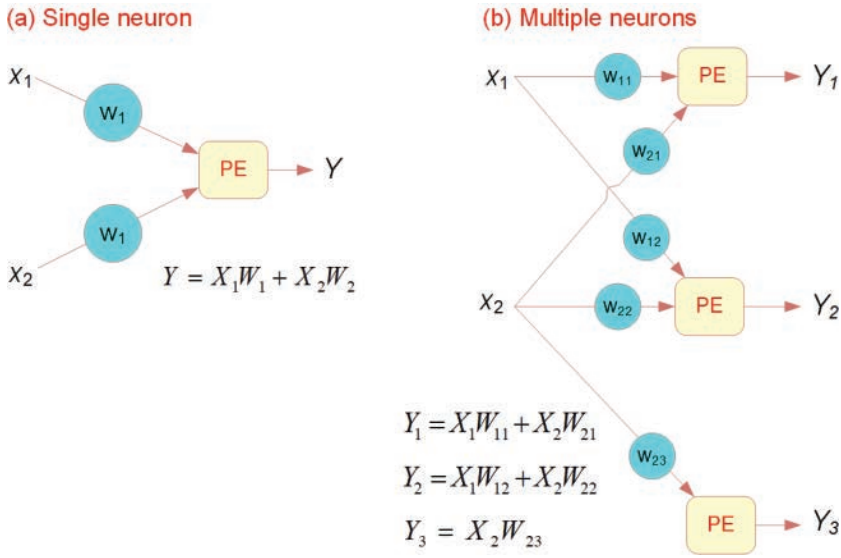


Figure 9.7 Summation Function for (a) a Single Neuron/PE and (b) Several Neurons/PEs

Transfer Function

The summation function computes the internal stimulation, or activation level, of the neuron. Based on this level, the neuron may or may not produce an output. The relationship between the internal activation level and the output can be linear or nonlinear. The relationship is expressed by one of several types of transformation (transfer) functions. (See Table 9.1 for a list of commonly used activation functions.) Selection of the specific activation function affects the network's operation. Figure 9.8 shows the calculation for a simple sigmoid-type activation function example.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

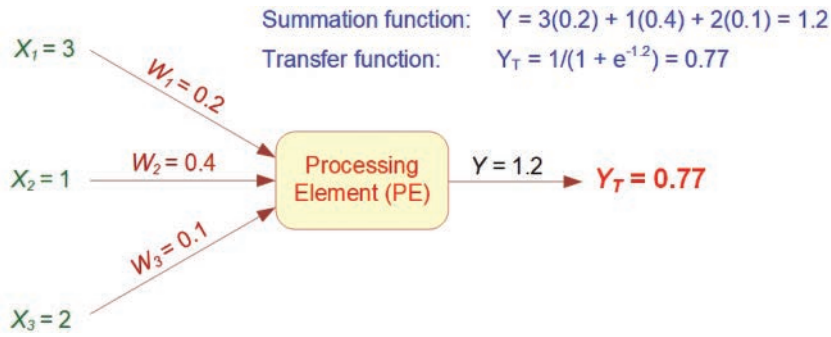


Figure 9.8 ANN Transfer Function Example

The transformation modifies the output levels to fit within a reasonable range of values (typically between 0 and 1). This transformation is performed before the output reaches the next level. Without such a transformation, the value of the output becomes very large, especially when there are several layers of neurons. Sometimes a threshold value is used instead of a transformation function. A threshold value is a hurdle value for the output of a neuron to trigger the next level of neurons. If an output value is smaller than the threshold value, it will not be passed to the next level of neurons. For example, any value of 0.5 or less becomes 0, and any value above 0.5 becomes 1. A transformation can occur at the output of each processing element, or it can be performed only at the final output nodes.

The Learning Process in ANNs

In *supervised learning*, the learning process is inductive; that is, connection weights are derived from existing cases. The usual process of learning involves three tasks (see Figure 9.9):

1. Compute temporary outputs.
2. Compare outputs with desired targets.
3. Adjust the weights and repeat the process.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

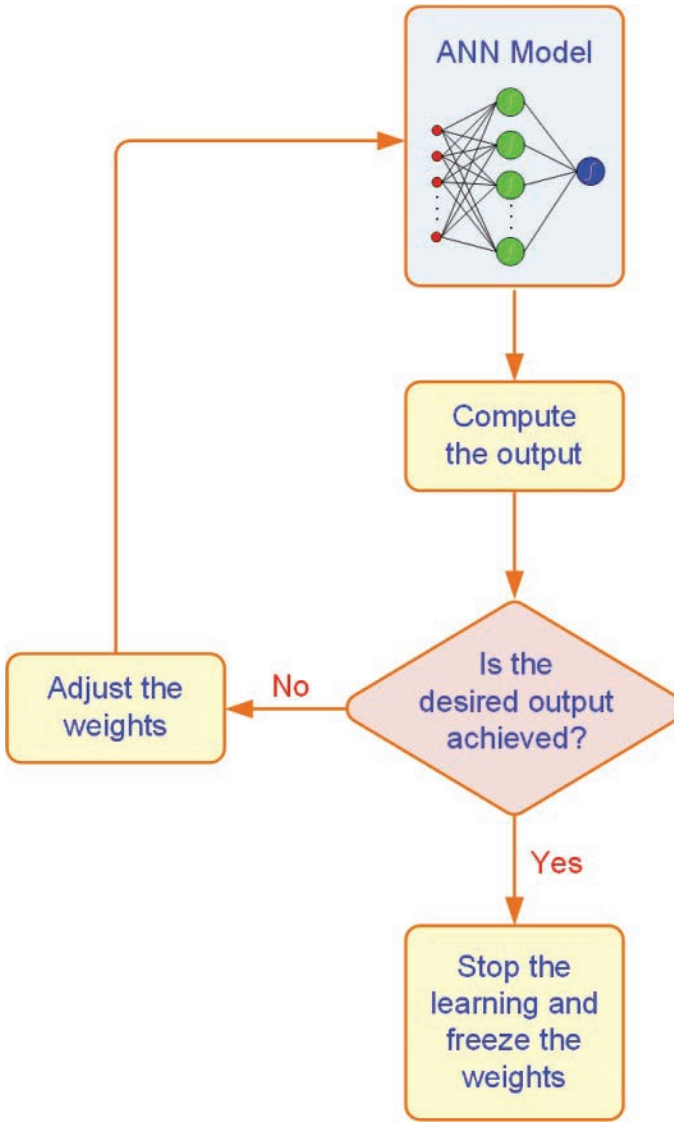


Figure 9.9 Supervised Learning Process in an ANN

Like any other supervised machine learning technique, neural network training is usually done by defining a performance function (F) (a.k.a. *cost function* or *loss function*) and optimizing (minimizing) that function by changing model parameters. Usually, the performance function is just a measure of error (i.e., the difference between

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

the actual input and the target) across all inputs of a network. There are several types of error measures (e.g., sum square errors, mean square errors, cross entropy, and even custom measures), all of which are designed to capture the difference between the network outputs and the actual outputs.

The training process begins by calculating outputs for a given set of inputs using some random weights and biases. Once the network outputs are on hand, the performance function can be computed. The difference between the actual output (Y or Y_T) and the desired output (Z) for a given set of inputs is an error called *delta*. (In calculus, the Greek symbol delta, Δ, means "difference.")

The objective is to minimize delta (i.e., reduce it to 0 if possible), which is done by adjusting the network's weights. The key is to change the weights in the proper direction, making changes that reduce delta (i.e., error). Different ANNs compute delta in different ways, depending on the learning algorithm being used. Hundreds of learning algorithms are available for various situations and ANN configurations.

Backpropagation for ANN Training

The optimization of performance (i.e., minimization of the error or delta) in a neural network is usually done by an algorithm called stochastic gradient descent (SGD), which is an iterative gradient-based optimizer used for finding the minimum (i.e., the lowest point) in performance functions, as in the case of neural networks. The idea behind the SGD algorithm is that the derivative of the performance function with respect to each current weight or bias indicates the amount of change in the error measure by each unit of change in that weight or bias element. These derivatives are referred to as *network gradients*. Calculation of network gradients in neural networks requires application of an algorithm called *backpropagation*, which is the most popular neural network learning algorithm and applies the *chain rule of calculus* to compute the derivatives of functions formed by composing other functions whose derivatives are known. (For

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

more on the mathematical details of this algorithm, see Rumelhart et al., 1986.)

Backpropagation (short for *back-error propagation*) is the most widely used supervised learning algorithm in neural computing (Principe et al., 2000). By using the SGD algorithm mentioned previously, the implementation of backpropagation is relatively straightforward. A neural network with backpropagation learning includes one or more hidden layers. This type of network is considered feedforward because there are no interconnections between the output of a processing element and the input of a node in the same layer or in a preceding layer. Externally provided correct patterns are compared with the neural network's output during (supervised) training, and feedback is used to adjust the weights until the network has categorized all training patterns as correctly as possible. (The error tolerance is set in advance.)

Starting with the output layer, errors between network-generated actual output and the desired outputs are used to correct/adjust the weights for the connections between the neurons (see Figure 9.10). For any output neuron j, the error (delta) $= (Z_j - Y_j)\,(df/dx)$ where Z and Y are the desired and actual outputs, respectively. Using the sigmoid function $f = [1 + \exp(-x)]^{-1}$, where x is proportional to the sum of the weighted inputs to the neuron, is an effective way to compute the output of a neuron in practice. With this function, the derivative of the sigmoid function $df/dx = f(1 - f)$ and of the error is a simple function of the desired and actual outputs. The factor $f(1 - f)$ is the logistic function, which serves to keep the error correction well bounded. The weight of each input to the jth neuron is then changed in proportion to this calculated error. A more complicated expression can be derived to work backward in a similar way from the output neurons through the hidden layers to calculate the corrections to the associated weights of the inner neurons. This complicated method is an iterative approach to solving a nonlinear optimization problem that is very similar in meaning to the one characterizing multiple linear regression.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

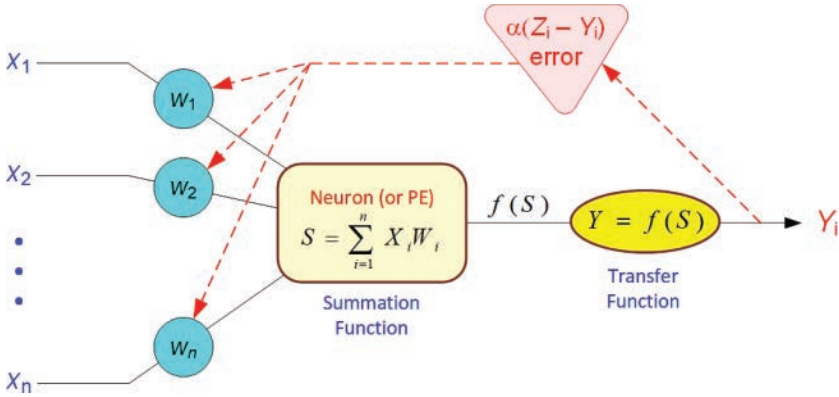


Figure 9.10 Backpropagation of Error for a Single Neuron

In backpropagation, the learning algorithm follows these steps:

1. Initialize weights with random values and set other parameters.
2. Read in the input vector and the desired output.
3. Compute the actual output via the calculations, working forward through the layers.
4. Compute the error.
5. Change the weights by working backward from the output layer through the hidden layers.

This procedure is repeated for the entire set of input vectors until the desired output and the actual output agree within some predetermined tolerance. Given the calculation requirements for one iteration, training a large network can take a very long time; therefore, in one variation, a set of cases is run forward and an aggregated error is fed backward to speed the learning. Sometimes, depending on the initial random weights and network parameters, the network does not converge to a satisfactory performance level. When this is the case, new random weights must be generated, and the network parameters, or even the structure of the network, may have to be modified before

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

another attempt is made. Current research is aimed at developing algorithms and using parallel computers to improve this process. For example, genetic algorithms can be used to guide the selection of the network parameters to maximize the performance of the desired output. In fact, most commercial ANN software tools are now using genetic algorithms to help users optimize the network parameters in a semiautomated manner.

A central concern in the training of any type of machine learning model is overfitting. It happens when the trained model is highly fitted to the training data set but performs poorly with regard to external data sets. Overfitting causes serious issues with respect to the generalizability of the model. A large group of strategies known as *regularization* strategies is designed to prevent models from overfitting by making changes or defining constraints for the model parameters or the performance function.

In the classic ANN models of small size, a common regularization strategy to avoid overfitting is to assess the performance function for a separate validation data set as well as the training data set after each iteration. Whenever the performance stops improving for the validation data, the training process is stopped. Figure 9.11 shows a typical graph of the error measure by the number of iterations of training. As shown, in the beginning, the error decreases in both training and validation data by running more and more iterations; but from a specific point (shown by the dashed line), the error starts increasing in the validation set while still decreasing in the training set. This means that beyond that number of iterations, the model becomes overfitted to the data set with which it is trained, and it cannot necessarily perform well when it is fed with some external data. That point actually represents the recommended number of iterations for training a given neural network.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

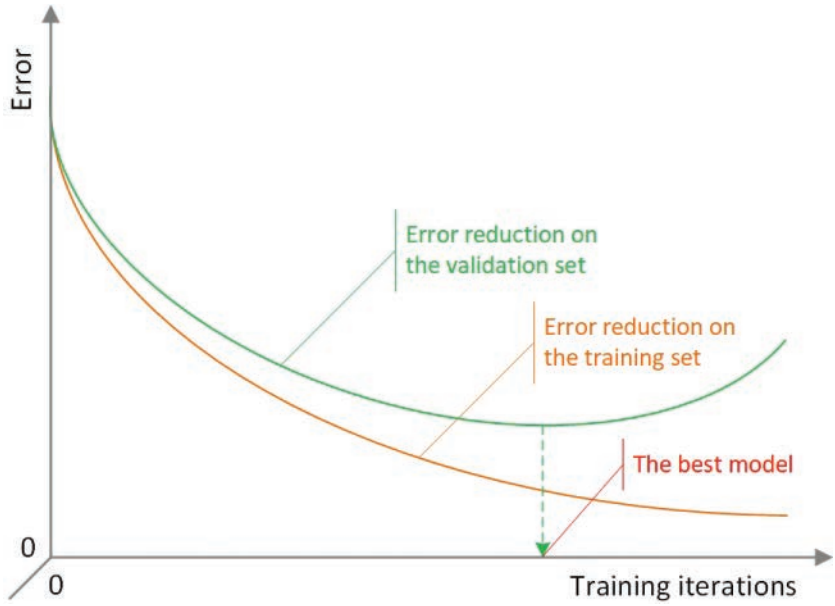


Figure 9.11 Illustration of the Overfitting in ANN—Gradually Changing Error Rates in the Training and Validation Data Sets As the Number of Iterations Increases

Deep Neural Networks

Until recently—that is, until the advent of deep learning—most neural network applications involved network architectures with only a few hidden layers and a limited number of neurons in each layer. Even in relatively complex business applications of neural networks, the number of neurons in networks hardly exceeded a few thousand. In fact, the processing capability of computers at the time was such a limiting factor that central processing units (CPUs) were hardly able to run networks involving more than a couple of layers in a reasonable time. In recent years, development of graphics processing units (GPUs) along with the associated programming languages (e.g., CUDA by NVIDIA) that enable people to use them for data analysis purposes have led to more advanced applications of neural networks. GPU technology has enabled us to successfully run neural networks

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

with over a million neurons. These larger networks are able to go deeper into the data features and extract more sophisticated patterns that could not be detected otherwise.

While deep networks can handle a considerably larger number of input variables, they also need relatively larger data sets to be trained satisfactorily; using small data sets for training deep networks typically leads to overfitting of the model to the training data and poor and unreliable results in the case of applying to external data. Thanks to the Internet- and Internet of Things (IoT)–based data-capturing tools and technologies, larger data sets are now available in many application domains for deeper neural network training.

The input to a regular ANN model is typically an array of size $R \times 1$, where R is the number of input variables. In the deep networks, however, we are able to use *tensors* (i.e., n-dimensional arrays) as input. For example, in image recognition networks, each input (i.e., image) can be represented by a matrix indicating the color codes used in the image pixels; or for video processing purposes, each video can be represented by several matrices (i.e., a 3D tensor), each representing an image involved in the video. In other words, tensors provide us with the ability to include additional dimensions (e.g., time, location) for analyzing the data sets.

Except for these general differences, the different types of deep networks involve various modifications to the architecture of standard neural networks that equip them with distinct capabilities for dealing with particular data types for advanced purposes. In the following section, we discuss some of these special network types and their characteristics.

Feedforward Multilayer Perceptron (MLP) Deep Networks

MLP deep networks, also known as *deep feedforward networks*, are the most general type of deep networks. These networks are simply

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

large-scale neural networks that can contain many layers of neurons and handle tensors as their input. The types and characteristics of the network elements (i.e., weight functions, transfer functions) are much the same as in the standard ANN models. These models are called *feedforward* because the flow of information that goes through them is always forwarding, and no feedback connections (i.e., connections in which outputs of a model are fed back to the model) are allowed. The neural networks in which feedback connections are allowed are called *recurrent neural networks* (*RNNs*). (General RNN architectures as well as a specific variation of RNNs called *long short-term memory networks* are discussed later in this chapter.)

Generally, a sequential order of layers has to be held between the input and the output layers in the MLP-type network architecture. This means that the input vector has to pass through all layers sequentially and cannot skip any of them; moreover, it cannot be directly connected to any layer except for the very first one; the output of each layer is the input to the subsequent layer. Figure 9.12 provides a vector representation of the first three layers of a typical MLP network. As shown, there is only one vector going into each layer, which is either the original input vector (p for the first layer) or the output vector from the previous hidden layer in the network architecture (a^{i-1} for the ith layer). There are, however, some special variations of MLP network architectures designed for specialized purposes in which these principles can be violated.

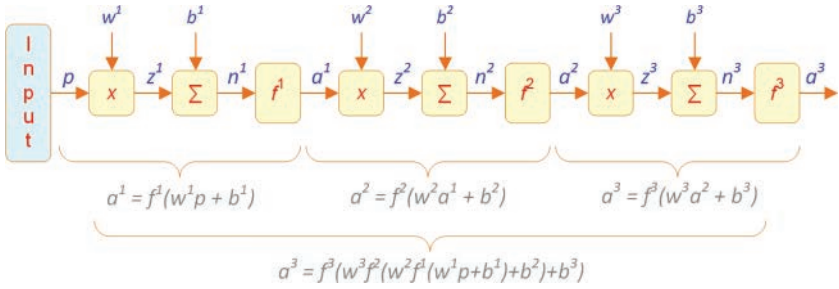


Figure 9.12 Vector Representation of the First Three Layers in a Typical MLP Network

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Impact of Random Weights in Deep MLP

Optimization of the performance (loss) function in many real applications of deep MLPs is a challenging issue. The problem is that applying the common gradient-based training algorithms with random initialization of weights and biases that is very efficient for finding the optimal set of parameters in shallow neural networks most of the time could lead to getting stuck in the locally optimal solutions rather than catching the global optimum values for the parameters. As the depth of a network increases, the chance of reaching a global optimum using random initializations with the gradient-based algorithms decreases. In such cases, pretraining the network parameters using some *unsupervised* deep learning methods such as deep belief networks (DBNs) can usually be helpful (Hinton et al., 2006). DBNs are a type of a large class of deep neural networks called *generative models*. The introduction of DBNs, which occurred in 2006, is considered the beginning of the current deep learning renaissance (Goodfellow et al., 2016) because prior to that, deep models were considered too difficult to optimize. In fact, the primary application of DBNs today is to improve classification models through pretraining of their parameters.

Using these unsupervised learning methods, we can train the MLP layers, one at a time, starting from the first layer; we can use the output of each layer as the input to the subsequent layer and initialize that layer with an unsupervised learning algorithm. At the end, we will have a set of initialized values for the parameters across the whole network. Those pretrained parameters can then be used instead of random initialized parameters as the initial values in the supervised learning of the MLP. This pretraining procedure leads to significant improvements to the deep classification applications. Figure 9.13 illustrates the classification errors that resulted from training a deep MLP network with (blue circles) and without (black triangles) pretraining of parameters (Bengio, 2009). In this example, the blue line represents the observed error rates of testing a classification

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

model (on 1,000 held-out examples) trained using a purely supervised approach with 10 million examples; the black line indicates the error rates on the same testing data set when 2.5 million examples were initially used for unsupervised training of network parameters (using DBN) and then the other 7.5 million examples along with the initialized parameters were used to train a supervised classification model. The diagrams clearly show a significant improvement in terms of the classification error rate in the model pretrained by a deep belief network.

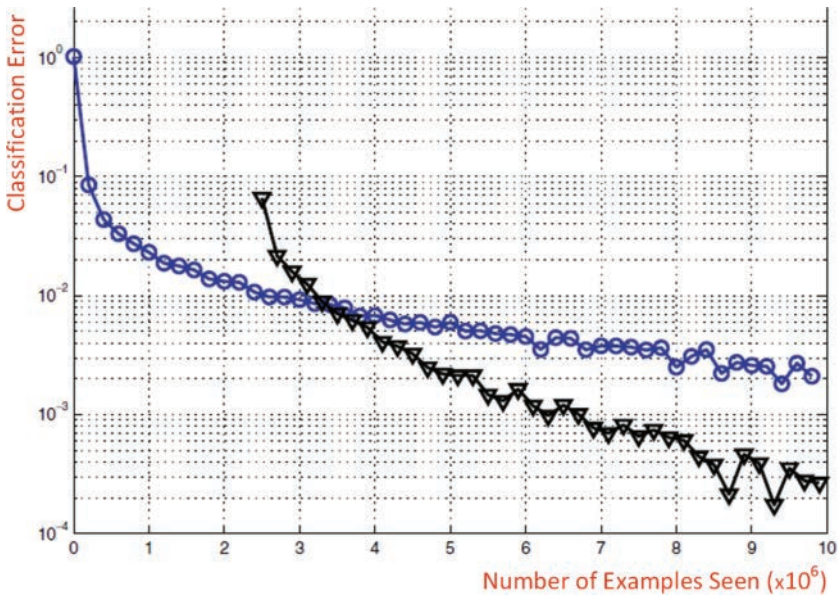


Figure 9.13 The Effect of Pretraining Network Parameters on Improving Results of a Classification-Type Deep Neural Network

More Hidden Layers Versus More Neurons?

An important question regarding the deep MLP models is "Would it make sense (and produce better results) to restructure such

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

networks with only a few layers but many neurons in each?" In other words, why do we need deep MLP networks with many layers when we can include the same number of neurons in just a few layers (i.e., wide networks instead of deep networks)? According to the universal approximation theorem (Cybenko, 1989; Hornik, 1991), a sufficiently large single-layer MLP network can approximate any function. Although theoretically founded, such a layer with many neurons may be prohibitively large and hence may fail to learn the underlying patterns correctly. A deeper network can reduce the number of neurons required at each layer and hence decrease the generalization error. Whereas theoretically it is still an open research question, practically using more layers in a network seems to be more effective and computationally more efficient than using many neurons in a few layers.

Like typical artificial neural networks, multilayer perceptron networks can also be used for various prediction, classification, and clustering purposes. Especially when a large number of input variables are involved or when the nature of input has to be an *n*-dimensional array, a deep multilayer network design needs to be employed.

Convolutional Neural Networks

Convolutional neural networks (CNNs) are among the most popular types of deep learning methods (LeCun et al., 1989). CNNs are, in essence, variations of the deep MLP architecture, initially designed for computer vision applications (e.g., image processing, video processing, text recognition) but also applicable to non-image data sets.

The main characteristic of the convolutional networks is having at least one layer involving a *convolution weight function* instead of general matrix multiplication. Figure 9.14 illustrates a typical convolutional unit.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

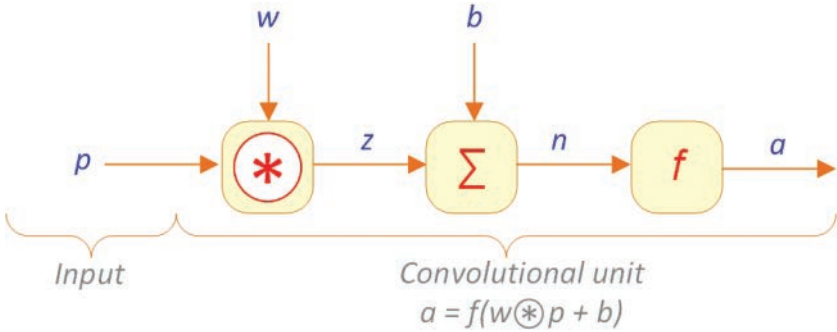


Figure 9.14 Typical Convolutional Network Unit

Convolution, typically shown by the ⊛ symbol, is a linear operation that essentially aims at extracting simple patterns from sophisticated data patterns. For instance, in processing an image containing several objects and colors, convolution functions can extract simple patterns such as the existence of horizontal or vertical lines or edges in different parts of the picture. (We discuss convolution functions in more detail in the next section.)

A layer containing a convolution function in a CNN is called a *convolution layer*. This layer is often followed by a *pooling* (a.k.a. *subsampling*) layer. Pooling layers are in charge of consolidating the large tensors into a tensor with a smaller size; they reduce the number of model parameters while keeping the important model features. Different types of pooling layers are also discussed in the following sections.

Convolution Functions

Earlier in this chapter, when we described MLP networks, we said that the weight function is generally a matrix manipulation function that multiplies the weight vector by the input vector to produce the output vector in each layer. Having a very large input vector/tensor, which is the case in most deep learning applications, we need a large number of weight parameters so that each single input to each neuron can be assigned a single weight parameter. For instance, in

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

an image-processing task using a neural network for images of size 150 × 150 pixels, each input matrix contains 22,500 (i.e., 150 times 150) integers, each of which should be assigned its own weight parameter for each neuron it goes into throughout the network. Therefore, having even only a single layer requires thousands of weight parameters to be defined and trained. As you might guess, this fact would dramatically increase the required time and processing power required to train a network because in each training iteration, all of those weight parameters have to be updated by the SGD algorithm. The solution to this problem is the convolution function.

A convolution function can be thought of as a trick to address the issue defined in the previous paragraph. The trick is called *parameter sharing*, which provides computational efficiency as well as other benefits. Specifically, in a convolution layer, instead of having a weight for each input, there is a set of weights, referred to as the *convolution kernel*, or *filter*, that is shared between inputs and moves around the input matrix to produce the outputs.

Pooling

Most of the times, a convolution layer is followed by another layer known as the *pooling* (a.k.a. *subsampling*) layer. The purpose of a pooling layer is to consolidate elements in the input matrix to produce a smaller output matrix while maintaining the important features. Normally, a pooling function involves an $r \times c$ (i.e., rows × columns) consolidation window (similar to a kernel in the convolution function) that moves around the input matrix and in each move calculates some summary statistics of the elements involved in the consolidation window so that it can be put it in the output image. For example, a particular type of pooling function called *average pooling* takes the average of the input matrix elements involved in the consolidation window and uses that average value as an element of the output matrix in the corresponding location. Similarly, the *max pooling*

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

function (Zhou et al., 1988) takes the maximum of the values in the window as the output element. Unlike with convolution, for the pooling function, given the size of the consolidation window (i.e., r and c), stride should be carefully selected so that there are no overlaps in the consolidations. The pooling operation using an $r \times c$ consolidation window reduces the number of rows of the input matrix by a factor of r and the number of columns by a factor of c. For example, using a 3×3 consolidation window, a 15×15 matrix is consolidated to a 5×5 matrix.

In addition to reducing the number of parameters, pooling is especially useful in deep learning image-processing applications in which the critical task is to determine whether a feature (e.g., a particular animal) is present in an image but the exact spatial location of that feature in the picture is not important. However, if the location of features is important in a particular context, applying a pooling function could potentially be misleading.

There are various types of pooling operations, such as max pooling, average pooling, the L^2 norm of a rectangular neighborhood, and weighted average pooling. The choice of proper pooling operation as well as the decision to include a pooling layer in the network depends to a great extent on the context and properties of the problem that the network is solving.

Image Processing Using Convolutional Networks

Real applications of deep learning in general and CNNs in particular depend on the availability of large, annotated data sets. Theoretically, CNNs can be applied to many practical problems, and today there are many large and feature-rich databases for such applications available. Nevertheless, the biggest challenge is that in supervised learning applications, you need an already annotated (i.e., labeled) data set to train the model before you can use it for prediction/identification of other unknown cases. Extracting features of data sets using CNN layers is an unsupervised task, but the extracted features will

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

not be much use without having labeled cases to develop a classification network in a supervised learning fashion. This is why image classification networks traditionally involve two pipelines: visual feature extraction and image classification.

ImageNet (http://www.image-net.org) is an ongoing research project that provides researchers with a large database of images, each linked to a set of synonym words (known as *synset*) from WordNet (a word hierarchy database). Each synset represents a particular concept in the WordNet.

Currently, WordNet includes more than 100,000 synsets, each of which is supposed to be illustrated by an average of 1,000 images in ImageNet. ImageNet is a huge database for developing image-processing-type deep networks. It contains more than 15 million labeled images in 22,000 categories. Because of its sheer size and proper categorization, ImageNet is by far the most widely used benchmarking data set for assessing the efficiency and accuracy of deep networks designed by deep learning researchers.

One of the first convolutional networks designed for image classification using the ImageNet data set was AlexNet (Krizhevsky et al., 2012). It was composed of five convolution layers followed by three fully connected (a.k.a. dense) layers. (Figure 9.15 provides a schematic representation of AlexNet.) One of the factors in this relatively simple architecture that made its training remarkably fast and computationally efficient was the use of rectified linear unit (ReLu) transfer functions instead of the traditional sigmoid functions in the convolution layers. By using ReLu transfer functions, the designers addressed an issue called the *vanishing gradient problem*, which is caused by very small derivatives of sigmoid functions in some regions of the images. The other important contribution of this network that has a dramatic role in improving the efficiency of deep networks was the introduction of the concept of dropout layers to the CNNs as a regularization technique to reduce overfitting. A dropout layer typically comes after the fully connected layers and applies a random probability to the neurons to switch off some of them and make the network sparser.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

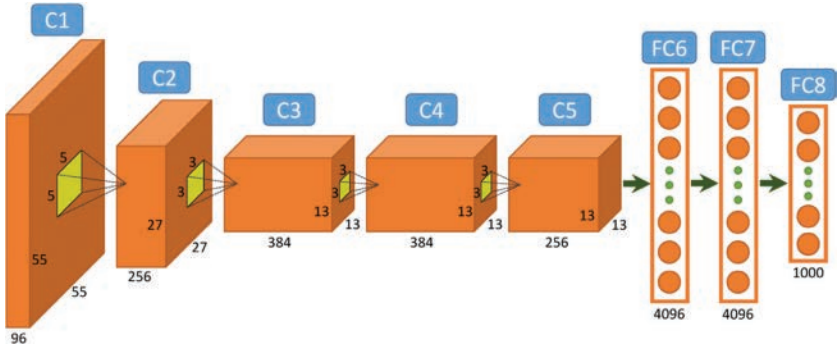


Figure 9.15 Architecture of AlexNet, a Convolutional Network for Image Classification

In recent years, in addition to a large number of data scientists showcasing their deep learning capabilities, a number of well-known industry-leading companies such as Microsoft, Google, and Facebook have participated in the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The goal of the ILSVRC classification task is to design and train networks that are capable of classifying 1.2 million input images into one of the 1,000 image categories. For instance, GoogLeNet, a deep convolutional network architecture designed by Google researchers, was the winning architecture of ILSVRC 2014, with a 22-layer network and only a 6.66% classification error rate, only slightly worse than the human-level classification error, which was 5.1% (Russakovsky et al., 2015). The main contribution of the GoogLeNet architecture was to introduce a module called *Inception*. The idea of Inception is that because one would have no idea about what size convolution kernel would perform the best on a particular data set, it is better to include multiple convolutions and let the network decide which one to use. Therefore, as shown in Figure 9.16, in each convolution layer, the data coming from the previous layer is passed through multiple types of convolution, and the outputs are concatenated before going to the next layer. This type of architecture allows the model to take into account both local features via smaller convolutions and high abstracted features via larger ones.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

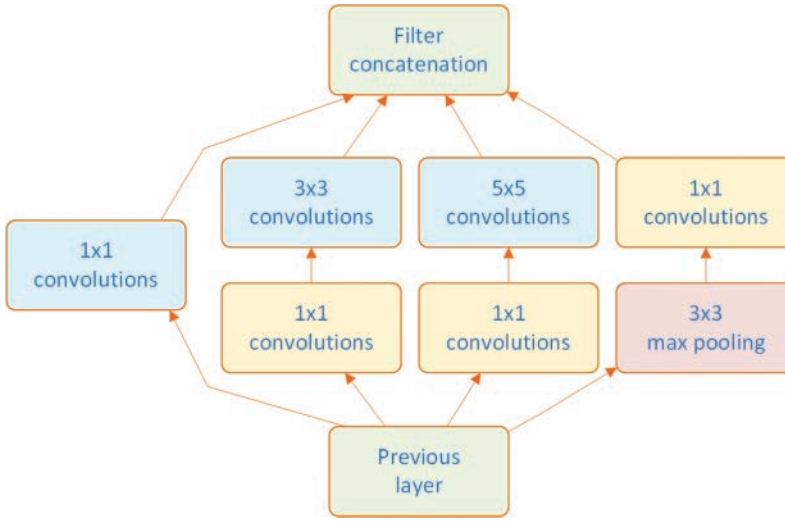


Figure 9.16 Conceptual Representation of the Inception Feature in GoogLeNet

Google recently launched a new service, Google Lens, that uses deep learning artificial neural network algorithms (along with other AI techniques) to deliver information about the images captured by users from nearby objects. It involves identifying the objects—products, plants, animals, and so on—and locations and providing information about them on the Internet. Some other features of this service are the capability of saving contact information from a business card image on the phone, identifying types of plants and species of animals, identifying books and movies from their cover photos, and providing information (e.g., stores, theaters, shopping, reservations) about them. Figure 9.17 shows two examples of using the Google Lens app on an Android mobile device.

Even though more accurate networks have been developed (e.g., He et al., 2015) in terms of efficiency and processing requirements (i.e., with a smaller number of layers and parameters), GoogLeNet is considered one of the best architectures to date. Apart from AlexNet and GoogLeNet, several other convolutional network architectures, including Residual Networks (ResNet), VGGNet, and Xception, have been developed and contributed to the image-processing area; all of them rely on the ImageNet database.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

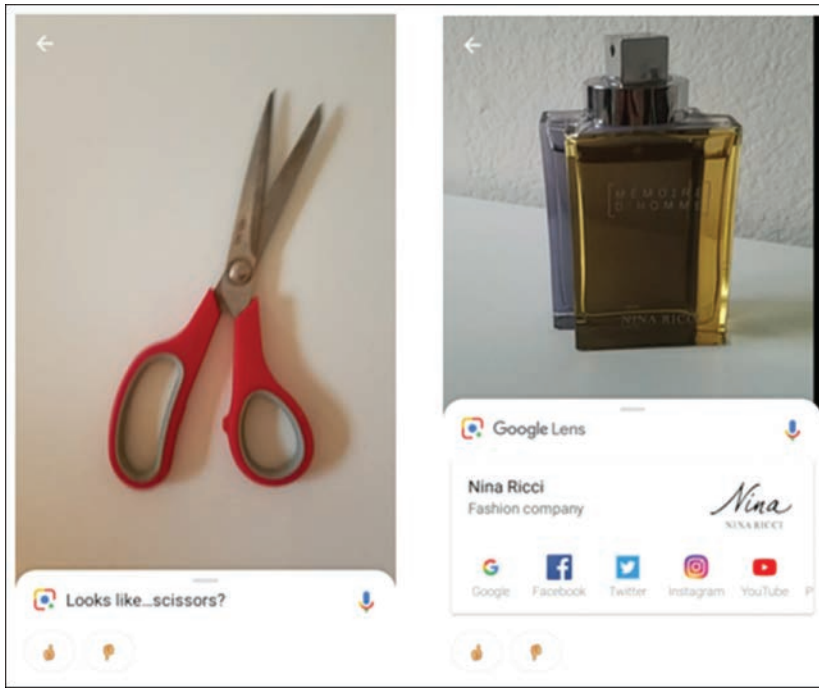


Figure 9.17 Two Examples of Using Google Lens, a Service Based on Convolutional Deep Networks for Image Recognition

In a May 2018 effort to address the labor-intensive task of labeling images on a large scale, Facebook published a *weakly supervised training* image recognition deep learning project (Mahajan et al., 2018). This project trained a deep learning image recognition model based on hashtags made by users on images posted on Instagram. The model was trained using 3.5 billion Instagram images labeled with around 17,000 hashtags using 336 GPUs working in parallel; the training procedure took a few weeks. A preliminary version of the model (trained using only 1 billion images and 1,500 hashtags) was then tested on the ImageNet benchmark data set and is reported to have outperformed the state-of-the-art models in terms of accuracy by more than 2%. This big achievement by Facebook surely will open

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

doors to a new world of image processing using deep learning since it can dramatically increase the size of available image data sets that are labeled for training purposes.

Text Processing Using Convolutional Networks

Image processing was the main reason for the development of convolutional networks, but these networks have also been shown to be useful in some large-scale text mining tasks as well. Especially since 2013, when Google published its word2vec project (Mikolov et al., 2013a, 2013b), the applications of deep learning for text mining have increased remarkably.

Word2vec is a two-layer neural network that gets a large text corpus as input and converts each word in the corpus to a numeric vector of any given size (typically ranging from 100 to 1,000) with very interesting features. Although word2vec itself is not a deep learning algorithm, its outputs (word vectors, also known as word embeddings) already have been widely used in many deep learning research and commercial projects as inputs.

One of the most interesting properties of word vectors created by the word2vec algorithm is maintaining the words' relative associations. For example, the vector operations

$$vector(\text{`}King\text{'}) - vector(\text{`}Man\text{'}) + vector(\text{`}Woman\text{'})$$

and

$$vector(\text{`}London\text{'}) - vector(\text{`}England\text{'}) + vector(\text{`}France\text{'})$$

will result in a vector very close to *vector*(*'Queen'*) and *vector*(*'Paris'*), respectively. Figure 9.18 shows a simple vector representation of the first example in a two-dimensional vector space.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

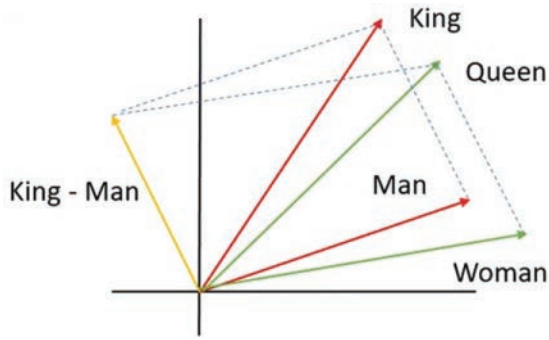


Figure 9.18 Typical Vector Representation of Word Embeddings in a Two-Dimensional Space

Moreover, the vectors are specified in such a way that those of a similar context are placed very close to each other in the *n*-dimensional vector space. For instance, in the word2vec model pretrained by Google using a corpus including about 100 billion words (taken from Google News), the closest vectors to *vector* (*'Sweden'*) in terms of cosine distance, as shown in Table 9.2, identified the names of European countries located geographically near Sweden.

Table 9.2 Example of the word2vec Project Indicating the Closest Word Vectors to the Word *Sweden*

Word	Cosine Distance
Norway	0.760124
Denmark	0.715460
Finland	0.620022
Switzerland	0.588132
Belgium	0.585635
Netherlands	0.574631
Iceland	0.562368
Estonia	0.547621
Slovenia	0.531408

Because word2vec takes into account the contexts in which a word has been used and the frequency of using it in each context in guessing the meaning of the word, we can represent each term with its

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

semantic context instead of just the syntactic/symbolic term itself. As a result, word2vec addresses several word variation issues that used to be problematic in traditional text mining activities. In other words, word2vec is able to handle and correctly represent words including typos, abbreviations, and informal conversations. For instance, the words *Frnce*, *Franse*, and *Frans* would all get roughly the same word embeddings as their original counterpart *France*. Word embeddings are also able to determine other interesting types of associations, such as distinction of entities (e.g., *vector*(*'human'*) − *vector*(*'animal'*) ~ *vector*(*'ethics'*) or geopolitical associations (e.g., *vector*(*'Iraq'*) − *vector*(*'violence'*) ~ *vector*(*'Jordan'*)).

By providing such a meaningful representation of textual data, in recent years, word2vec has driven many deep learning–based text mining projects in a wide range of contexts (e.g., medical, computer science, social media, marketing), and various types of deep networks have been applied to the word embeddings created by this algorithm to accomplish different objectives.

Recurrent Networks and Long Short-Term Memory Networks

Human thinking and understanding to a great extent rely on *context*. It is crucial for us, for example, to know that a particular speaker uses very sarcastic language (based on his previous speeches) to fully catch all the jokes that he makes. Or to understand the meaning of the word *fall* (i.e., either *autumn* or *to collapse*) in the sentence "It is not a nice fall," we would need to know the other words in the surrounding sentences, or we would only be guessing at the meaning. Knowledge of context is typically formed based on observing events that happened in the past. In fact, human thoughts are persistent, and we use every piece of information we previously acquired about an event in the process of analyzing it; we do not throw away our past knowledge and think from scratch every time we face similar events or situations.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Hence, there seems to be a recurrence in the way humans process information.

While deep MLP and convolutional networks are specialized for processing a *static* grid of values such as an image or a matrix of word embeddings, sometimes the *sequence* of input values is also important to the operation of the network to accomplish a given task and hence should be taken into account. Another popular type of neural networks is the recurrent neural network (RNN) (Rumelhart et al., 1986), which is specifically designed to process sequential inputs. An RNN basically models a *dynamic* system where (at least in one of its hidden neurons) the state of the system (i.e., output of a hidden neuron) at each time point t depends on both the inputs to the system at that time and its state at the previous time point $t - 1$. In other words, RNNs are the type of neural networks that have memory and that apply that memory to determine their future outputs. For instance, in designing a neural network to play chess, it is important to take into account several previous moves while training the network because a wrong move by a player can lead to the eventual loss of the game in the subsequent 10 to 15 plays. In addition, to understand the real meaning of a sentence in an essay, sometimes we need to rely on the information in the previous several sentences or paragraphs. In such situations, to get a true understanding, we need the context built sequentially and collectively over time. Therefore, it is crucial to consider a memory element for the neural network that takes into account the effect of prior moves (in the chess example) and prior sentences and paragraphs (in the essay example) to determine the best output. This memory portrays and creates the context required for learning and understanding.

In static networks like MLP-type CNNs, we are trying to find some functions (i.e., network weights, biases) that map the inputs to some outputs that are as close as possible to the actual target. In dynamic networks such as RNNs, on the other hand, both inputs and outputs are sequences (patterns). Therefore, a dynamic network is a

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

dynamic system rather than a function because its output depends not only on the input but also on the previous outputs.

Graphically, a recurrent unit in a network can be depicted in a circuit diagram like the one shown in Figure 9.19. In this figure, D represents the *tap delay lines*, or simply the *delay* element of the network that, at each time point t, contains $a^{(t)}$, the previous output value of the unit. Sometimes instead of storing just one value, we store several previous output values in D to account for the effects of all of them. Also, *iw* and *lw* represent the weight vectors applied to the input and the delay, respectively.

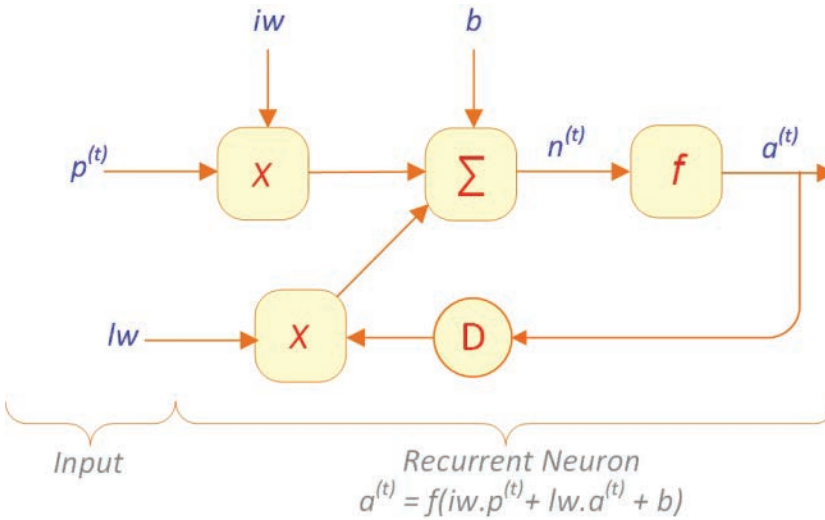


Figure 9.19 Typical Recurrent Network Unit

Technically speaking, any network with feedback can actually be called a *deep network* because even with a single layer, the loop created by the feedback can be thought of as a static MLP-type network with many layers. (See Figure 9.20 for a graphical illustration of this structure.) However, in practice, each recurrent neural network would involve dozens of layers, each with feedback to itself, or even to the previous layers, which makes a recurrent neural network even deeper and more complicated.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

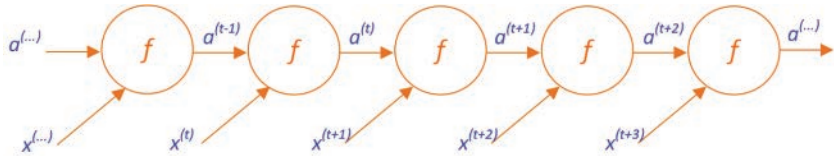


Figure 9.20 Unfolded View of a Typical Recurrent Network

Because of the feedbacks, computation of gradients in the recurrent neural networks would be somewhat different from the general backpropagation algorithm used for the static MLP networks. There are two alternative approaches for computing the gradients in the RNNs: real-time recurrent learning (RTRL) and backpropagation through time (BTT) (whose explanation is beyond the scope of this chapter). Nevertheless, the general purpose remains the same: Once the gradients have been computed, the same procedures are applied to optimize the learning of the network parameters.

Long Short-Term Memory Networks

Long short-term memory (*LSTM*) networks (Hochreiter & Schmidhuber, 1997) are variations of recurrent neural networks that are today the most effective networks for sequence modeling and are the foundation of many practical applications. In a dynamic network, the weights are called the *long-term memory*, and the feedbacks are the *short-term memory*.

In essence, only the short-term memory (i.e., feedbacks, or previous events) provides a network with the context. In a typical RNN, the information on the short-term memory is continuously replaced as new information is fed back into the network over time. That is why RNNs perform well when the gap between the relevant information and the place that is needed is small. For instance, for predicting the last word in the sentence "The referee blew his whistle," we just need to know an earlier word in the sentence (i.e., *referee*) to correctly predict the final word. In this case the gap between the relevant information (i.e., *referee*) and where it is needed (i.e., to predict *whistle*) is

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

small, and an RNN could easily perform this learning and prediction task.

However, sometimes the relevant information required to perform a task is far away from where it is needed (i.e., the gap is large). In such situations, it is quite likely that it would have already been replaced by other information in the short-term memory by the time it is needed for the creation of the proper context. For instance, to predict the last word in "I went to a car wash yesterday. It cost $5 to wash my *car*," there is a relatively larger gap between the relevant information (i.e., *car wash*) and where it is needed (i.e., to predict *car*). Sometimes we may even need to refer to the previous paragraphs to reach the relevant information for predicting the true meaning of a word. In such cases, RNNs usually do not perform well because they cannot keep the information in their short-term memory for a long enough time. Fortunately, LSTM networks do not have such a shortcoming. The term *long short-term memory network* refers to a network that is used to remember what happened in the past (i.e., feedbacks, or previous outputs of the layers) for a long enough time that it can be used/leveraged in accomplishing the task when needed.

From an architectural viewpoint, the memory concept (i.e., remembering "what happened in the past") is incorporated in LSTM networks by incorporating four additional layers into the typical recurrent network architecture: three gate layers—*input gate*, *forget* (a.k.a. *feedback*) *gate*, and *output gate*—and an additional layer called the *Constant Error Carousel* (*CEC*), also known as the *state unit*, which integrates those gates and allows them to interact with the other layers. Each gate is simply a layer with two inputs: one from the network inputs and the other a feedback from the final output of the whole network. The gates involve log-sigmoid transfer functions. Therefore, their outputs will be between 0 and 1 and will describe how much of each component (either input, feedback, or output) should be let through the network. Also, CEC is a layer that falls between the input and the output layers in a recurrent network architecture and applies the gate outputs to make the short-term memory long.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

To have a long short-term memory means that we want to keep the effect of previous outputs for a longer time. However, we typically do not want to indiscriminately remember everything that has happened in the past. Therefore, gating provides us with the capability of remembering prior outputs selectively. The input gate allows selective inputs to the CEC; the forget gate clears the CEC from the unwanted previous feedbacks; and the output gate allows selective outputs from the CEC. Figure 9.21 shows a simple depiction of a typical LSTM architecture.

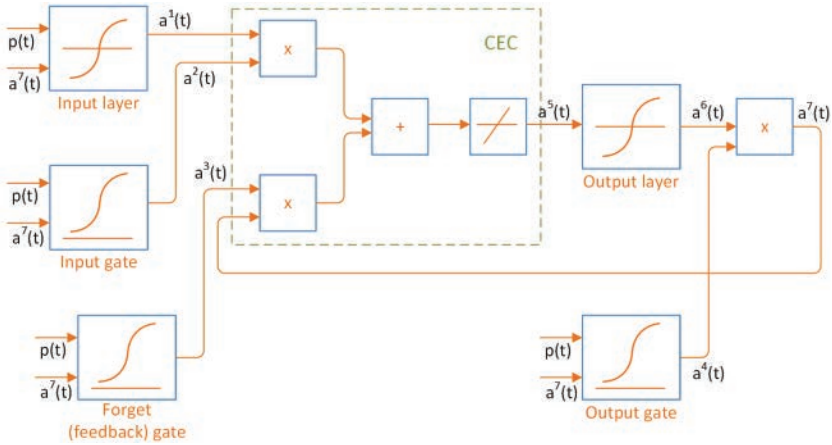


Figure 9.21 Typical LSTM Network Architecture

In summary, the gates in the LSTM network control the flow of information through the network and dynamically change the time scale of integration based on the input sequence. As a result, LSTM networks are able to learn long-term dependencies among the sequence of inputs more easily than are regular RNNs.

LSTM Network Applications

Since their emergence in the late 1990s (Hochreiter & Schmidhuber, 1997), LSTM networks have been widely used in many sequence modeling applications, including image captioning (i.e., automatically

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

describing the content of images) (Vinyals et al., 2015, 2017), handwriting recognition and generation (Graves, 2013; Keysers et al., 2017), parsing (Liang et al., 2016), speech recognition (Graves & Jaitly, 2014), and machine translation (Bahdanau et al., 2014; Sutskever et al., 2014).

Currently, we are surrounded by multiple deep learning solutions working on the basis of speech recognition, such as Apple's Siri, Google's Now, Microsoft's Cortana, and Amazon's Alexa, several of which many people deal with on a daily basis to check on the weather, ask for a web search, call a friend, or ask for directions. Note taking is not a difficult, frustrating task anymore because we can easily record a speech or lecture, upload the digital recording on one of the several cloud-based speech-to-text service providers' platforms, and download the transcript in a few seconds. The Google cloud-based speech-to-text service, for example, supports 120 languages and their variants and has the ability to convert speech to text either in real time or using recorded audios. The Google service automatically handles the noise in the audio; accurately punctuates the transcripts with commas, question marks, and periods; and can be customized by the user to a specific context by getting a set of terms and phrases that are very likely to be used in a speech and recognizing them appropriately.

In 2014, Microsoft launched its Skype Translator service, a free voice translation service involving both speech recognition and machine translation; it has the ability to translate real-time conversations in 10 languages. Using this service, people who speak different languages can talk to each other in their own languages via a Skype voice or video call, and the system recognizes their voices and translates every sentence spoken by using a translator bot in near real time for the other party. To provide more accurate translations, the deep networks used in the back end of this system were trained using conversational language (i.e., using materials such as translated web pages, movie subtitles, and casual phrases taken from people's conversations on social networking websites) rather than the formal

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

language commonly used in documents. The output of the speech recognition module of the system then goes through TrueText, a Microsoft technology for normalizing text that is capable of identifying mistakes and disfluencies (e.g., pauses during the speech, or repetitions of some parts of the speech, or fillers like "um" and "ah" when speaking) and accounting for them in order to make better translations. Figure 9.22 shows the four-step process involved in the Skype Translator by Microsoft, each of which relies on the LSTM type of deep neural networks.

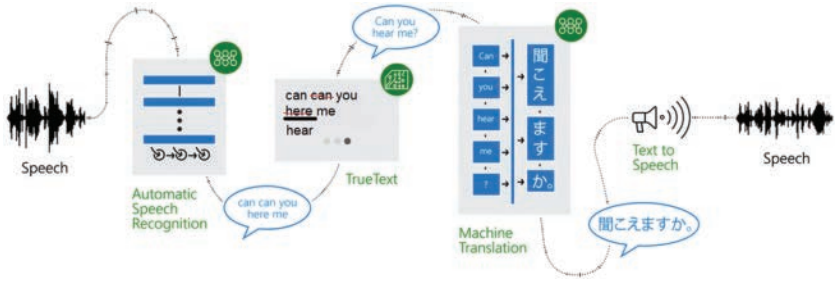


Figure 9.22 Four-Step Process of Translating Speech Using Deep Networks in Microsoft Skype Translator

Computer Frameworks for Implementation of Deep Learning

To a great extent, deep learning owes its recent popularity to advances in the software and hardware infrastructure required for the implementation of deep learning. In the past few decades, GPUs have been revolutionized to support the playing of high-resolution videos as well as advanced video games and virtual reality applications. However, GPUs' huge processing potential had not been effectively utilized for purposes other than graphics processing up until a few years ago. Software libraries such as Theano (Bergstra et al., 2010), Torch (Collobert et al., 2011), Caffe (Jia et al., 2014), PyLearn2 (Goodfellow et al., 2013), TensorFlow (Abadi et al., 2016), and MXNet (Chen

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

et al., 2015) were developed to program GPUs for general-purpose processing and particularly for deep learning and analysis of big data. GPUs have become a critical enabler of modern analytics. The operation of these libraries mostly relies on a parallel computing platform and application programming interface (API) developed by NVIDIA called *Compute Unified Device Architecture* (*CUDA*), which enables software developers to use GPUs made by NVIDIA for general-purpose processing. In fact, each deep learning framework consists of a high-level scripting language (e.g., Python, R, Lua) and a library of deep learning routines usually written in C (for using CPUs) or CUDA (for using GPUs).

The following sections introduce some of the most popular software libraries used for deep learning, including Torch, Caffe, TensorFlow, Theano, and Keras, and discuss some of their specific properties.

Torch

Torch (Collobert et al., 2011) is an open source scientific computing framework (available at www.torch.ch) for implementing machine learning algorithms using GPUs. The Torch framework is a library based on LuaJIT, a compiled version of the popular Lua programming language (www.lua.org). Torch adds a number of valuable features to Lua that make deep learning analyses possible; Whereas Lua typically uses only tables (i.e., two-dimensional arrays) for data structuring, Torch enables support for *n*-dimensional arrays (i.e., tensors). In addition, Torch includes routine libraries for manipulating (i.e., indexing, slicing, transposing) tensors, linear algebra, neural network functions, and optimization. More importantly, while Lua by default uses CPUs to run programs, Torch enables use of GPUs for running programs written in the Lua language.

The easy and extremely fast scripting properties of LuaJIT along with its flexibility have made Torch a very popular framework for practical deep learning applications such that today its latest version,

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Torch7, is widely used by a number of big companies in the deep learning area, including Facebook, Google, and IBM in their research labs as well as for their commercial applications.

Caffe

Caffe is another open source deep learning framework (available at http://caffe.berkeleyvision.org). It was created by Yangqing Jia (2013), a PhD student at the University of California, Berkeley, which Berkeley AI Research (BAIR) then further developed. Caffe has multiple options for use as a high-level scripting language, including the command line, Python, and MATLAB interfaces. The deep learning libraries in Caffe are written in the C++ programming language.

In Caffe, everything is done using text files instead of code. For example, to implement a network, generally you need to prepare two text files with the .prototxt extension that are communicated by the Caffe engine via JavaScript Object Notation (JSON) format. The first text file, known as the *architecture* file, defines the architecture of the network, layer by layer, with each layer defined by a name, a type (e.g., data, convolution, output), the names of the previous (bottom) and next (top) layers in the architecture, and some required parameters (e.g., kernel size and stride for a convolutional layer). The second text file, known as the *solver* file, specifies the properties of the training algorithm, including the learning rate, the maximum number of iterations, and the processing unit (CPU or GPU) to be used for training the network.

While Caffe supports multiple types of deep network architectures, such as CNN and LSTM networks, it is particularly known as an efficient framework for image processing due to its incredible speed in processing image files. According to its developers, it is able to process more than 60 million images per day (i.e., 1 ms/image) using a single NVIDIA K40 GPU. In 2017, Facebook released an improved version of Caffe called Caffe2 (www.caffe2.ai), and then joined forces

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

to create a research and production platform called PyTorch with the aim of improving the original framework to be effectively used for deep learning architectures, with a special emphasis on portability for performing cloud and mobile computations while maintaining scalability and performance.

TensorFlow

Another popular open source deep learning framework is TensorFlow. It was originally developed and written in Python and C++ by the Google Brain Group in 2011 as DistBelief, and it was further developed into TensorFlow in 2015. TensorFlow is at this time the only deep learning framework that, in addition to supporting CPUs and GPUs, supports tensor processing units (TPUs). Google created TPUs in 2016 for the specific purpose of neural network machine learning (in particular, the TensorFlow framework).

Although Google has not yet made TPUs available to the market, it has reportedly used them in a number of its commercial services, such as Google Search, Street View, Google Photos, and Google Translate, and significant improvements have been reported. A detailed study performed by Google shows that TPUs deliver 30 to 80 times higher performance per watt than contemporary CPUs and GPUs (Sato et al., 2017). For example, Ung (2016) reported that in Google Photos, an individual TPU can process more than 100 million images per day (i.e., 0.86 ms/image). As soon as Google makes TPUs commercially available, this unique feature will probably put TensorFlow way ahead of the other alternative frameworks.

Another interesting feature of TensorFlow is its visualization module, TensorBoard. Implementing a deep neural network is a complex and confusing task. TensorBoard is a web application that involves a handful of tools for visualizing network graphs and plotting quantitative network metrics. It helps users better understand what is going on during training procedures and debug possible issues.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Theano

In 2007, the Deep Learning Group at the University of Montreal developed the initial version of the Theano Python library (http://deeplearning.net/software/theano) to define, optimize, and evaluate mathematical expressions involving multidimensional arrays (i.e., tensors) on CPU or GPU platforms. Theano was one of the first deep learning frameworks, and it became a source of inspiration for the developers of TensorFlow. In both Theano and TensorFlow, a typical network implementation involves two sections: In the first section, a computational graph is built by defining the network variables and operations to be done on them, and the second section runs that graph. (Theano compiles a graph into a *function*, and TensorFlow creates a *session*.) In fact, what happens in these libraries is that the user defines the structure of the network by providing some simple and symbolic syntax that is understandable even for beginners in programming, and the library automatically generates appropriate code in either C (for CPU processing) or CUDA (for GPU processing) to implement the defined network. Hence, users who have no knowledge of programming in C or CUDA and who have minimal knowledge of Python are able to efficiently design and implement deep learning networks.

Theano also includes some built-in functions for visualizing computational graphs as well as plotting network performance metrics, even though its visualization features are not comparable to those of TensorBoard.

Keras: An Application Programming Interface

For all of the previously described deep learning frameworks, users must be familiar with the particular syntax of the framework in order to be able to successfully train a network. Keras (https://keras.io/) is easier and more user-friendly. Keras is an open source neural network library written in Python that functions as a high-level application

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

programming interface (API) and is able to run on top of various deep learning frameworks, including Theano and TensorFlow. In essence, with Keras you provide the key properties of network building blocks (i.e., type of layers, transfer functions, and optimizers) via extremely simple syntax. Keras then automatically generates syntax in one of the deep learning frameworks and runs that framework in the background. While Keras is efficient enough to build and run general deep learning models in just a few minutes, it does not provide several advanced operations that are provided by TensorFlow or Theano. Therefore, in dealing with special deep network models that require advanced settings, you still need to directly use one of those frameworks instead of Keras (or another API, such as Lasagne) as a proxy.

Cognitive Computing

We are witnessing a significant increase in the way technology is evolving. Things that once took decades are now taking months, and the things that we used to see only in sci-fi movies are becoming reality, one after another. In the next decade or two, technological advancements will transform how people live, learn, and work in a rather dramatic fashion. The interactions between humans and technology will become intuitive, seamless, and perhaps transparent. Cognitive computing will play a significant role in this transformation. Generally speaking, *cognitive computing* refers to computing systems that use mathematical models to emulate (or partially simulate) the human cognition process to find solutions to complex problems and situations where the potential answers can be imprecise. While the term *cognitive computing* is often used interchangeably with AI and smart search engines, the phrase itself is closely associated with IBM's cognitive computing system Watson and its success on the television show *Jeopardy!* (see Chapter 1, "Introduction to Analytics").

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

According to the Cognitive Computing Consortium (2018), cognitive computing makes a new class of problems computable. It addresses highly complex situations that are characterized by ambiguity and uncertainty; in other words, it handles the kinds of problems that are thought to be solvable with human ingenuity and creativity. In today's dynamic, information-rich, and unstable situations, data tends to change frequently, and it often conflicts with other data. The goals of users evolve as they learn more and redefine their objectives. To respond to the fluid nature of users' understanding of their problems, a cognitive computing system offers a synthesis not just of information sources but also of influences, contexts, and insights. To achieve such a high level of performance, cognitive systems often need to weigh conflicting evidence and suggest an answer that is "best" rather than "right." Figure 9.23 illustrates a general framework for cognitive computing in which data and AI technologies are used to solve complex real-world problem.



Figure 9.23 Conceptual Framework for Cognitive Computing and Its Promises

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

How Does Cognitive Computing Work?

As you would guess from the name, cognitive computing works much like a human thought process, reasoning mechanism, and cognitive system. These cutting-edge computation systems can find and synthesize data from various information sources and weigh context and conflicting evidence inherent in the data to provide the best possible answers to a given question or problem. To achieve this, cognitive systems include self-learning technologies that use data mining, pattern recognition, deep learning, and natural language processing (NLP) to mimic the way the human brain works.

Using computer systems to solve the types of problems that humans are typically tasked with requires vast amounts of structured and unstructured data fed to machine learning algorithms. Over time, cognitive systems are able to refine the way in which they learn and recognize patterns and the way they process data to become capable of anticipating new problems and modeling and proposing possible solutions.

To achieve those capabilities, cognitive computing systems must have the following key attributes, as defined by the Cognitive Computing Consortium (2018):

- **Adaptive.** Cognitive systems must be flexible enough to learn as information changes and goals evolve. The systems must be able to digest dynamic data in real time and make adjustments as the data and environment change.
- **Interactive.** Human–computer interaction (HCI) is a critical component of cognitive systems. Users must be able to interact with cognitive machines and define their needs as those needs change. The technologies must also be able to interact with other processors, devices, and cloud platforms.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

- **Iterative and stated problem.** Cognitive computing technologies can identify problems by asking questions or pulling in additional data if a stated problem is vague or incomplete. The systems do this by maintaining information about similar situations that have previously occurred.
- **Contextual.** Understanding context is critical in thought processes, so cognitive systems must understand, identify, and mine contextual data, such as syntax, time, location, domain, requirements, and a specific user's profile, tasks, or goals. Cognitive systems may draw on multiple sources of information, including structured and unstructured data and visual, auditory, or sensor data.

How Does Cognitive Computing Differ from AI?

The term *cognitive computing* is often used interchangeably with *AI*, the umbrella term for technologies that rely on data and scientific methods/computations to make (or help in making) decisions. But there are differences between the two terms, largely in terms of their purposes and applications. AI technologies include—but are not limited to—machine learning, neural computing, NLP, and, most recently, deep learning. With AI systems, especially in machine learning systems, data are fed into the algorithm for processing (an iterative and time-demanding process that is often called *training*) so that the systems "learn" variables and interrelationships among those variables so that they can produce predictions (or characterizations) about a given complex problem or situation. Applications based on AI and cognitive computing include intelligent assistants, such as Amazon's Alexa, Google Home, and Apple's Siri. Table 9.3 provides a simple comparison between cognitive computing and AI (CCC, 2018; Reynolds & Feldman, 2014).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Table 9.3 Cognitive Computing Versus Artificial Intelligence (AI)

Characteristic	Cognitive Computing	Artificial Intelligence (AI)
Technologies used	Machine learning Natural language processing Neural networks Deep learning Text mining Sentiment analysis	Machine learning Natural language processing Neural networks Deep leaning
Capabilities offered	Simulate human thought processes to assist humans in finding solutions to complex problems	Find hidden patterns in a variety of data sources to identify problems and provide potential solutions
Purpose	Augment human capability	Automate complex processes by acting like a human in certain situations
Industries	Customer service, marketing, health care, entertainment, service sector	Manufacturing, finance, health care, banking, securities, retail, government

As can be seen in Table 9.3, the differences between AI and cognitive computing are rather minimal. This is expected because cognitive computing is often characterized as a subcomponent of AI or an application of AI technologies tailored for a specific purpose. AI and cognitive computing both use similar technologies and are applied to similar industry segments and verticals. The main difference between the two is the purpose: Whereas cognitive computing is aimed at helping humans solve complex problems, AI is aimed at automating processes that are performed by humans; at the extreme, AI strives to replace humans with machines for tasks requiring "intelligence," one at a time.

In recent years, cognitive computing typically has been used to describe AI systems that aim to simulate human thought processes. Human cognition involves real-time analysis of environment, context, and intent among many other variables that inform a person's ability to solve problems. A number of AI technologies are required for a computer system to build cognitive models that mimic human

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

thought processes, including machine learning, deep learning, neural networks, NLP, text mining, and sentiment analysis.

In general, cognitive computing is used to assist humans in their decision-making process. Some examples of cognitive computing applications include supporting medical doctors in their treatment of disease. IBM Watson for Oncology, for example, has been used at Memorial Sloan Kettering Cancer Center to provide oncologists evidence-based treatment options for cancer patients. When medical staff input questions, Watson generates a list of hypotheses and offers treatment options for doctors to consider. Whereas AI relies on algorithms to solve a problem or to identify patterns hidden in data, cognitive computing systems have the loftier goal of creating algorithms that mimic the human brain's reasoning process to help humans solve an array of problems as the data and the problems constantly change.

In dealing with complex situations, context is important, and cognitive computing systems make context computable. They identify and extract context features—such as time, location, task, history, or profile—to present a specific set of information that is appropriate for an individual or for a dependent application engaged in a specific process at a specific time and place. According to the Cognitive Computing Consortium, these systems provide machine-aided serendipity by wading through massive collections of diverse information to find patterns and then apply those patterns to respond to the needs of the user at a particular moment. In a sense, cognitive computing systems aim to redefine the nature of the relationship between people and their increasingly pervasive digital environment. They may play the role of assistant or coach for the user, and they may act virtually autonomously in many problem-solving situations. The boundaries of the processes and domains these systems can affect are still elastic and emergent. Their output may be prescriptive, suggestive, instructive, or simply entertaining.

In the short time of its existence, cognitive computing has proved to be useful in many domains and in a variety of complex situations,

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

and it is evolving into many more. The typical use cases for cognitive computing include the following:

- Development of smart and adaptive search engines
- Effective use of natural language processing
- Speech recognition
- Language translation
- Context-based sentiment analysis
- Face recognition and facial emotion detection
- Risk assessment and mitigation
- Fraud detection and mitigation
- Behavioral assessment and recommendations

Cognitive analytics is a term that refers to cognitive computing–branded technology platforms, such as IBM Watson, that specialize in processing and analyzing large, unstructured data sets. Typically, word processing documents, emails, videos, images, audio files, presentations, web pages, social media, and many other data formats need to be manually tagged with metadata before the data can be fed into a traditional analytics engine and big data tools for computational analyses and insight generation. The principal benefit of using cognitive analytics over those traditional big data analytics tools is that for cognitive analytics, such data sets do not need to be pretagged. Cognitive analytics systems can use machine learning to adapt to different contexts with minimal human supervision. Such a system can be equipped with a chatbot or search assistant that understands queries, explains data insights, and interacts with humans in human languages.

Cognitive Search

Cognitive search is the new-generation search method that uses AI (i.e., advanced indexing, NLP, and machine learning) to return results that are much more relevant to users. Forrester defines

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

cognitive search and knowledge discovery solutions as "a new generation of enterprise search solutions that employ AI technologies such as natural language processing and machine learning to ingest, understand, organize, and query digital content from multiple data sources" (Gualtieri, 2017). Cognitive search creates searchable information out of nonsearchable content by leveraging cognitive computing algorithms to create an indexing platform.

Searching for information is a tedious task. Although current search engines do a very good job finding relevant information in a timely manner, their sources are limited to publicly available data over the Internet. Cognitive search proposes the next generation of search, tailored for use in enterprises. Cognitive search is different from traditional search because, according to Gualtieri (2017):

- **It can handle a variety of data types.** Search is no longer just about unstructured text contained in documents and in web pages. Cognitive search solutions can also accommodate structured data contained in databases and even nontraditional enterprise data, such as images, video, audio, and machine/sensor generated logs from IoT devices.
- **It can contextualize the search space.** In information retrieval, the context is important. Context takes the traditional syntax-/symbol-driven search to a new level, where data is defined by semantics and meaning.
- **It uses advanced AI technologies.** The distinguishing characteristic of cognitive search solutions is that they use NLP and machine learning to understand and organize data, predict the intent of a search query, improve the relevancy of results, and automatically tune the relevancy of results over time.
- **It enables developers to build enterprise-specific search applications.** Search is not just about a text box on an enterprise portal. Enterprises build search applications that embed search in customer 360 applications, pharmaceutical research tools, and many other business process applications. Virtual

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

digital assistants such as Amazon Alexa, Google Now, and Siri would be useless without powerful searches behind the scenes. Enterprises wishing to build similar applications for their customers will also benefit from cognitive search solutions. Cognitive search solutions provide software development kits (SDKs), APIs, and/or visual design tools that allow developers to embed the power of the search engine in other applications.

Figure 9.24 shows the progressive evolution of search methods from good old keyword searches to modern-day cognitive searches on two dimensions: ease of use and value proposition.

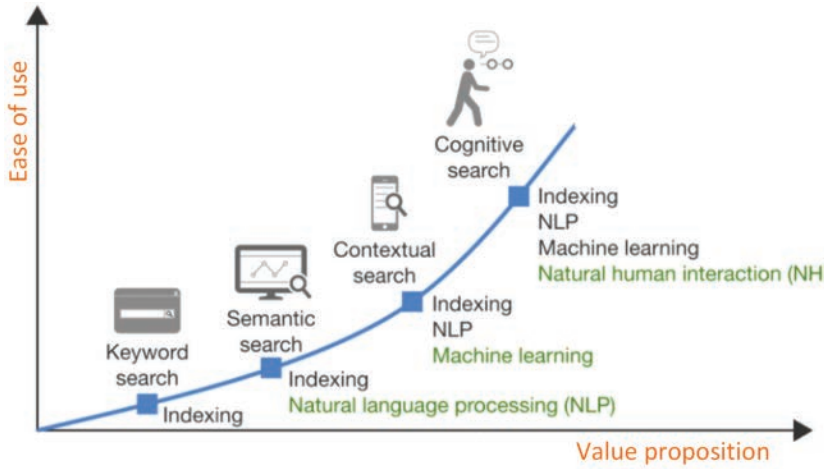


Figure 9.24 Progressive Evolution of Search Methods

Application Case: Fighting Fraud with Deep Learning

Danske Bank is a Scandinavian bank with strong local roots and bridges to the rest of the world. Founded in October 1871, Danske Bank has helped people and businesses in Scandinavia realize their ambitions for over 145 years. Its headquarters is in Denmark, and its core markets are Denmark, Finland, Norway, and Sweden. The compete case study on which this example is based, as well as related documents and videos, can be found in Teradata (2020).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The Business Problem

Mitigating fraud is a top priority for banks. According to the Association of Certified Fraud Examiners, businesses lose more than $3.5 trillion each year to fraud (https://www.pymnts.com/fraud-attack/2016/fraud-costs-security-hackers/). The problem is pervasive across the financial industry and is becoming more prevalent and sophisticated each month. As customers conduct more banking online across a wider variety of channels and devices, there are more opportunities for fraud to occur. Adding to the problem, fraudsters are becoming more creative and technologically savvy: They are using advanced technologies such as machine learning, and new schemes to defraud banks are evolving rapidly.

Old methods for identifying fraud, such as using human-written rule engines, catch only a small percentage of fraud cases and produce a significantly high number of false positives. While false negatives end up costing the bank money, chasing after a large number of false positives not only costs time and money but also damages customer trust and satisfaction. To improve probability predictions and identify a much higher percentage of actual cases of fraud while reducing false alarms, banks need new forms of analytics, including the use of artificial intelligence.

Danske Bank, like other global banks, is seeing a seismic shift in customer interactions. In the past, most customers handled their transactions in bank branches. Today, almost all interactions take place digitally through a mobile phone, a tablet, an ATM, or a call center. This provides more "surface area" for fraud to occur. The bank needed to modernize its fraud detection defenses as it had a low 40% fraud detection rate. Danske Bank was managing up to 1,200 false positives per day, and 99.5% of all cases the bank

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

was investigating were not fraud related. That large number of false alarms required a substantial investment of people, time, and money to investigate what turned out to be dead ends. Working with Think Big Analytics, a Teradata company, Danske Bank made a strategic decision to apply innovative analytic techniques, including AI, to better identify instances of fraud while reducing false positives.

The Solution: Deep Learning Enhances Fraud Detection

Danske Bank integrated deep learning with GPU appliances that were also optimized for deep learning. The new software system helps the analytics team to identify potential cases of fraud while intelligently avoiding false positives. Operational decisions are shifted from users to AI systems. However, human intervention is still necessary in some cases. For example, the model can identify anomalies, such as debit card purchases taking place around the world, but analysts are needed to determine whether that is fraud or a bank customer simply made an online purchase that sent a payment to China and then bought an item the next day from a retailer based in London.

Danske Bank's analytic approach employs a "champion/challenger" methodology. With this approach, deep learning systems compare models in real time to determine which one is most effective. Each challenger processes data in real time, learning as it goes which traits are most likely to indicate fraud. If a process dips below a certain threshold, the model is fed more data, such as the geolocation of customers or recent ATM transactions. When a challenger outperforms other challengers, it transforms into a champion, giving the other models a roadmap to successful fraud detection.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

The Results

Danske Bank implemented a modern enterprise analytic solution leveraging AI and deep learning, and this solution has paid big dividends:

- The bank realized a 60% reduction in false positives and expects to reach an 80% reduction.
- The bank increased true positives by 50%.
- The bank can now focus resources on actual cases of fraud.

Figure 9.25 shows how Danske Bank's true and false positive rates improved with advanced analytics (including deep learning). The red dot represents the old rules engine, which caught only about 40% of all fraud. Deep learning improved significantly on machine learning, allowing Danske Bank to better detect fraud with a much lower false positive rate.

Enterprise analytics is rapidly evolving and moving into new learning systems enabled by AI. At the same time, hardware and processors are becoming more powerful and specialized, and algorithms, including open source algorithms, are becoming more accessible. All these advances give banks the powerful solutions needed to identify and mitigate fraud. As Danske Bank learned, building and deploying an enterprise-grade analytics solution that meets its specific needs and leverages its data sources delivers more value than traditional off-the-shelf tools could have provided. With AI and deep learning, Danske Bank now has the ability to better uncover fraud without being burdened by an unacceptable number of false positives. The solution also allows the bank's engineers, data scientists, lines of business, and investigative officers from Interpol, local police, and other agencies to collaborate to uncover fraud, including sophisticated fraud rings. With its enhanced capabilities, the enterprise analytic solution is now being used across other business areas of the bank to deliver additional value.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

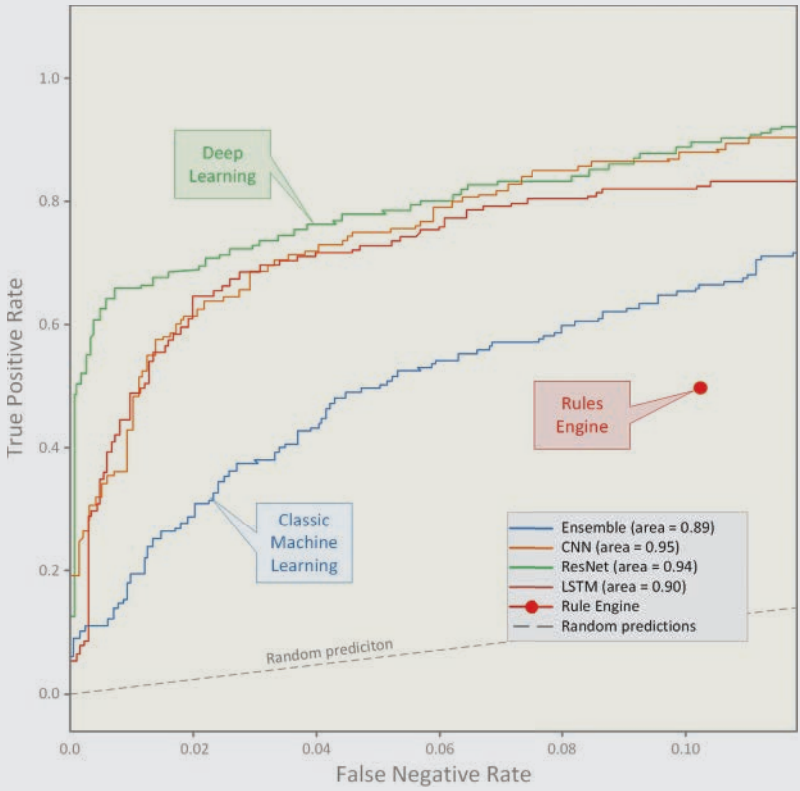


Figure 9.25 Deep Learning Improved Both True Positives and True Negatives

Because these technologies are still evolving, implementing deep learning and AI solutions can be difficult for companies to achieve on their own. They can benefit by partnering with a company that has the proven capabilities to implement technology-enabled solutions that deliver high-value outcomes. As shown in this case, Think Big Analytics has the expertise to configure specialized hardware and software frameworks to enable new operational processes. The Danske Bank project entailed integrating open source solutions and deploying production models and then applying deep learning analytics to extend and improve the models. A framework was created to manage and track the models in the production system and to

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

make sure the models could be trusted. These models enabled the underlying system to make autonomous decisions in real time that aligned with the bank's procedural, security, and high-availability guidelines. The solution provided new levels of detail, such as time series and sequences of events, to better assist the bank with its fraud investigations. The entire solution was implemented very quickly—from kickoff to going-live in only five months. Figure 9.26 shows a generalized framework for AI and deep learning–based enterprise-level analytics solutions.



Figure 9.26 A Generalized Framework for AI and Deep Learning–Based Analytics

Summary

Deep learning is a contemporary application of artificial intelligence and machine learning. Large collections of unstructured data sources (i.e., big data) along with a new generation of neural network architectures set the stage for deep learning and the unprecedented promises it brought. The largest data-driven companies, including Google, Facebook, Twitter, LinkedIn, Apple, and Microsoft, are investing millions of dollars into deep learning and related technologies, and many of them are making their findings publicly accessible to foster community-based crowdsourced development projects.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

This chapter starts with a clear definition of what deep learning is and where it came from. It discusses artificial neural networks and covers fundamental concepts of deep learning. It discusses two popular deep learning architectures, CNN and RNN/LSTM, and related community projects and freely accessible class libraries. The chapter also covers cognitive computing and its concepts and applications.

References

Abadi, M., P. Barham, J. Chen, Z. Chen, et al. (2016). "TensorFlow: A System for Large-Scale Machine Learning." *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, 16: 265–283.

Bahdanau, D., K. Cho, & Y. Bengio. (2014). "Neural Machine Translation by Jointly Learning to Align and Translate," *ArXiv*, 1409.0473.

Bengio, Y. (2009). "Learning Deep Architectures for AI," *Foundations and Trends in Machine Learning*, 2(1): 1–127.

Bergstra, J., O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, et al. (2010). "Theano: A CPU and GPU Math Compiler in Python," *Proceedings of the Ninth Python in Science Conference*, 1: 3–10.

Chen, T., M. Li, Y. Li, M. Lin, et al. (2015). "Mxnet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems," *ArXiv*, 1512.01274.

Cognitive Computing Consortium. (2018). *Cognitive Computing Defined*. https://cognitivecomputingconsortium.com/resources/cognitive-computing-defined/#1467829079735-c0934399-599a (accessed September 2020).

Collobert, R., K. Kavukcuoglu, & C. Farabet. (2011). "Torch7: A MATLAB-Like Environment for Machine Learning." BigLearn, NIPS workshop.

Cybenko, G. (1989). "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals and Systems*, 2(4): 303–314.

Goodfellow, I., Y. Bengio, & A. Courville. (2016). *Deep Learning*. MIT Press.

Goodfellow, I. J., D. Warde-Farley, P. Lamblin, V. Dumoulin, et al. (2013). "Pylearn2: A Machine Learning Research Library," *ArXiv*, 1309.4214.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Graves, A. (2013). "Generating Sequences with Recurrent Neural Networks," *ArXiv*, 1309.0850.

Graves, A., & N. Jaitly. (2014). "Towards End-to-End Speech Recognition with Recurrent Neural Networks," *Proceedings on International Conference on Machine Learning*, 32(2): 1764–1772.

Gualtieri, M. (2017). *Cognitive Search Is the AI Version of Enterprise Search*. go.forrester.com/blogs/17-06-12-cognitive_search_is_the_ai_version_of_enterprise_search/ (accessed September 2020).

Haykin, S. S. (2009). *Neural Networks and Learning Machines*, 3rd ed. Prentice Hall.

He, K., X. Zhang, S. Ren, & J. Sun. (2015). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034.

Hinton, G. E., S. Osindero, & Y.-W. Teh. (2006). "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, 18(7): 1527–1554.

Hochreiter, S., & J. Schmidhuber. (1997). "Long Short-Term Memory," *Neural Computation*, 9(8), 1735–1780.

Hornik, K. (1991). "Approximation Capabilities of Multilayer Feedforward Networks," *Neural Networks*, 4(2): 251–257.

Jia, Y. (2013). "Caffe: An Open Source Convolutional Architecture for Fast Feature Embedding." *Proceedings of the 2014 ACM Conference on Multimedia*, doi:10.1145/2647868.2654889.

Keysers, D., T. Deselaers, H. A. Rowley, L.-L. Wang, & V. Carbune. (2017). "Multi-Language Online Handwriting Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6): 1180–1194.

Krizhevsky, A., I. Sutskever, & G. Hinton. (2012). "ImageNet Classification with Deep Convolutional Neural Networks," *Neural Information Processing Systems*, 25: 1097–1105.

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, et al. (1989). "Backpropagation Applied to Handwritten ZIP Code Recognition," *Neural Computation*, 1(4): 541–551.

Liang, X., X. Shen, J. Feng, L. Lin, & S. Yan. (2016). "Semantic Object Parsing with Graph LSTM," *European Conference on Computer Vision*. Springer, pp. 125–143.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Mahajan, D., R. Girshick, V. Ramanathan, M. Paluri, & L. van der Maaten. (2018). *Advancing State-of-the-Art Image Recognition with Deep Learning on Hashtags*. https://code.facebook.com/posts/1700437286678763/advancing-state-of-the-art-image-recognition-with-deep-learning-on-hashtags/.

Mikolov, T., K. Chen, G. Corrado, & J. Dean. (2013a). "Efficient Estimation of Word Representations in Vector Space," *ArXiv*, 1301.3781.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, & J. Dean. (2013b). "Distributed Representations of Words and Phrases and Their Compositionality," *Advances in Neural Information Processing Systems*, 26: 3111–3119.

Principe, J. C., N. R. Euliano, & W. C. Lefebvre. (2000). *Neural and Adaptive Systems: Fundamentals Through Simulations*. Wiley.

Reynolds, H., & S. Feldman. (2014, July/August). "Cognitive Computing: Beyond the Hype," *KM World*, 23(7): 21.

Rumelhart, D. E., G. E Hinton, & R. J. Williams. (1986). "Learning Representations by Back-Propagating Errors," *Nature*, 323(6088): 533.

Russakovsky, O., J. Deng, H. Su, J. Krause, et al. (2015). "Imagenet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, 115(3): 211–252.

Sato, K., C. Young, & D. Patterson. (2017). *An In-Depth Look at Google's First Tensor Processing Unit (TPU)*. https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu (accessed September 2020).

Sutskever, I., O. Vinyals, & Q. V. Le. (2014). "Sequence to Sequence Learning with Neural Networks," *Advances in Neural Information Processing Systems*, 4: 3104–3112.

Teradata. (2020). *Danske Bank Fights Fraud with Deep Learning and AI*. https://www.teradata.com/Resources/Case-Studies/Danske-Bank-Fight-Fraud-With-Deep-Learning-and-AI (accessed August 2020).

Ung, G. M. (2016, May). "Google's Tensor Processing Unit Could Advance Moore's Law 7 Years into the Future," *PCWorld*. https://pcworld.com/article/3072256/google-io/googles-tensor-processing-unit-said-to-advance-moores-law-seven-years-into-the-future.html (accessed September 2020).

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Vinyals, O., A. Toshev, S. Bengio, & D. Erhan. (2015). "Show and Tell: A Neural Image Caption Generator," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), pp. 3156–3164.

Vinyals, O., A. Toshev, S. Bengio, & D. Erhan (2017). "Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge," *Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4): 652–663.

Zhou, Y.-T., R. Chellappa, A. Vaid, & B. K. Jenkins. (1988). "Image Restoration Using a Neural Network," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7): 1141–1151.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

A

KNIME and the Landscape of Tools for Business Analytics and Data Science

To keep up with the tremendous growth in the popularity of business analytics and data science that we have been experiencing in recent years (both in academia and industry), the tools used to implement these technologies have been evolving and expanding at an unprecedented pace. As shown in Figure 2.4 in Chapter 2, "Introduction to Predictive Analytics and Data Mining," there are numerous software tools, ranging from commercial/paid to free/open source, from graphical workflow-type to language-based programming platforms. Some of these tools are cloud based, some are individual local installations, and some are both, providing local tools and complementary cloud-based platforms for computational enhanced efficiency and ease of model deployment. Some of these tools are good at data preprocessing and/or flexible and powerful model building, and others are superior in terms of ease of learning, use, and deployment. As evidenced from the continually expending landscape of tool offerings, there is no one tool that is best at everything. Most business analytics and data science professionals tend to use a portfolio of tools and programming languages collectively to produce the best possible outcomes for their data analytics projects.

The success of data scientists at work depends largely on the tools they rely on. A data scientist needs to have mathematical and statistical knowledge of the underlying algorithms, experience with the most effective analytics platforms, domain knowledge/wisdom, and team/people management skills. However, there are other more contingent

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

and complementary factors that also influence the final success and the impression left by a data science project on the stakeholders.

Clearly, every project has a deadline, a list of requirements, and a budget to meet. Most data mining projects are designed so stringently that there is not much room for sandboxing or an extended trial-and-error approach. A data scientist is expected to identify and implement a solution in a short time and make sure it is implemented correctly—that is, that it meets the project requirements. To do so, the data scientist may need to quickly experiment with different techniques to identify and adopt the best possible roadmap and repeated mechanisms for the project. Of course, every project has a budget, too. The fast implementation of the right solution is often additionally constrained by a limited budget.

Some projects are quite complex and require specialized algorithms beyond just the classic general machine learning algorithms. Sometimes data scientists are forced to learn new techniques and new algorithms on the spot and quickly to meet the deadline. The more time a data scientist has to dedicate to theory and math, the more quickly the learning will proceed.

Finally, the passage from prototype to production must be as fast (again) and as secure as possible. A data scientist cannot degrade the parts of the solution while moving it into production or risk exposing it to less secure execution, considering that the solution will be consumed by a different set of less expert users. If and when required, a different set of solutions—with more or less interaction and sophistication and with a more secure framework—can and should be made available for the consumption of the larger public.

Many other contingent factors—such as ease of learning, speed of prototype implementation, debugging and testing options to ensure the correctness of the solution, flexibility to experiment with different approaches, availability of help from external contributors and experts, and automation and security capabilities—contribute to the success of the data science project beyond the math, the experience, and the

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

domain knowledge. All these contingent factors depend heavily on the tool and the data science platform the data scientists choose to use.

I have used many of the analytics platforms, commercial and free/open source for educational (teaching business analytics data science courses at every level, undergraduate, masters, doctoral, and executive education) and commercial (consulting to for-profit companies, nonprofit organization, and government agencies) purposes in my 30-plus years in the field. Although I still use a variety of analytics tools and data science platforms, for over 6 years now, I have been using KNIME as my primary analytics platform. With KNIME, I have been able to create and deploy data science solutions for complex problems that have resulted in value-creating outcomes for my clients, high-impact publications in reputable journals, and case studies for my textbooks. One particular feature of KNIME that makes it very attractive for me is its open and expandable architecture, which enables me to incorporate the capabilities of popular data science tools and programming languages, including Python, R, Spark, and H2O, into the workflow in a seamless manner. The following sections describe the various reasons KNIME works well as an analytics platform.

Project Constraints: Time and Money

KNIME Analytics Platform is open source software for all your data needs. It is free to download from www.knime.com/downloads and free to use—not only for learning and education but for commercial use too. As a free open source product, KNIME reduces the licensing related legal headaches and the impact on the project budget.

The download, which is the full version of the software (not a restricted community version, covers a rich collection of data wrangling and machine learning techniques and is based on visual programming. Visual programming has become quite popular recently as an alternative to traditional coding. In visual programming, a graphical user interface (GUI) guides you through all the necessary steps to

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

build a pipeline (workflow) of dedicated blocks (nodes). Each node implements a given task; each workflow of nodes takes your data from the beginning to the end of the designed journey. A workflow replaces a traditional script, and a node replaces one or more script lines.

In KNIME Analytics Platform, you create nodes by dragging and dropping (or double-clicking) from the Node Repository into the workflow editor in the central part of the KNIME workbench. Node after node, in a logical order, you build the pipeline. Figure A.1 shows the KNIME Analytics Platform graphical user interface, which includes the following components (clockwise from the top-left corner):

- KNIME Explorer to store the workflows
- Workflow Coach for recommendations
- Node Repository as the node storage
- Outline for the overview
- Console for the errors
- Description to learn more
- The workflow editor (in the middle) to assemble the workflow

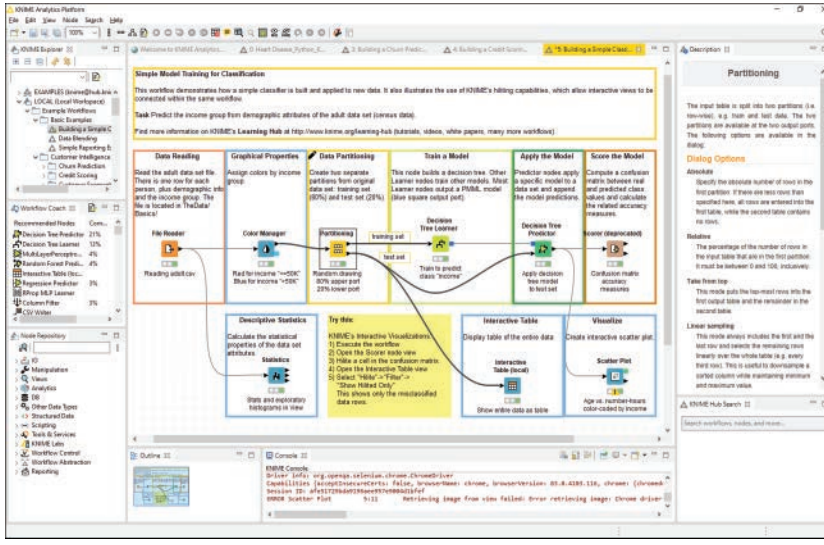


Figure A.1 The KNIME Analytics Platform

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Visual programming allows for quick prototyping, as well as ease of explaining and understanding the underlying logic of the project. It makes KNIME very easy to learn and use by less-expert consumers of business analytics. You can quickly produce a few different experimental prototypes before deciding the final direction of the project. The ease of implementation frees up time to think more deeply about possible theoretical alternatives to the current solution. I, as an instructor, like the fact that it makes my job easier to explain the complete process of converting data into actionable insight in a logical/orderly fashion with the help of an intuitive graphical workflow.

The Learning Curve

Visual programming has a less steep learning curve than code-based tools. Data science is now used in more or less all disciplines, including the humanities, languages, life sciences, economics, social sciences, and engineering. Not all scientists (or managers/decision makers) are expert programmers, and not all of them have enough spare time to become expert programmers. A GUI-based visual tool can be learned and applied in much less time than a code-based tool, freeing up precious time and resources for more important investigations, conceptualizations, and problem solving.

Also, those who are preparing to become future scientists in the humanities, languages, life sciences, economics, or other disciplines appreciate the ease of use a GUI-based tool. The idea is to shorten the time spent on syntactic details of programming in order to spend more quality time on the semantics of model building and problem solving. I have too often seen entire months dedicated to learning the syntactic nature of coding practices before any data analysis techniques are approached or implemented. With KNIME Analytics Platform, in just a few weeks, you can assemble quite capable workflows for

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

data transformation and for the training and testing of a wide variety of machine learning algorithms. That said, for uniquely complex and novel projects, a data scientist might need to incorporate some unique and advanced functionalities that exist only in analytics programming languages, such as Python and R, into the model building and problem-solving process. Because it is built on an open and extendable architecture, KNIME Analytics Platform also allows for those type of extensions through a selection of scripting and wrapper nodes.

Plenty of educational resources about KNIME Analytics Platform are available on the Internet (especially on the KNIME site, of course) to help you get going quickly. Starting from the generic LEARNING page (https://www.knime.com/resources) on the KNIME site, you can move on to courses with an instructor (https://www.knime.com/ learning/events) or completely self-taught courses (https://www.knime.com/knime-introductory-course), and you can eventually obtain certification for the platform. You may also decide to learn by yourself with the help of a book, like this one.

Help for beginners is also available at the KNIME Hub (https://hub.knime.com/), where the creators of the platform have expanded the already existing set of resource for KNIME Analytics Platform within the KNIME community.

The KNIME Community

The KNIME Hub is the public repository for the KNIME community. Here, you can share your workflows and download workflows from other KNIME users. Just type in the keywords, and you get a list of related workflows, components, extensions, and more. It is a great place to start with plenty of examples! For example, enter "basic" or "beginners" in the search box, and you get a list of sample

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

workflows illustrating basic concepts in KNIME Analytics Platform; type in "read file," and you get a list of sample workflows illustrating how to read CSV files, table files, Excel files, and so on. (Note that some of these sample workflows are also reported in the EXAMPLES server in the KNIME Explorer in the top-left corner of the KNIME workbench.)

When you isolate a workflow that you might be interested in, you can click on it to open its page and then download it or open it in KNIME Analytics Platform. In your local workspace, you can start adapting the workflow to your data and your needs. After searching for ready-to-use pieces of code, you can just download, reuse, and adapt workflows or pieces of workflows from the KNIME Hub as appropriate for your own problem.

Of course, you can also share your own work on the KNIME Hub for the public good. Just copy the workflows to share from your local workspace into the *My-KNIME-Hub/Public* folder in the KNIME Explorer panel (see Figure A.2).

The KNIME community does not stop at the KNIME Hub. You can get tips and tricks on the KNIME Forum (https://forum.knime.com/) as well. Here, you can ask questions or search for previous answers. The community is very active, and it is highly likely that somebody has already asked your question.

Finally, contributions by the community are available as posts on the KNIME Blog (https://www.knime.com/blog), as books in KNIME Press (https://www.knime.com/knimepress), and as videos on KNIMETV (https://www.youtube.com/user/KNIMETV) channel on YouTube.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

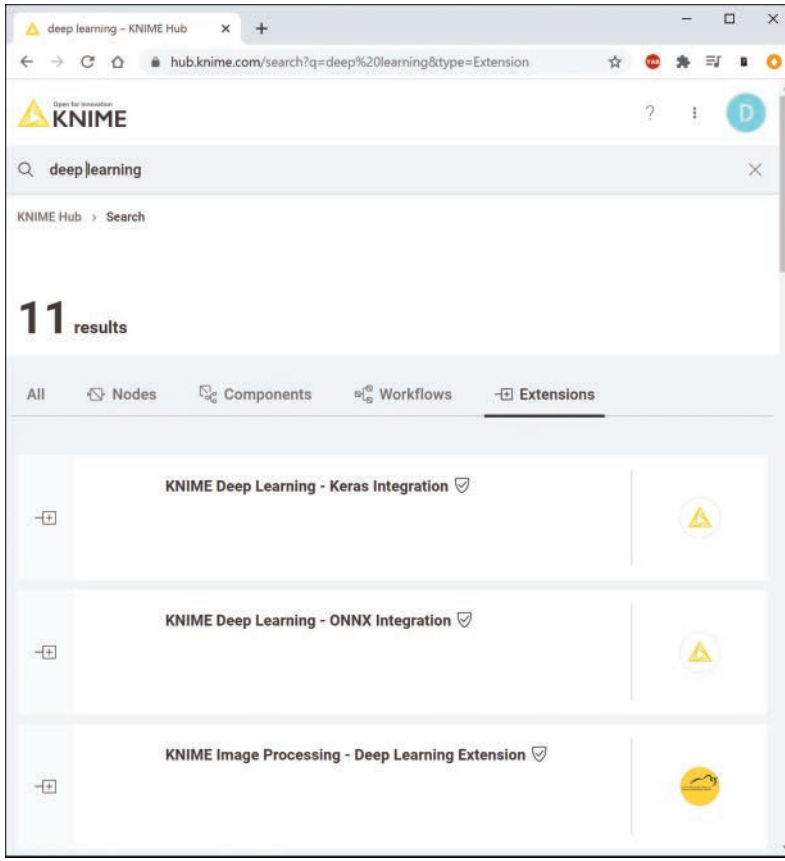


Figure A.2 List of Workflows Resulting from a Search for "deep learning" on the KNIME Hub

Correctness and Flexibility

Easy is good, but correctness is paramount. You need to know that a tool is flexible enough to experiment with alternative methods and procedures. This is indeed a key issue because for many of the software tools available today, easy is available only if you sacrifice control and flexibility. Many tools seem like canned, black-box solutions. For a data scientist, such magical solutions have always been questionable. It is important for a data scientist to have full control of the semi-experimental process and its underlying options to finding the most optimal solutions.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Automated machine learning (also called AutoML or AML) has become very popular recently. It carries the promise of getting your data and spitting out some results without requiring you to lift a finger. As fascinating as this option might sound, it carries some risk. First of all, it works as a black box: Its decision process is not transparent. When I feed the box with my data, I have to completely trust the machine that the data analysis process is correct, fits the data characteristics and distributions, and is tuned on the problem I wish to solve. I, personally, prefer to make sure that all steps in the analysis are implemented according to best practices, under my control, and as appropriate to the original design of the application. Although AutoML is a desired final stage in data science, because of the artistic nature of the solution development, it is still a work in progress, a utopia, and for real data scientist, it is nothing more than an early experimental step in piloting.

When executing a node in KNIME Analytics Platform, you can inspect the node options and output results (by simply right-clicking and then selecting the last option in the context menu) for debugging and validating the application step by step and make sure that all steps work as required by design. The world is never perfect/ideal, and neither are the data that comes out of it. Automated analysis might work well on perfect data that contains no unbalanced classes, no outliers, no dirty records, and so on. But perfect data only comes in fake data sets. In real life, there are always adjustments to make for the unbalanced classes, for the dirty data, for the missing values, for the outliers and inconsistencies, and so on. Adding one cleaning step or another, introducing an optimization cycle on one parameter or another might change the final results of the workflow. In order to experiment with new strategies, it is imperative that the tool be flexible and customizable enough at each step of the analysis. A black-box automated approach does not allow for much flexibility, does not allow you to change intermediate steps in the analysis, to customize features, or to tune parameters. KNIME, on the other hand, is very modular and, with extensive coverage of data wrangling operations and machine

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

learning algorithms, it allows for ample flexibility; it is flexible enough that you can exchange one data manipulation node with another with a simple drag-and-drop, introduce an optimization loop on the fly, or systematically and easily change/optimize the value of a parameter in the training of a specific machine learning model.

Note that if you open a workflow developed on an older version of KNIME Analytics Platform, you will probably find deprecated or legacy nodes. Current use of these nodes is discouraged as better, newer nodes have likely been developed. However, deprecated and legacy nodes do still exist, and they do still work. Indeed, backward compatibility with previous versions is a key and highly desired feature of analytics platforms like KNIME. Ensuring that old workflows still work exactly the same way they were designed to work can guarantee the reproducibility of the results. As you know, this is an extremely important argument for the validation of results in scientific research.

Extensive Coverage of Data Science Techniques

Another necessary complement to the ease of use in an analytics tool is the coverage of data science techniques. Ease of use would be of limited convenience without extensive coverage of data wrangling techniques, machine learning algorithms, handlers of a wide range of data types and formats, and integration with the most commonly used database software and data sources, reporting tools, and other scripting languages. A capable analytics platform is expected to keep up with the latest trends and technologies through native implementations and/or seamless external connections. Since no tool can do it all, all the time, integrating with and playing nicely with other tools and technologies in the fast-evolving business analytics data science landscape is not just a nice-to-have capability but a requirement for a comprehensive analytics platform.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Machine Learning Algorithms

KNIME Analytics Platform covers most machine learning algorithms—from traditional ones (e.g., linear and logistic regression, decision trees, neural networks, support vector machines, Naïve Bayes, *k*-nearest neighbor) to more recent ensembles (e.g., random forest, gradient boosted trees) and deep learning algorithms, from recommendation engines to a number of clustering techniques. Most of these algorithms are native in KNIME Analytics Platform, and some are integrated from other open source libraries through scripting nodes. One specific example is the deep learning capabilities, where layers, units, and preconfigured architectures are made available through the KNIME Deep Learning–Keras integration (https://www.knime.com/deeplearning/keras). This integration allows for easy use of the Keras libraries within KNIME Analytics Platform by offering them to the end users via the familiar KNIME graphical workflow-type user interface. It is possible to drag and drop nodes to create neuron layers for complex deep neural network architectures and train and test the final network without the need to write a single line of code. Figure A.3 shows a snapshot of the machine learning techniques included in KNIME.

Rich Collection of Data Handlers

A number of KNIME nodes are available to implement a myriad of data wrangling techniques. By logically connecting and combining individual nodes dedicated to small/atomic data processing tasks, you can implement very complex data preprocessing and transformation operations. The assembling of such operations is so easy that KNIME is often used for data preparation for other tools, such as to generate a consolidated data set for Excel or the data for creating reports in Tableau or PowerBI or to create summarized and synthesized data to be pushed into a data warehouse.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

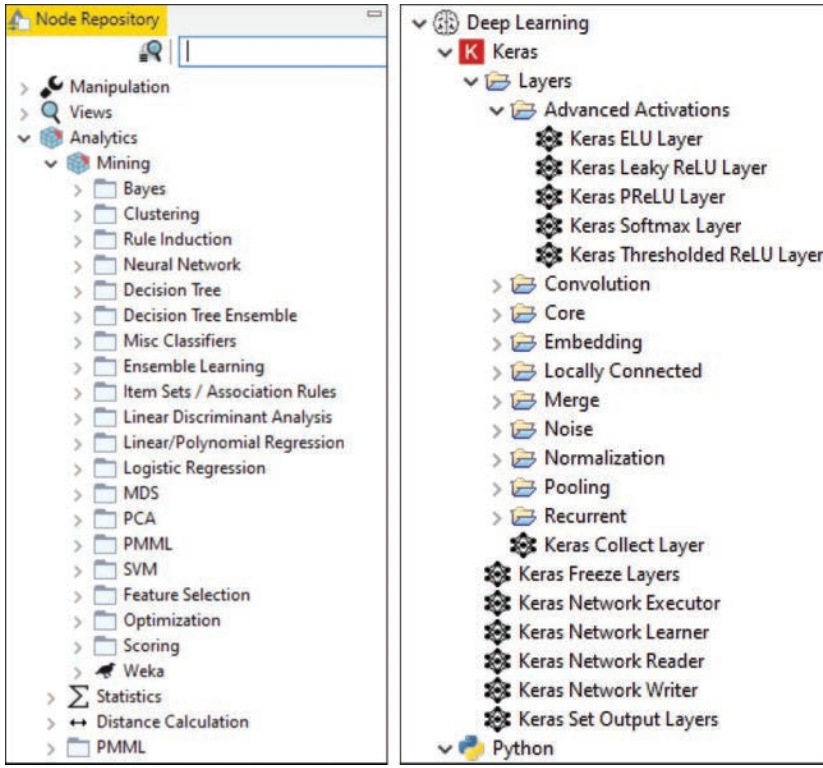


Figure A.3 The Mining Category in the Node Repository Holding the Machine Learning Algorithms

KNIME Analytics Platform also connects to most of the commonly used *data sources*: from databases to cloud repositories, from big data platforms to individual flat files. So no matter how weird your data source and your data types might be, it is quite likely that you can connect to them and consume them from within KNIME Analytics Platform.

Community Extensions and Third-party Integrations

What if all that is not enough? What if you need a specific procedure for DNA analysis or molecule conversion? What if you need a specific network manipulation function from Python? What if you need to export your results into a highly specialized web report?

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Where KNIME Analytics Platform cannot reach with its native nodes, there is usually at least one third-party extension from the KNIME community that provides the missing functionality for that particular domain specification and the related data type. Furthermore, there are also the integrations with other scripting and programming languages, such as Python, R, Java, and JavaScript, to consume additional functionality of other libraries and community resources.

KNIME Analytics Platform allows for seamless integration with the BIRT reporting tool, which is also open source. Integrations with other reporting platforms, such as Tableau, QlikView, PowerBI, and Spotfire, are also available. However, in those cases, you would need to buy the necessary license and install the destination reporting software.

In most cases, it turns out that it is not even necessary to move to an external reporting software tool to visualize your data and your analysis results. A number of JavaScript-based nodes are already available within KNIME, in the Views/JavaScript category in the Node repository. These nodes implement data visualization using a variety of plots and charts—from a simple scatterplot to a more complex sunburst chart, from a simple histogram to a parallel coordinate plot, and more. These nodes seem simple but are potentially quite powerful. If you combine them within a component (i.e., an integrated modeling construct in KNIME), you can enable interactive selection of data points across multiple charts. The component thus inherits and combines all views from the contained nodes and connects them together such that, if points are selected and visualized in one chart, they can also be selected and visualized in the other charts of the component's composite view. For example, the three plots in Figure A.4 visualize the same data and are connected so that, when you select data in the bar chart, you also select and can optionally visualize only the selected data in the other two charts.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

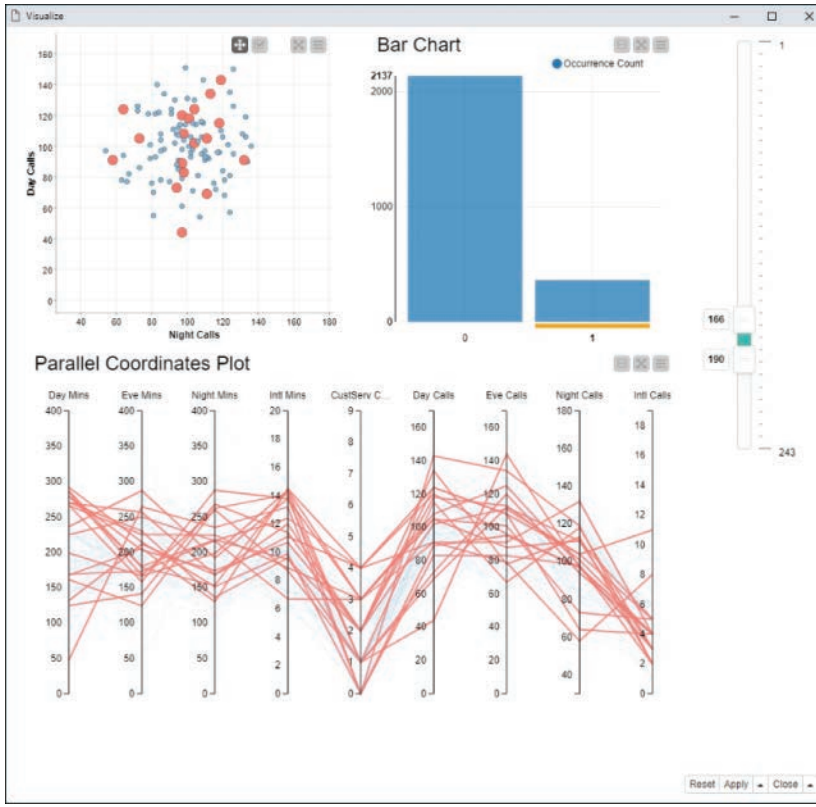


Figure A.4 The Composite View from a Component Containing a Scatterplot, a Bar Chart, and a Parallel Coordinate Plot

Data Science in the Enterprise

One last step is to deploy the outputs of the modeling efforts into production so that the insight generated by these models can be consumed for better and faster decision making. In an enterprise, this deployment must be quick, easy/intuitive, seamless, and secure. This is the last and perhaps the most critical step in the chain of actions in a data science project. When the workflow is finished (i.e., the model is trained and tested and the performance is measured), you need to push the application out into the real world to deal with real-life data to provide value to the intended user base. This process of moving

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

the application into the real world is called *moving into production*. The process of including the model in this final application is called *deployment*. These phases are deeply connected and can be quite problematic because all errors in the application design can and will show up at this stage.

It is possible to move an application into production by using KNIME Analytics Platform. If you, as a lone data scientist or a data science student, do not regularly deploy applications and models, KNIME Analytics Platform is probably enough for your needs. However, if you are just a bit more in an enterprise kind of environment, where scheduling, versioning, access rights, and all other typical functions of a production server are needed, then just using KNIME Analytics Platform for production can be cumbersome and may not produce sufficient value to the enterprise. In such a case, the KNIME Server, which is not open source but sold for an annual license fee, can make your life much easier. First, this server-based deployment fits better with the governance of the enterprise's IT environment. In addition, it offers a protected collaboration environment for your group and the whole data science lab. Finally, it makes model deployment and moving into production easier, faster, and safer, thanks to KNIME's integrated deployment feature and one-click deployment into production. End users can run the application from a KNIME Analytics Platform client or, even better, from a web-based application from within a browser.

The composite views mentioned earlier, which offer interactive and interconnected views of selected points, become fully formed web pages and easy-to-use applications when the solution is executed on a web browser via the KNIME Server's web portal. By using the components as touchpoints within the workflow, you get a guided analytics (https://www.knime.com/knime-introductory-course/chapter1/section4/guided-analytics) application within the web browser. The end user can take advantage of these touchpoints to insert knowledge or preferences and to steer the analysis in the desired direction.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Summary and Conclusions

With the brief introduction in this appendix, I hope that I have convinced you to give KIME a try and see for yourself why KNIME Analytics Platform is an excellent choice as a business analytics and data science tool to support you in your future endeavors as a data scientist.

KNIME's ease of use will save you time and allow you to dedicate more of your brain and energy to more important research topics. And remember that it is free, and solid community support it is available. KNIME also offers debugging capability that allows you to check the correctness of the implemented operations and remember the atomic tasks for all nodes so you can easily experiment with different analysis strategies. The easy prototyping is indeed one of the best features of KNIME Analytics Platform, enabling quick evaluation of and experimentation with new techniques.

Machine learning algorithms, data wrangling techniques, and accessible data sources make KNIME Analytics Platform a very reliable tool, but it is also easy to use. If the native nodes and extensions are not enough, community extensions, integrations with other scripting and programming languages, and integration with reporting tools can make up for the missing domain-specific functionalities.

Finally, the use of the KNIME Server as a complement to KNIME Analytics Platform for the enterprise allows for easy and secure deployment and seamless integration in the company IT environment.

Table A.1 summarizes the main reasons to consider KNIME Analytics Platform as a potential tool for your data science learning, teaching, and execution, using a top-10 list analogy.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Table A.1 The Top 10 Reasons to Consider KNIME

10	Ease of use	The graphical user interface with intuitive workflow model-building logic makes the analytics platform very easy and intuitive to learn, teach, and use.
9	Deployability	Putting the workflows in perpetual use via deployment to the cloud (on KNIME Server or on your own space in AWS or Azure) is a quick and straightforward process.
8	Expandability	KNIME allows you to build your own nodes and use nodes built by others. Also, you can expand the functionality and expandability via JavaScript, Python, and R integration nodes.
7	Community support	Access to KNIME gurus as well as a large and highly active community of users for help on questions, modeling hints, and example workflows is just a few clicks away on the KNIME website.
6	Modularity	KNIME allows for creation of reusable modeling artefacts at varying levels of granularity via the use of meta-nodes and components.
5	Connectivity	KNIME connects to almost any data source (local, web-based, or cloud) and consumes any and every data type (structured, semi-structured, unstructured)
4	Rich functionality	With more than 2,000 native nodes and many more from third parties (via extensions), KNIME offers perhaps the richest collection of data science functions.
3	Platform agnostic	KNIME natively runs on all three popular operating systems: Windows, MacOS, and Linux.
2	Open source	The source of the analytics platform is open to everyone for exploration and innovation.
1	Free	It is free! And it's not a scaled-down, time-limited, trial, or community version but the full functional version. And it can be used for everything—learning, teaching, and consulting.

Acknowledgment

A significant portion of this appendix was created, with permission, based on a blog post written by Ms. Rosaria Silipo (https://medium.com/swlh/why-knime-98c835afc186), who is a principal data scientist and the team leader for the educational and evangelical activities at KNIME.

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

This page intentionally left blank

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Index

A

Abbott, D., 207
ACM (Association for Computing Machinery), SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining), 191
AdaBoost, 198–199, 202–203
affordability, analytics, 8–9
AI (artificial intelligence), 325. *See also* cognitive computing
  and cognitive computing, 376–379
  cognitive search, 379–381
  deep learning, 326–328, 330
    applications, 326
    complexity, 329–330
    modeling, 328–329
AlexNet, 355
algorithms. *See also* classification; *k*-nearest-neighbor algorithm; machine learning; training
  Apriori, 132–133
  data mining, 99
  decision trees, 120
  genetic, 118
  k-means, 127
  k-NN, 152–158
  LDA (latent Dirichlet allocation), 271–272
  Naive Bayes, 148–152
  NLP (natural language processing), 245
  OLS (ordinary least squares), 172–173
  predictive analytics, 147
  TreeNet, 202–203
analytics, 1–2, 6–7, 33–34
  and analysis, 6
  applications, 2, 9
  availability and affordability, 8–9
  cognitive, 379
  and culture change, 9
  data preprocessing, 99–101
  data stream, 295
  history of, 16–17
    1970s, 13–14
    1980s, 14
    1990s, 14–15
    2000s, 15–16
    2010s and 2020s, 16
  IBM Watson, 22–23
    DeepQA, 24–26
    and finance, 28–29
    future of, 26
    and government, 30
    and health care, 27–28
    Jeopardy!, 23–24
    and research, 30
    and retail industry, 29–30
    and security, 28
  KDnuggets poll findings, 55–57
  market-based, 50–51
  need for, 7–8
  predictive, 106–107
  roadblocks to adoption
    analytics talent, 10
    culture, 10
    data, 11
    ROI (return on investment), 10–11
    security and privacy, 11–12
    technology, 11
  stream
    and big data, 315–318
    in the energy industry, 317–318
  taxonomy, 17–22
    descriptive, 19–20
    predictive, 20
    prescriptive, 20–21
  text, 235, 236
  visual, 52
analytics tools
  correctness and flexibility, 398–400
  extensive coverage of data science techniques, 400
ANNs (artificial neural networks), 158, 331. *See also* neural networks
  versus biological neural networks, 160–163
  information processing, 162–163
  learning process, 340–342

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

backpropagation for training, 342–346
training, 342, 345
network structure, 336–337
connection weights, 338
input, 338
outputs, 338
summation function, 338–339
transfer function, 339–340
PEs (processing elements), 336
versus SVMs, 168–169
APIs (application programming interfaces), Keras, 372–373
Apriori algorithm, 132–133
area under the ROC curve methodology, 115
assessment, model, 173–175
associations, 47, 50–51, 128–132, 260–261
Apriori algorithm, 132–133
metrics, 130–132
assumptions
in linear regression, 176–177
in topic modeling, 268
attributes, 119
automated data collection systems, 8
AutoML, 399
average causal analysis, 226
averaging, 180–181

B

backpropagation, 342–346
SGD (stochastic gradient descent), 342–343
bagging, 196–198, 220–221
and boosting, 200
bootstrap replicates, 198
random forests, 201–202
bag-of-words, 241–243
banking and finance, fraud detection, 381–386
Bayes theorem, 148–150
Bayesian classifiers, 118
beta coefficients, 226
bias, 193–194, 210
big data, 3, 4–5, 16, 289–290
challenges, 300–301
data scientists, 312–314
educational background, 314
skills, 314–315
data stream mining, 318–319
fundamental concepts, 296–297
for political campaigns, 319–321
problems addressed by, 302
sources of, 290–292
and stream analytics, 315–318
success factors, 298–300
technologies, 302–303
Hadoop, 305–311
MapReduce, 303–304, 305
NoSQL, 312
three Vs, 320
variety, 294
velocity, 294–295
volume, 292–294
value proposition, 296
variability, 295–296
veracity, 295
biological neural networks, versus ANNs, 160–163
Blei, D. M., 271–272
boosting, 198–201, 220–221
and bagging, 200
bootstrapping, 116–117
Breiman, L., 196, 198
business analytics, 1–2, 3
and data science, 5–6
business intelligence, 3, 15–16, 20, 53

C

Caffe, 370–371
cancer research, data mining, 89–93
candidate generation, 132
case-based reasoning, 118
categorical data, 75, 96–97
Chun, H. W., 250–251
classification, 20, 49, 107, 108–111, 259
accuracy metrics, 112–113
bagging, 196–197
Bayesian, 118
case-based reasoning, 118
confusion matrix, 111
cross-validation methodologies
area under the ROC curve, 115
bootstrapping, 116–117
jackknifing, 117
k-fold, 114
leave-one-out, 116–117
simple split, 113–114

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

decision trees, 50, 117, 119–121
algorithms, 120
attributes, 119
Gini index, 121
splits, 119–120
ensemble models, 118
genetic algorithms, 118
ILSVRC (ImageNet Large Scale Visual Recognition Challenge), 356
image, 354–359
k-nearest-neighbor algorithm, 118
Movie Forecast Guru, 62–63
multi-criteria decision-making situation, 109–110
Naive Bayes, 148–150
testing phase, 152
training phase, 151–152
NCAA bowl games prediction, 135–144
neural networks, 50, 117–118, 158–160
biological, 159
biological versus artificial, 160–163
information processing, 162–163
rough sets, 118
rule induction, 50
statistical analysis, 117
support vector machines, 118
two-step methodology, 109
Clinton, H., 321
cluster analysis, 121–122
clustering methods, 123–124
determining the number of clusters, 123
external assessment, 126
internal assessment, 125–126
clustering, 47, 51–52, 259–260, 264
k-means, 126, 127
CNNs (convolutional neural networks), 351–352
convolution functions, 352–353
GoogLeNet, 356, 357
image processing, 354–359
AlexNet, 355
Facebook, 358–359
ILSVRC (ImageNet Large Scale Visual Recognition Challenge), 356
ReLu (rectified linear unit) transfer functions, 355
parameter sharing, 353
pooling, 353–354
text processing, 359–361
cognitive analytics, 379
cognitive computing, 325, 373–374, 375
and AI, 376–379
key attributes, 375–376
use cases, 378–379
cognitive search, 379–381
complexity
deep learning, 329–330
of ensemble models, 207
and performance, 221–222
computer frameworks for implementation of deep learning, 368–369
Caffe, 370–371
TensorFlow, 371
Theano, 372
Torch, 369–370
confusion matrix, 111
context, 361–362
correlation, 198
versus regression, 170–171
CRISP-DM (Cross-Industry Standard Process for Data Mining), 71–73, 89, 136
business understanding, 73
data preparation, 75–76
data understanding, 73–75
deployment, 79–80
model building, 77–78
versus SEMMA, 84
testing and evaluation, 78–79
CRM (customer relationship management), 41–42
and NLP, 245
VOC (voice of the customer), 275
VOM (voice of the market), 276
Crossland, M., 262–263, 264
cross-validation methodologies, 137, 156–157
area under the ROC curve, 115
bootstrapping, 116–117
jackknifing, 117
k-fold, 114
leave-one-out, 116–117
simple split, 113–114
crowdsourcing, 192–193
CUDA (Compute Unified Device Architecture), 368–369
culture change, and analytics, 9

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

D

data, 36–37, 95–96, 99
  categorical, 96
  imbalanced, 215–217
    mitigation, 217–221
  interval, 98
  and knowledge, 34, 35
  nominal, 97
  numeric, 97–98
  ordinal, 97
  outliers, 76
  preprocessing, 99–101, 104–105, 166–167
  qualitative, 75
  quantitative, 75
  ratio, 98
  records, 103
  spatial, 98
  structured, 96
  unstructured, 96, 235, 239
  velocity, 294–295
  volume, 292–294
data analytics, 2
data deluge, 1–2
data mining, 6–7, 15–16, 33, 34, 36–39, 47–48, 95, 105–106. *See also* classification; cluster analysis; prediction; predictive analytics; standardized processes for predictive analytics; text mining
  algorithms, 99
  associations, 128–132
  BI tools, 53
  and cancer research, 89–93
  cluster analysis, 121–122
    clustering methods, 123–124
    determining the number of clusters, 123
    external assessment, 126
    internal assessment, 125–126
  commercial tools, 52–53
  CRISP-DM (Cross-Industry Standard Process for Data Mining), 71–73
    business intelligence, 73
    data preparation, 75–76
    data understanding, 73–75
    deployment, 79–80
    model building, 77–78
    testing and evaluation, 78–79
  data preprocessing, 99–105
  for Hollywood managers, 61–65
  KDnuggets poll findings, 55–57
  misconceptions, 39–41, 133–135
  most common applications
    banking and finance, 42
    brokerages and securities trading, 43
    computer hardware and software, 43–44
    CRM (customer relationship management), 41–42
    entertainment, 45
    government and defense, 44
    health and health care, 44–45
    homeland security and law enforcement, 45–46
    insurance, 43
    manufacturing, 43
    marketing, 41–42
    medicine, 45
    retail, 42
    sports, 46
    travel and lodging, 44
  and OLAP (online analytical processing), 40
  open source/free tools, 53
  patterns, 46–47, 48
    associations, 47
    clusters, 47
    predictions, 47
  popularity of, 35–36
  prediction, 106–108
    classification, 108–111
    and health care, 181–185
  and privacy, 58–60
  software tools, 54–55
  software vendors, 52–53
  SQL Server, 55
  and statistics, 40–41
  taxonomy
    classification, 49–51
    clustering, 51–52
    prediction, 49
  variables, 99, 107–108
  visual analytics, 52
data science, 3
  and business analytics, 5–6
data scientists, 312–314, 392–393
  educational background, 314
  projects, 391–392

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

skills, 314–315
data scrubbing, 100
data stream analytics, 295
data stream mining, 318–319
data warehousing, 69–70
Davenport, T., 33–34, 312–313
Dean, J., 303
deception detection, 249
text-based, 282–285
decision trees, 50, 90–91, 117, 119–121
algorithms, 120
bagging, 196
Gini index, 121
ID_3, 121
splits, 119–120
Deep Blue, 22, 23
deep learning, 325, 326–328, 330
applications, 326
CNNs (convolutional neural networks), 351–352
convolution functions, 352–353
GoogLeNet, 356, 357
image processing, 354–359
pooling, 353–354
text processing, 359–361
complexity, 329–330
computer frameworks, 368–369
Caffe, 370–371
TensorFlow, 371
Theano, 372
Torch, 369–370
fraud detection, 381–384
Google Lens, 357
modeling, 328–329
unsupervised, 349–350
deep networks, 346–347, 363
MLP (multilayer perceptron), 347–348
impact of random weights, 349–350
layers, 350–351
DeepQA, 24–26
Delen, D., 62, 63, 90–92, 136, 182, 184, 262–263, 264, 264, 318–319
descriptive analytics, 1, 19–20
dimensional reduction, 102–103
discretization, 108
DWs (data warehouses), 14–15

E

education, and IBM Watson, 29–30
ensemble models, 118, 189–190
accuracy, 192
advantages and shortcomings, 208–209
bagging, 196–198
bootstrap replicates, 198
random forests, 201–202
bias-variance trade-off, 193–194
boosting, 198–201
stochastic gradient, 202–203
complexity, 207
and crowdsourcing, 192–193
heterogeneous, 195–196
homogeneous, 195–196
information fusion, 205–206
need for, 190–192
stacking, 203–205
transparency, 207–209
entropy, 121
ERP systems, 14
ESs (expert systems), 13–14
Euclidean distance, 155
evidence-based medicine, 242–243

F

Fayyad, U., 69–70
feedforward models, 347–348
finance industry, and IBM Watson, 28–29
fitting, 212
under-, 212–213
over-, 212
forecasting, 49. *See also* prediction
time-series, 52, 179–181
fraud detection, 381–386
Freund, Y., 198
Friedman, J., 202–203

G

Galton, F., 169–170
generative models, 349–350
genetic algorithms, 118
Ghani, R. K., 248
Ghemawat, S., 303

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

GIGO (garbage-in, garbage-out), 99–101
Gini index, 121
global interpretability, 226
Google, word2vec, 359–361
Google Lens, 357
GoogLeNet, 356, 357
government, and IBM Watson, 30
GPUs (graphics processing units), 346–347, 368–369
Gualtieri, M., 379–380

H

Hadoop, 305–306, 307
  HDFS (Hadoop Distributed File System), 306
  job tracker, 306
  MapReduce phase, 306–307
  misconceptions, 309–311
  pros and cons, 308–309
  technical components, 307–308
health care
  and IBM Watson, 27–28
  predictive modeling techniques, 181–185
heterogeneous ensembles, 195–196. *See also* ensemble models
hierarchical clustering methods, 124–125
history of analytics, 16–17
  1970s, 13–14
  1980s, 14
  1990s, 14–15
  2000s, 15–16
  2010s and 2020s, 16
homogeneous ensembles, 195–196. *See also* ensemble models
hyperplanes, 164–165

I

IBM Watson, 22–23
  applications, health care, 27–28
  DeepQA, 24–26
  and education, 29–30
  and finance, 28–29
  future of, 26
  and government, 30
  Jeopardy!, 23–24
  and research, 30
  and retail industry, 29
  and security, 28
ImageNet, 355
imbalanced data, 215–217
  mitigation, 217–221
Inception, 356
information fusion, 205–206
INFORMS (Institute for Operations Research and Managements Science), Capgemeni study, 17–19
intercept, 173
interval data, 98
IoT (Internet of Things), 3, 4–5
irreducible error, 215

J-K

jackknifing, 117
Jennings, K., 23

K

Kadam, A., 90–91
Kaggle, 191
KDD (knowledge discovery in databases), 69–71
KDnuggets poll findings, 55–57
Keras, 372–373
k-fold cross-validation, 114
k-means, 51–52, 126, 127
k-nearest-neighbor algorithm, 118, 152–154, 157–158
  cross-validation, 156–157
  Euclidean distance, 155
  Manhattan distance, 155
  Minkowski distance, 155
  parameter selection, 156
  similarity measure, 154–156
KNIME Analytics Platform, 53, 393–394, 406
  community extensions and third-party integrations, 402–403
  correctness and flexibility, 398–399
  data handlers, 401–402
  machine learning algorithms, 401–402
  nodes, 399–400
  reasons to consider, 407

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

user community, 396–397
visual programming, 393–395
learning curve, 395–396
knowledge, and data, 34, 35
knowledge extraction
associations, 260–261
classification, 259
clustering, 259–260
trend analysis, 261
Kohenen, T., 51–52

L

LDA (latent Dirichlet allocation), 270–271
algorithms, 271–272
learning, 331. *See also* deep learning; machine learning; supervised learning
leave-one-out methodology, 116–117, 224–226
LeCun, Y., 329–330
LIME (Local Interpretable Model-Agnostic Explanations), 227
and SHAP (Shapley Additive Explanations), 228
linear regression, 169–170
assumptions, 176–177
beta coefficients, 226
link analysis, 51
local interpretability, 226
logistic regression, 90–91, 177–179
coefficients, 179
link function, 178
simple, 177–178
LSA (latent semantic analysis), 268–269
LSTM (long short-term memory) networks, 364–366
applications, 366–368

M

machine learning, 3, 4–5, 108–109, 213, 325
and complexity, 221–222
deep learning, 326–328, 330
applications, 326
complexity, 329–330
modeling, 328–329
imbalanced data problems, mitigation, 220
LIME (Local Interpretable Model-Agnostic Explanations), 227
sensitivity analysis, 222
and statistical methods, 147
SVMs (support vector machines), 163–165
versus ANNs, 168–169
hyperplanes, 164–165
modeling, 165–168
training, overfitting, 345
Manhattan distance, 155
MapReduce, 303–304, 305
market-based analysis, 50–51, 128–129. *See also* associations
marketing
and data mining, 41–42
and privacy, 58–59
and text mining, 247–248
MART (multiple additive regression trees), 202–203
McCulloch, 159–160
memory
LSTM (long short-term memory) networks, 364–368
RNNs (recurrent neural networks), 362–364
message feature mining, 283–284
Microsoft SQL Server, 55
Minkowski distance, 155
Minsky, 159–160
MLP (multilayer perceptron) deep networks, 330–331, 347–348
impact of random weights, 349–350
layers, 350–351
modeling
assessment, 173–175
bias, 210
bias-variance trade-off, 210–211
CRISP-DM (Cross-Industry Standard Process for Data Mining), 77–78
deep learning, 328–329
deployment, 404–405
ensemble, 189–190
accuracy, 192
advantages and shortcomings, 208–209
bagging, 196–198, 201–202
bias-variance trade-off, 193–194

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

boosting, 198–201, 202–203
complexity, 207
and crowdsourcing, 192–193
heterogeneous, 195–196
homogeneous, 195–196
information fusion, 205–206
need for, 190–192
stacking, 203–205
transparency, 207–209
LIME (Local Interpretable Model-Agnostic Explanations), 227
regression, 172–173
SEMMA, 83
sensitivity analysis, 222
SHAP (Shapley Additive Explanations), 227
SVMs (support vector machines), 165–168
topic, 268
assumptions, 268
LSA (latent semantic analysis), 268–269
shortcomings, 270
variable importance methods, 222–223
variance, 210
Movie Forecast Guru, 61–65
moving into production, 404–405
multi-criteria decision-making situation, 109–110

N

Naive Bayes classification, 148–150
boosting, 201
testing phase, 152
training phase, 151–152
natural language processing, 3, 4–5
NCAA bowl games prediction, 135–144
evaluation, 141
methods, 141
results, 142–144
variables, 137–141
nearest neighbor algorithm, 152–154, 157–158
cross-validation, 156–157
Euclidean distance, 155
Manhattan distance, 155
Minkowski distance, 155
parameter selection, 156
similarity measure, 154–156
network gradients, 342–343
neural networks, 50, 90–91, 117–118, 158–160, 331. *See also* CNNs (convolutional neural networks); deep learning; deep networks
artificial, 160–163
biological, 159
general single-input artificial neuron representation, 331–332, 333–334
information processing, 162–163
MLP (multilayer perceptron), 330–331
network structure, 336–337
connection weights, 338
input, 338
outputs, 338
summation function, 338–339
transfer function, 339–340
PEs (processing elements), 336
recurrent, 347–348, 362–364
with three layers and eight neurons, 335
transfer functions, 332–334
typical multiple-input neuron with R individual inputs, 334–335
NLP (natural language processing), 235, 243
algorithms, 245
applications, 245–246
bag-of-words, 241–243
part-of-speech tagging, 244
speech acts, 244
text segmentation, 244
'*understanding*', 243–244
word sense disambiguation, 244
WordNet, 245
nominal data, 97
NoSQL, 312
numeric data, 97–98

O

Obama, B., 320–321
OLAP (online analytical processing), 40
OLS (ordinary least squares), 172–173
ordinal data, 97

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

OTMI (Open Text Mining Interface), 251–252
overfitting, 212, 345, 347
oversampling, 218–219

P

Papert, 159–160
parameter selection, nearest neighbor algorithm, 156
parameter sharing, 353
partitive clustering methods, 125
Patil, D. J., 312–313
Pearson, K., 169–170
PEs (processing elements), 336
Piri, S., 216, 219–220
Pitts, 159–160
polarity identification, 282
prediction, 49, 106–108. *See also* classification; modeling; statistical analysis
  assessment metrics, 109–111
  and cancer research, 90–93
  classification, 108–111
    accuracy metrics, 112–113
    assessment metrics, 109–111
    Bayesian, 118
    case-based reasoning, 118
    decision trees, 117, 119–121
    ensemble models, 118
    genetic algorithms, 118
    k-nearest-neighbor algorithm, 118
    multi-criteria decision-making situation, 109–110
    neural networks, 117–118, 158–160
    rough sets, 118
    statistical analysis, 117
    support vector machines, 118
    two-step methodology, 109
  Movie Forecast Guru, 61–65
  NCAA bowl games, 135–144
  regression, 108–109, 169–170
    versus correlation, 170–171
    linear, 169–177
    logistic, 177–179
    model assessment, 173–175
    modeling, 172–173
    simple versus multiple, 171
  time-series forecasting, 179–181
predictive analytics, 1, 20, 95, 189. *See also* data mining; standardized processes for predictive analytics; statistical analysis
  algorithms
    k-NN, 152–158
    Naive Bayes, 148–152
  bias-variance trade-off, 209–210, 214–215
  classification, confusion matrix, 111
  CRISP-DM (Cross-Industry Standard Process for Data Mining), 71–73
    business understanding, 73
    data preparation, 75–76
    data understanding, 73–75
    deployment, 79–80
    model building, 77–78
    testing and evaluation, 78–79
  fitting, 212
    under-, 212–213
    over-, 212
  and health care, 181–185
  imbalanced data problems, 215–217
    mitigation, 217–221
  irreducible error, 215
  KDD (knowledge discovery in databases), 69–71
  misconceptions, 133–135
  predicting outcomes of drug court cases, 228–232
  SEMMA, 80–81
    assessment, 83
    exploring, 82
    model building, 83
    modification, 82–83
    sampling, 81–82
  Six Sigma, 84–86
    analyze stage, 87
    control stage, 87–88
    define stage, 85
    improve stage, 87
    measurement stage, 85
  variable importance methods, 222–223
prescriptive analytics, 1, 20–21
privacy, and data mining, 58–60
probability, Bayes theorem, 148–150
probability distribution, 75

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

Q-R

qualitative data, 75
quantitative data, 75
random forests, 201–202
RapidMiner, 53
ratio data, 98
RBF kernel, 167–168
RDBM (relational database management) systems, 14
real-time data warehousing, 15–16
records, 103
regression, 20, 90–91, 107, 108–109. *See also* linear regression
  bagging, 196–197
  versus correlation, 170–171
  linear, 169–170
    assumptions, 176–177
    beta coefficients, 226
    model assessment, 173–175
  logistic, 177–179
    coefficients, 179
    link function, 178
    simple, 177–178
  model assessment, 173–175
  modeling, 172–173
  OLS (ordinary least squares), 172–173
  simple versus multiple, 171
regularization, 345
research, and IBM Watson, 30
research literature, text mining, 261–265
retail industry
  and data mining, 42
  and IBM Watson, 29
RFID (radio frequency identification), 8
Ribeiro, M. T., "Why Should I Trust You?", 221–222
right-time data warehousing, 15–16
RNNs (recurrent neural networks), 347–348, 362–364. *See also* LSTM (long short-term memory) networks
ROC (receiver operating characters), 115
ROI (return on investment), analytics, 10–11
Romney, M., 321
rough sets, 118
rule induction, 50
rule-based expert systems, 13–14
Rutter, B., 23

S

sampling
  under-, 219
  bootstrap, 198
  over-, 218–219
Schapire, R. E., 198
security
  and IBM Watson, 28
  and text mining, 248–249
SEER (Surveillance, Epidemiology, and End Results), 91–92
SEMMA, 80–81
  assessment, 83
  versus CRISP-DM, 84
  exploring, 82
  model building, 83
  modification, 82–83
  sampling, 81–82
sensitivity analysis, 65, 222–223
  leave-one-out methodology, 224–226
sentiment analysis, 272–274
  applications, 274–275
    brand management, 276
    financial markets, 276
    government intelligence, 277
    politics, 277
    VOC (voice of the customer), 275
    VOE (voice of the employee), 276
    VOM (voice of the market), 276
  collection and aggregation, 281–282
  N-P polarity classification, 280–281
  polarity identification, 282
  sentiment detection, 280
  target identification, 281
sentiment classification, 273
sequence mining, 51
SHAP (Shapley Additive Explanations), 227
  and LIME (Local Interpretable Model-Agnostic Explanations), 228
Sharda, R., 62, 63
Shatkay, H. A., 250
Shen, G., 320–321
similarity measure, 154–156

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

simple split methodology, 113–114
Six Sigma, 84–86
  analyze stage, 87
  control stage, 87–88
  define stage, 85
  improve stage, 87
  measurement stage, 85
Skype Translator Service, 367–368
slope, 173
spatial data, 98
SQL Server, 55
stacking, 203–205
standardized processes for predictive analytics
  comparison of methodologies, 88
  CRISP-DM (Cross-Industry Standard Process for Data Mining), 71–73, 89
    business understanding, 73
    data preparation, 75–76
    data understanding, 73–75
    deployment, 79–80
    model building, 77–78
    testing and evaluation, 78–79
  KDD (knowledge discovery in databases), 69–71
  SEMMA, 80–81
    assessment, 83
    versus CRISP-DM, 84
    exploring, 82
    model building, 83
    modification, 82–83
    sampling, 81–82
  Six Sigma, 84–86
    analyze stage, 87
    control stage, 87–88
    define stage, 85
    improve stage, 87
    measurement stage, 85
statistical analysis, 117
  linear regression, assumptions, 176–177
  and machine learning, 147
  model assessment, 173–175
  regression, 169–170
    versus correlation, 170–171
    logistic, 177–179
    modeling, 172–173
    OLS (ordinary least squares), 172–173
    simple versus multiple, 171
statistics
  and data mining, 40–41
  k-means, 51–52
stochastic gradient boosting, 202–203
stream analytics
  and big data, 315–318
  in the energy industry, 317–318
  tuples, 316–317
structured data, 96
Summit, 22
supervised learning, 330, 349–350, 354–355
  backpropagation, 342–346
    SGD (stochastic gradient descent), 342–343
  learning process, 340–342
Surowiecki, J., *The Wisdom of the Crowds*, 192–193
SVMs (support vector machines), 118, 163–165, 219–220
  versus ANNs, 168–169
  hyperplanes, 164–165
  modeling, 165–166
    data preprocessing, 166–167
    deployment, 168
    development, 167–168
    RBF kernel, 167–168
synsets, 355

T

taxonomy
  analytics, 17–22
    descriptive, 19–20
    predictive, 20
    prescriptive, 20–21
  of clustering methods, 124–125
  of data
    categorical, 96
    interval, 98
    nominal, 97
    numeric, 97–98
    ordinal, 97
    ratio, 98
    spatial, 98
  data mining
    classification, 49–51
    clustering, 51–52
    prediction, 49
TDM (term-document matrix), 255

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

normalization, 257–258
reducing the dimensionality, 258–259
representing the indices, 257–258
stemming, 256
stop terms, 256
SVD (singular value decomposition), 258–259
synonyms, 256
TensorFlow, 371
test set, 113
text analytics, 235, 236
text categorization, 259
text mining, 15–16, 235, 236–237, 252–254. *See also* NLP (natural language processing); TDM (term-document matrix)
applications, 237–239
biomedical, 249–252
marketing, 247–248
security, 248–249
bag-of-words, 241–243
clustering, 264
concepts, 239–240
corpus, 239
ECHELON, 248
knowledge extraction
associations, 260–261
classification, 259
clustering, 259–260
trend analysis, 261
lemmatization, 240
morphology, 241
NLP (natural language processing), 243
algorithms, 245
applications, 245–246
part-of-speech tagging, 244
speech acts, 244
text segmentation, 244
word sense disambiguation, 244
WordNet, 245
OASIS (Overall Analysis System for Intelligence Support), 248–249
part-of-speech tagging, 241
polysemes, 240
research literature, 261–265
singular-value decomposition, 241
stemming, 240
stop words, 240
synonyms, 240
tasks
create the term-document matrix, 255–259
establish the corpus, 254–255
extract the knowledge, 259–261
TDM (term-document matrix)
normalization, 257–258
reducing the dimensionality, 258–259
representing the indices, 257–258
stemming, 256
stop terms, 256
SVD (singular value decomposition), 258–259
synonyms, 256
term dictionary, 241
term-by-document matrix, 241
terms, 239
text-based deception detection, 282–285
tokens, 240
tools
commercial software, 266
free software, 267
word frequency, 241
text-based deception detection, 282–285
Theano, 372
time-series forecasting, 20, 52, 107, 179–181
topic modeling, 268
assumptions, 268
LSA (latent semantic analysis), 268–269
shortcomings, 270
Torch, 369–370
training
backpropagation, 342–359
determining optimum level, 213–215
overfitting, 212, 345, 347
underfitting, 212–213
training set, 113
transparency, of ensemble models, 207–209
TreeNet, 202–203
trend analysis, 261
TrueText, 367–368
Trump, D., 321

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.

U

underfitting, 212–213
undersampling, 219
unstructured data, 96, 235, 239. *See also* text mining
unsupervised deep learning, 349–350

V

Valenti, J., 61
vanishing gradient problem, 355
variables, 99, 102, 107–108
  attributes, 119
  leave-one-out methodology, 224–226
  linear regression, 169–170
  NCAA bowl games prediction, 137–141
variance, 193–194, 210
velocity, 294–295
visual analytics, 52
Vorhies, W., 191

W

Walker, G., 90–91
Watson. *See* IBM Watson
Watson, T. J., 23
Weka, 53
word vectors, 359–360
word2vec, 359–361
WordNet, 245, 355

X-Y-Z

Yangqing, Jia, 370
Zadeh, A. H., 91–92
zettabyte (ZB), 293, 294
Zolbanin, H. M., 91–92

Humble Bundle Pearson AI Bundle — © Pearson. Do Not Distribute.