CHAPTER 1

Introduction

Data science is a collection of techniques used to extract value from data. It has become an essential tool for any organization that collects, stores, and processes data as part of its operations. Data science techniques rely on finding useful patterns, connections, and relationships within data. Being a buzzword, there is a wide variety of definitions and criteria for what constitutes data science. Data science is also commonly referred to as knowledge discovery, machine learning, predictive analytics, and data mining. However, each term has a slightly different connotation depending on the context. In this chapter, we attempt to provide a general overview of data science and point out its important features, purpose, taxonomy, and methods.

In spite of the present growth and popularity, the underlying methods of data science are decades if not centuries old. Engineers and scientists have been using predictive models since the beginning of nineteenth century. Humans have always been forward-looking creatures and predictive sciences are manifestations of this curiosity. So, who uses data science today? Almost every organization and business. Sure, we didn't call the methods that are now under data science as "*Data Science.*" The use of the term *science* in data science indicates that the methods are evidence based, and are built on empirical knowledge, more specifically historical observations.

As the ability to collect, store, and process data has increased, in line with Moore's Law - which implies that computing hardware capabilities double every two years, data science has found increasing applications in many diverse fields. Just decades ago, building a production quality regression model took about several dozen hours (Parr Rud, 2001). Technology has come a long way. Today, sophisticated machine learning models can be run, involving hundreds of predictors with millions of records in a matter of a few seconds on a laptop computer.

The process involved in data science, however, has not changed since those early days and is not likely to change much in the foreseeable future. To get meaningful results from any data, a major effort preparing, cleaning,

Data Science. DOI: https://doi.org/10.1016/B978-0-12-814761-0.00001-0

scrubbing, or standardizing the data is still required, before the learning algorithms can begin to crunch them. But what may change is the automation available to do this. While today this process is iterative and requires analysts' awareness of the best practices, soon smart automation may be deployed. This will allow the focus to be put on the most important aspect of data science: interpreting the results of the analysis in order to make decisions. This will also increase the reach of data science to a wider audience.

When it comes to the data science techniques, are there a core set of procedures and principles one must master? It turns out that a vast majority of data science practitioners today use a handful of very powerful techniques to accomplish their objectives: decision trees, regression models, deep learning, and clustering (Rexer, 2013). A majority of the data science activity can be accomplished using relatively few techniques. However, as with all 80/20 rules, the long tail, which is made up of a large number of specialized techniques, is where the value lies, and depending on what is needed, the best approach may be a relatively obscure technique or a combination of several not so commonly used procedures. Thus, it will pay off to learn data science and its methods in a systematic way, and that is what is covered in these chapters. But, first, how are the often-used terms artificial intelligence (AI), machine learning, and data science explained?

1.1 AI, MACHINE LEARNING, AND DATA SCIENCE

Artificial intelligence, Machine learning, and data science are all related to each other. Unsurprisingly, they are often used interchangeably and conflated with each other in popular media and business communication. However, all of these three fields are distinct depending on the context. Fig. 1.1 shows the relationship between artificial intelligence, machine learning, and data science.

Artificial intelligence is about giving machines the capability of mimicking human behavior, particularly cognitive functions. Examples would be: facial recognition, automated driving, sorting mail based on postal code. In some cases, machines have far exceeded human capabilities (sorting thousands of postal mails in seconds) and in other cases we have barely scratched the surface (search "artificial stupidity"). There are quite a range of techniques that fall under artificial intelligence: linguistics, natural language processing, decision science, bias, vision, robotics, planning, etc. Learning is an important part of human capability. In fact, many other living organisms can learn.

Machine learning can either be considered a sub-field or one of the tools of artificial intelligence, is providing machines with the capability of learning

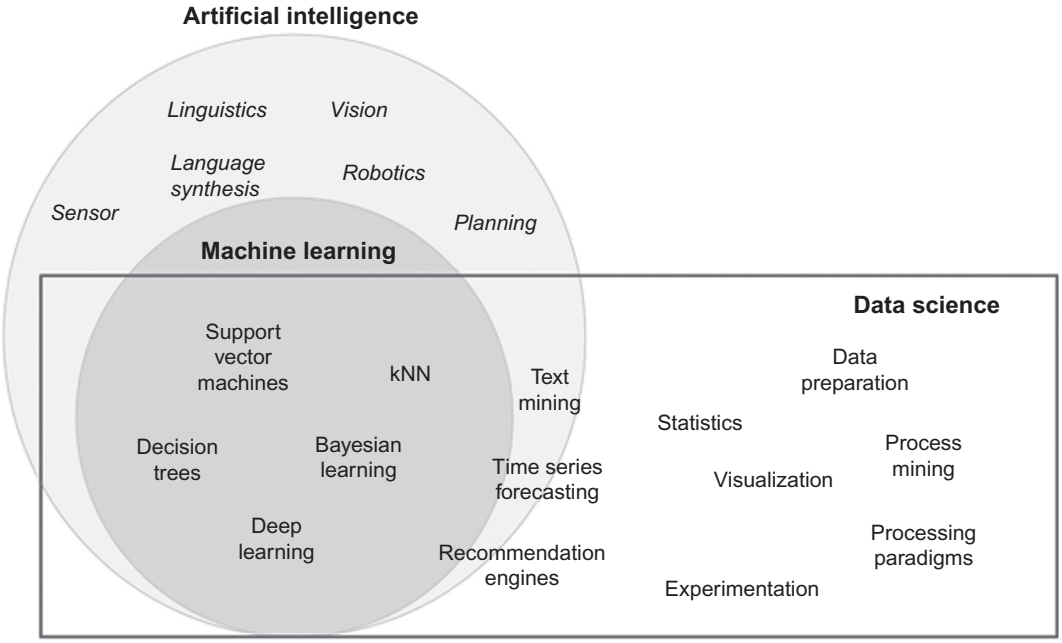

FIGURE 1.1
Artificial intelligence, machine learning, and data science.

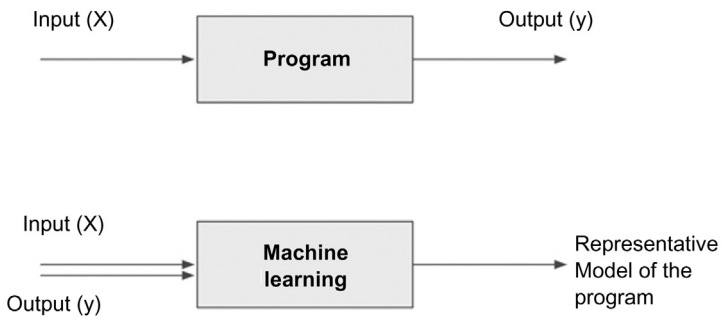

FIGURE 1.2
Traditional program and machine learning.

from experience. Experience for machines comes in the form of data. Data that is used to teach machines is called training data. Machine learning turns the traditional programing model upside down (Fig. 1.2). A program, a set of instructions to a computer, transforms input signals into output signals using predetermined rules and relationships. Machine learning algorithms,

also called "learners", take both the known input and output (training data) to figure out a model for the program which converts input to output. For example, many organizations like social media platforms, review sites, or forums are required to moderate posts and remove abusive content. How can machines be taught to automate the removal of abusive content? The machines need to be shown examples of both abusive and non-abusive posts with a clear indication of which one is abusive. The learners will generalize a pattern based on certain words or sequences of words in order to conclude whether the overall post is abusive or not. The model can take the form of a set of "if—-then" rules. Once the data science rules or model is developed, machines can start categorizing the disposition of any new posts.

Data science is the business application of machine learning, artificial intelligence, and other quantitative fields like statistics, visualization, and mathematics. It is an interdisciplinary field that extracts value from data. In the context of how data science is used today, it relies heavily on machine learning and is sometimes called data mining. Examples of data science user cases are: recommendation engines that can recommend movies for a particular user, a fraud alert model that detects fraudulent credit card transactions, find customers who will most likely churn next month, or predict revenue for the next quarter.

1.2 WHAT IS DATA SCIENCE?

Data science starts with *data*, which can range from a simple array of a few numeric observations to a complex matrix of millions of observations with thousands of variables. Data science utilizes certain specialized computational *methods* in order to discover meaningful and useful structures within a dataset. The discipline of data science coexists and is closely associated with a number of related areas such as database systems, data engineering, visualization, data analysis, experimentation, and business intelligence (BI). We can further define data science by investigating some of its key features and motivations.

1.2.1 Extracting Meaningful Patterns

Knowledge discovery in databases is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns or relationships within a dataset in order to make important decisions (Fayyad, Piatetsky-shapiro, & Smyth, 1996). Data science involves inference and iteration of many different hypotheses. One of the key aspects of data science is the process of *generalization* of patterns from a dataset. The generalization should be valid, not just for the dataset used to observe the pattern, but also for new unseen data. Data science is also a process with defined steps, each

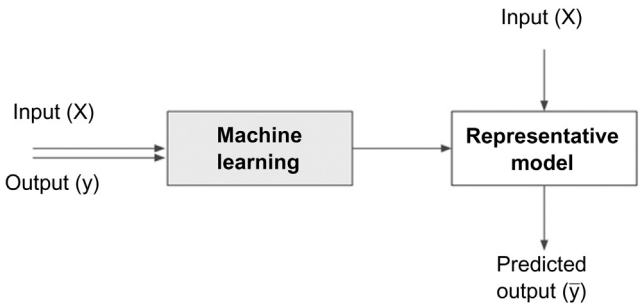

FIGURE 1.3
Data science models.

with a set of tasks. The term *novel* indicates that data science is usually involved in finding previously unknown patterns in data. The ultimate objective of data science is to find potentially useful conclusions that can be acted upon by the users of the analysis.

1.2.2 Building Representative Models

In statistics, a model is the representation of a relationship between variables in a dataset. It describes how one or more variables in the data are related to other variables. Modeling is a process in which a representative abstraction is built from the observed dataset. For example, based on credit score, income level, and requested loan amount, a model can be developed to determine the interest rate of a loan. For this task, previously known observational data including credit score, income level, loan amount, and interest rate are needed. Fig. 1.3 shows the process of generating a model. Once the representative model is created, it can be used to predict the value of the interest rate, based on all the input variables.

Data science is the process of building a representative model that fits the observational data. This model serves two purposes: on the one hand, it predicts the output (interest rate) based on the new and unseen set of input variables (credit score, income level, and loan amount), and on the other hand, the model can be used to understand the relationship between the output variable and all the input variables. For example, does income level really matter in determining the interest rate of a loan? Does income level matter more than credit score? What happens when income levels double or if credit score drops by 10 points? A Model can be used for both predictive and explanatory applications.

1.2.3 Combination of Statistics, Machine Learning, and Computing

In the pursuit of extracting useful and relevant information from large datasets, data science borrows computational techniques from the disciplines of statistics, machine learning, experimentation, and database theories. The algorithms used in data science originate from these disciplines but have since evolved to adopt more diverse techniques such as parallel computing, evolutionary computing, linguistics, and behavioral studies. One of the key ingredients of successful data science is substantial prior knowledge about the data and the business processes that generate the data, known as *subject matter expertise*. Like many quantitative frameworks, data science is an iterative process in which the practitioner gains more information about the patterns and relationships from data in each cycle. Data science also typically operates on large datasets that need to be stored, processed, and computed. This is where database techniques along with parallel and distributed computing techniques play an important role in data science.

1.2.4 Learning Algorithms

We can also define data science as a process of discovering previously unknown patterns in data using *automatic iterative methods*. The application of sophisticated learning algorithms for extracting useful patterns from data differentiates data science from traditional data analysis techniques. Many of these algorithms were developed in the past few decades and are a part of machine learning and artificial intelligence. Some algorithms are based on the foundations of Bayesian probabilistic theories and regression analysis, originating from hundreds of years ago. These iterative algorithms automate the process of searching for an optimal solution for a given data problem. Based on the problem, data science is classified into *tasks* such as classification, association analysis, clustering, and regression. Each data science task uses specific learning algorithms like decision trees, neural networks, *k*-nearest neighbors (*k*-NN), and *k*-means clustering, among others. With increased research on data science, such algorithms are increasing, but a few classic algorithms remain foundational to many data science applications.

1.2.5 Associated Fields

While data science covers a wide set of techniques, applications, and disciplines, there a few associated fields that data science heavily relies on. The techniques used in the steps of a data science process and in conjunction with the term "data science" are:

- *Descriptive statistics*: Computing mean, standard deviation, correlation, and other descriptive statistics, quantify the aggregate structure of a

dataset. This is essential information for understanding any dataset in order to understand the structure of the data and the relationships within the dataset. They are used in the exploration stage of the data science process.

- *Exploratory visualization*: The process of expressing data in visual coordinates enables users to find patterns and relationships in the data and to comprehend large datasets. Similar to descriptive statistics, they are integral in the pre- and post-processing steps in data science.
- *Dimensional slicing*: Online analytical processing (OLAP) applications, which are prevalent in organizations, mainly provide information on the data through dimensional slicing, filtering, and pivoting. OLAP analysis is enabled by a unique database schema design where the data are organized as dimensions (e.g., products, regions, dates) and quantitative facts or measures (e.g., revenue, quantity). With a well-defined database structure, it is easy to slice the yearly revenue by products or combination of region and products. These techniques are extremely useful and may unveil patterns in data (e.g., candy sales decline after Halloween in the United States).
- *Hypothesis testing*: In confirmatory data analysis, experimental data are collected to evaluate whether a hypothesis has enough evidence to be supported or not. There are many types of statistical testing and they have a wide variety of business applications (e.g., A/B testing in marketing). In general, data science is a process where many hypotheses are generated and tested based on observational data. Since the data science algorithms are iterative, solutions can be refined in each step.
- *Data engineering*: Data engineering is the process of sourcing, organizing, assembling, storing, and distributing data for effective analysis and usage. Database engineering, distributed storage, and computing frameworks (e.g., Apache Hadoop, Spark, Kafka), parallel computing, extraction transformation and loading processing, and data warehousing constitute data engineering techniques. Data engineering helps source and prepare for data science learning algorithms.
- *Business intelligence*: Business intelligence helps organizations consume data effectively. It helps query the ad hoc data without the need to write the technical query command or use dashboards or visualizations to communicate the facts and trends. Business intelligence specializes in the secure delivery of information to right roles and the distribution of information at scale. Historical trends are usually reported, but in combination with data science, both the past and the predicted future data can be combined. BI can hold and distribute the results of data science.

1.3 CASE FOR DATA SCIENCE

In the past few decades, a massive accumulation of data has been seen with the advancement of information technology, connected networks, and the businesses it enables. This trend is also coupled with a steep decline in data storage and data processing costs. The applications built on these advancements like online businesses, social networking, and mobile technologies unleash a large amount of complex, heterogeneous data that are waiting to be analyzed. Traditional analysis techniques like dimensional slicing, hypothesis testing, and descriptive statistics can only go so far in information discovery. A paradigm is needed to manage the massive volume of data, explore the inter-relationships of thousands of variables, and deploy machine learning algorithms to deduce optimal insights from datasets. A set of frameworks, tools, and techniques are needed to intelligently assist humans to process all these data and extract valuable information (Piatetsky-Shapiro, Brachman, Khabaza, Kloesgen, & Simoudis, 1996). Data science is one such paradigm that can handle large volumes with multiple attributes and deploy complex algorithms to search for patterns from data. Each key motivation for using data science techniques are explored here.

1.3.1 Volume

The sheer volume of data captured by organizations is exponentially increasing. The rapid decline in storage costs and advancements in capturing every transaction and event, combined with the business need to extract as much leverage as possible using data, creates a strong motivation to store more data than ever. As data become more granular, the need to use large volume data to extract information increases. A rapid increase in the volume of data exposes the limitations of current analysis methodologies. In a few implementations, the time to create generalization models is critical and data volume plays a major part in determining the time frame of development and deployment.

1.3.2 Dimensions

The three characteristics of the Big Data phenomenon are high volume, high velocity, and high variety. The variety of data relates to the multiple types of values (numerical, categorical), formats of data (audio files, video files), and the application of the data (location coordinates, graph data). Every single record or data point contains multiple attributes or variables to provide context for the record. For example, every user record of an ecommerce site can contain attributes such as products viewed, products purchased, user demographics, frequency of purchase, clickstream, etc. Determining the most effective offer for an ecommerce user can involve computing information across

these attributes. Each attribute can be thought of as a dimension in the data space. The user record has multiple attributes and can be visualized in multi-dimensional space. The addition of each dimension increases the complexity of analysis techniques.

A simple linear regression model that has one input dimension is relatively easy to build compared to multiple linear regression models with multiple dimensions. As the dimensional space of data increase, a scalable framework that can work well with multiple data types and multiple attributes is needed. In the case of text mining, a document or article becomes a data point with each unique word as a dimension. Text mining yields a dataset where the number of attributes can range from a few hundred to hundreds of thousands of attributes.

1.3.3 Complex Questions

As more complex data are available for analysis, the complexity of information that needs to get extracted from data is increasing as well. If the natural clusters in a dataset, with hundreds of dimensions, need to be found, then traditional analysis like hypothesis testing techniques cannot be used in a scalable fashion. The machine-learning algorithms need to be leveraged in order to automate searching in the vast search space.

Traditional statistical analysis approaches the data analysis problem by assuming a stochastic model, in order to predict a response variable based on a set of input variables. A linear regression is a classic example of this technique where the parameters of the model are estimated from the data. These hypothesis-driven techniques were highly successful in modeling simple relationships between response and input variables. However, there is a significant need to extract nuggets of information from large, complex datasets, where the use of traditional statistical data analysis techniques is limited (Breiman, 2001)

Machine learning approaches the problem of modeling by trying to find an algorithmic model that can better predict the output from input variables. The algorithms are usually recursive and, in each cycle, estimate the output and "learn" from the predictive errors of the previous steps. This route of modeling greatly assists in exploratory analysis since the approach here is not validating a hypothesis but generating a multitude of hypotheses for a given problem. In the context of the data problems faced today, both techniques need to be deployed. John Tuckey, in his article "We need both exploratory and confirmatory," stresses the importance of both exploratory and confirmatory analysis techniques (Tuckey, 1980). In this book, a range of data science techniques, from traditional statistical modeling techniques like regressions to the modern machine learning algorithms are discussed.

1.4 DATA SCIENCE CLASSIFICATION

Data science problems can be broadly categorized into *supervised* or *unsupervised* learning models. Supervised or directed data science tries to infer a function or relationship based on labeled training data and uses this function to map new unlabeled data. Supervised techniques predict the value of the output variables based on a set of input variables. To do this, a model is developed from a *training* dataset where the values of input and output are previously known. The model generalizes the relationship between the input and output variables and uses it to predict for a dataset where only input variables are known. The output variable that is being predicted is also called a class label or target variable. Supervised data science needs a sufficient number of labeled records to learn the model from the data. Unsupervised or undirected data science uncovers hidden patterns in unlabeled data. In unsupervised data science, there are no output variables to predict. The objective of this class of data science techniques, is to find patterns in data based on the relationship between data points themselves. An application can employ both supervised and unsupervised learners.

Data science problems can also be classified into tasks such as: classification, regression, association analysis, clustering, anomaly detection, recommendation engines, feature selection, time series forecasting, deep learning, and text mining (Fig. 1.4). This book is organized around these data science tasks. An overview is presented in this chapter and an in-depth discussion of the

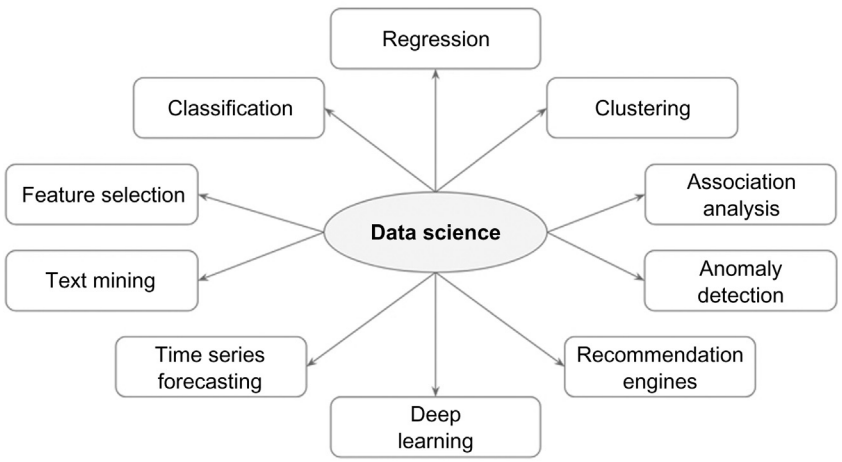

FIGURE 1.4
Data science tasks.

concepts and step-by-step implementations of many important techniques will be provided in the upcoming chapters.

Classification and *regression* techniques predict a target variable based on input variables. The prediction is based on a generalized model built from a previously known dataset. In regression tasks, the output variable is numeric (e.g., the mortgage interest rate on a loan). Classification tasks predict output variables, which are categorical or polynomial (e.g., the yes or no decision to approve a loan). *Deep learning* is a more sophisticated artificial neural network that is increasingly used for classification and regression problems. *Clustering* is the process of identifying the natural groupings in a dataset. For example, clustering is helpful in finding natural clusters in customer datasets, which can be used for market segmentation. Since this is unsupervised data science, it is up to the end user to investigate why these clusters are formed in the data and generalize the uniqueness of each cluster. In retail analytics, it is common to identify pairs of items that are purchased together, so that specific items can be bundled or placed next to each other. This task is called market basket analysis or *association analysis*, which is commonly used in cross selling. *Recommendation engines* are the systems that recommend items to the users based on individual user preference.

Anomaly or outlier detection identifies the data points that are significantly different from other data points in a dataset. Credit card transaction fraud detection is one of the most prolific applications of anomaly detection. *Time series forecasting* is the process of predicting the future value of a variable (e.g., temperature) based on past historical values that may exhibit a trend and seasonality. *Text mining* is a data science application where the input data is text, which can be in the form of documents, messages, emails, or web pages. To aid the data science on text data, the text files are first converted into document vectors where each unique word is an attribute. Once the text file is converted to document vectors, standard data science tasks such as classification, clustering, etc., can be applied. *Feature selection* is a process in which attributes in a dataset are reduced to a few attributes that really matter.

A complete data science application can contain elements of both supervised and unsupervised techniques (Tan et al., 2005). Unsupervised techniques provide an increased understanding of the dataset and hence, are sometimes called descriptive data science. As an example of how both unsupervised and supervised data science can be combined in an application, consider the following scenario. In marketing analytics, clustering can be used to find the natural clusters in customer records. Each customer is assigned a cluster label at the end of the clustering process. A labeled customer dataset can now be used to develop a model that assigns a cluster label for any new customer record with a supervised classification technique.

1.5 DATA SCIENCE ALGORITHMS

An algorithm is a logical step-by-step procedure for solving a problem. In data science, it is the blueprint for how a particular data problem is solved. Many of the learning algorithms are recursive, where a set of steps are repeated many times until a limiting condition is met. Some algorithms also contain a random variable as an input and are aptly called *randomized algorithms*. A classification task can be solved using many different learning algorithms such as decision trees, artificial neural networks, *k*-NN, and even some regression algorithms. The choice of which algorithm to use depends on the type of dataset, objective, structure of the data, presence of outliers, available computational power, number of records, number of attributes, and so on. It is up to the data science practitioner to decide which algorithm (s) to use by evaluating the performance of multiple algorithms. There have been hundreds of algorithms developed in the last few decades to solve data science problems.

Data science algorithms can be implemented by custom-developed computer programs in almost any computer language. This obviously is a time-consuming task. In order to focus the appropriate amount of time on data and algorithms, data science tools or statistical programing tools, like R, RapidMiner, Python, SAS Enterprise Miner, etc., which can implement these algorithms with ease, can be leveraged. These data science tools offer a library of algorithms as functions, which can be interfaced through programming code or configurated through graphical user interfaces. Table 1.1 provides a summary of data science tasks with commonly used algorithmic techniques and example cases.

1.6 ROADMAP FOR THIS BOOK

It's time to explore data science techniques in more detail. The main body of this book presents: the concepts behind each data science algorithm and a practical implementation (or two) for each. The chapters do not have to be read in a sequence. For each algorithm, a general overview is first provided, and then the concepts and logic of the learning algorithm and how it works in plain language are presented. Later, how the algorithm can be implemented using RapidMiner is shown. RapidMiner is a widely known and used software tool for data science (Piatetsky, 2018) and it has been chosen particularly for ease of implementation using GUI, and because it is available to use free of charge, as an open source data science tool. Each chapter is

Table 1.1 Data Science Tasks and Examples

Tasks	Description	Algorithms	Examples
Classification	Predict if a data point belongs to one of the predefined classes. The prediction will be based on learning from a known dataset	Decision trees, neural networks, Bayesian models, induction rules, *k*-nearest neighbors	Assigning voters into known buckets by political parties, e.g., soccer moms Bucketing new customers into one of the known customer groups
Regression	Predict the numeric target label of a data point. The prediction will be based on learning from a known dataset	Linear regression, logistic regression	Predicting the unemployment rate for the next year Estimating insurance premium
Anomaly detection	Predict if a data point is an outlier compared to other data points in the dataset	Distance-based, density-based, LOF	Detecting fraudulent credit card transactions and network intrusion
Time series forecasting	Predict the value of the target variable for a future timeframe based on historical values	Exponential smoothing, ARIMA, regression	Sales forecasting, production forecasting, virtually any growth phenomenon that needs to be extrapolated
Clustering	Identify natural clusters within the dataset based on inherit properties within the dataset	*k*-Means, density-based clustering (e.g., DBSCAN)	Finding customer segments in a company based on transaction, web, and customer call data
Association analysis	Identify relationships within an item set based on transaction data	FP-growth algorithm, a priori algorithm	Finding cross-selling opportunities for a retailer based on transaction purchase history
Recommendation engines	Predict the preference of an item for a user	Collaborative filtering, content-based filtering, hybrid recommenders	Finding the top recommended movies for a user

LOF, *local outlier factor;* ARIMA, *autoregressive integrated moving average;* DBSCAN, *density-based spatial clustering of applications with noise;* FP, *frequent pattern.*

concluded with some closing thoughts and further reading materials and references are listed. Here is a roadmap of the book.

1.6.1 Getting Started With Data Science

Successfully uncovering patterns in a dataset is an iterative process. Chapter 2, Data Science Process, provides a framework to solve the data science problems. A five-step process outlined in this chapter provides guidelines on gathering subject matter expertise; exploring the data with statistics and visualization; building a model using data science algorithms; testing

and deploying the model in the production environment; and finally reflecting on new knowledge gained in the cycle.

Simple data exploration either visually or with the help of basic statistical analysis can sometimes answer seemingly tough questions meant for data science. Chapter 3, Data Exploration, covers some of the basic tools used in knowledge discovery before deploying data science techniques. These practical tools increase one's understanding of data and are quite essential in understanding the results of the data science process.

1.6.2 Practice using RapidMiner

Before delving into the key data science techniques and algorithms, two specific things should be noted regarding how data science algorithms can be implemented while reading this book. It is believed that learning the concepts and implementing them enhances the learning experience. First, it is recommended that the free version of RapidMiner Studio software is downloaded from http://www.rapidminer.com and second, the first few sections of Chapter 15: Getting started with RapidMiner, should be reviewed in order to become familiar with the features of the tool, its basic operations, and the user interface functionality. Acclimating with RapidMiner will be helpful while using the algorithms that are discussed in this book. Chapter 15: Getting started with RapidMiner, is set at the end of the book because some of the later sections in the chapter build on the material presented in the chapters on tasks; however, the first few sections are a good starting point to become familiar with the tool.

Each chapter has a dataset used to describe the concept of a particular data science task and in most cases the same dataset is used for the implementation. The step-by-step instructions on practicing data science using the dataset are covered in every algorithm. All the implementations discussed are available at the companion website of the book at www.IntroDataScience.com. Though not required, it is advisable to access these files to as a learning aid. The dataset, complete RapidMiner processes (*.rmp files), and many more relevant electronic files can be downloaded from this website.

1.6.3 Core Algorithms

Classification is the most widely used data science task in business. The objective of a classification model is to predict a target variable that is binary (e.g., a loan decision) or categorical (e.g., a customer type) when a set of input variables are given. The model does this by learning the generalized relationship between the predicted target variable with all other input attributes

from a known dataset. There are several ways to skin this cat. Each algorithm differs by how the relationship is extracted from the known training dataset. Chapter 4, Classification, on classification addresses several of these methods.

- *Decision trees* approach the classification problem by partitioning the data into purer subsets based on the values of the input attributes. The attributes that help achieve the cleanest levels of such separation are considered significant in their influence on the target variable and end up at the root and closer-to-root levels of the tree. The output model is a tree framework than can be used for the prediction of new unlabeled data.
- *Rule induction* is a data science process of deducing "if–then" rules from a dataset or from the decision trees. These symbolic decision rules explain an inherent relationship between the input attributes and the target labels in the dataset that can be easily understood by anyone.
- *Naïve Bayesian* algorithms provide a probabilistic way of building a model. This approach calculates the probability for each value of the class variable for given values of input variables. With the help of conditional probabilities, for a given unseen record, the model calculates the outcome of all values of target classes and comes up with a predicted winner.
- Why go through the trouble of extracting complex relationships from the data when the entire training dataset can be memorized and the relationship can appear to have been generalized? This is exactly what the *k-NN* algorithm does, and it is, therefore, called a "lazy" learner where the entire training dataset is memorized as the model.
- Neurons are the nerve cells that connect with each other to form a biological neural network in our brain. The working of these interconnected nerve cells inspired the approach of some complex data problems by the creation of *artificial neural networks*. The neural networks section provides a conceptual background of how a simple neural network works and how to implement one for any general prediction problem. Later on we extend this to deep neural networks which have revolutionized the field of artificial intelligence.
- *Support vector machines (SVMs)* were developed to address optical character recognition problems: how can an algorithm be trained to detect boundaries between different patterns, and thus, identify characters? SVMs can, therefore, identify if a given data sample belongs within a boundary (in a particular class) or outside it (not in the class).
- *Ensemble learners* are "meta" models where the model is a combination of several different individual models. If certain conditions are met, ensemble learners can gain from the wisdom of crowds and greatly reduce the generalization error in data science.

The simple mathematical equation $y = ax + b$ is a linear regression model. Chapter 5, Regression Methods, describes a class of data science techniques in which the target variable (e.g., interest rate or a target class) is functionally related to input variables.

- *Linear regression*: The simplest of all function fitting models is based on a linear equation, as previously mentioned. Polynomial regression uses higher-order equations. No matter what type of equation is used, the goal is to represent the variable to be predicted in terms of other variables or attributes. Further, the predicted variable and the independent variables all have to be numeric for this to work. The basics of building regression models will be explored and how predictions can be made using such models will be shown.
- *Logistic regression*: Addresses the issue of predicting a target variable that may be binary or binomial (such as 1 or 0, yes or no) using predictors or attributes, which may be numeric.

Supervised data science or directed data science predict the value of the target variables. Two important *unsupervised* data science tasks will be reviewed: Association Analysis in Chapter 6 and Clustering in Chapter 7. Ever heard of the beer and diaper association in supermarkets? Apparently, a supermarket discovered that customers who buy diapers also tend to buy beer. While this may have been an urban legend, the observation has become a poster child for association analysis. Associating an item in a transaction with another item in the transaction to determine the most frequently occurring patterns is termed *association analysis*. This technique is about, for example, finding relationships between products in a supermarket based on purchase data, or finding related web pages in a website based on clickstream data. It is widely used in retail, ecommerce, and media to creatively bundle products.

Clustering is the data science task of identifying natural groups in the data. As an unsupervised task, there is no target class variable to predict. After the clustering is performed, each record in the dataset is associated with one or more cluster. Widely used in marketing segmentations and text mining, clustering can be performed by a range of algorithms. In Chapter 7, Clustering, three common algorithms with diverse identification approaches will be discussed. The *k-means clustering* technique identifies a cluster based on a central prototype record. *DBSCAN* clustering partitions the data based on variation in the density of records in a dataset. *Self-organizing maps* create a two-dimensional grid where all the records related with each other are placed next to each other.

How to determine which algorithms work best for a given dataset? Or for that matter how to objectively quantify the performance of any algorithm on a dataset? These questions are addressed in Chapter 8, Model Evaluation,

which covers performance evaluation. The most commonly used tools for evaluating classification models such as a confusion matrix, ROC curves, and lift charts are described.

Chapter 9, Text Mining, provides a detailed look into the area of text mining and text analytics. It starts with a background on the origins of text mining and provides the motivation for this fascinating topic using the example of IBM's Watson, the Jeopardy—winning computer program that was built using concepts from text and data mining. The chapter introduces some key concepts important in the area of text analytics such as term frequency—inverse document frequency scores. Finally, it describes two case studies in which it is shown how to implement text mining for document clustering and automatic classification based on text content.

Chapter 10, Deep Learning, describes a set of algorithms to model high level abstractions in data. They are increasingly applied to image processing, speech recognition, online advertisements, and bioinformatics. This chapter covers the basic concepts of deep learning, popular use cases, and a sample classification implementation.

The advent of digital economy exponentially increased the choices of available products to the customer which can be overwhelming. Personalized recommendation lists help by narrowing the choices to a few items relevant to a particular user and aid users in making final consumption decisions. Recommendation engines, covered in Chapter 11, are the most prolific utilities of machine learning in everyday experience. Recommendation engines are a class of machine learning techniques that predict a user preference for an item. There are a wide range of techniques available to build a recommendation engine. This chapter discusses the most common methods starting with *collaborative filtering* and *content-based filtering* concepts and implementations using a practical dataset.

Forecasting is a common application of time series analysis. Companies use sales forecasts, budget forecasts, or production forecasts in their planning cycles. Chapter 12 on Time Series Forecasting starts by pointing out the distinction between standard supervised predictive models and time series forecasting models. The chapter covers a few time series forecasting methods, starting with time series decomposition, moving averages, exponential smoothing, regression, ARIMA methods, and machine learning based methods using windowing techniques.

Chapter 13 on Anomaly Detection describes how outliers in data can be detected by combining multiple data science tasks like classification, regression, and clustering. The fraud alert received from credit card companies is the result of an anomaly detection algorithm. The target variable to be

predicted is whether a transaction is an outlier or not. Since clustering tasks identify outliers as a cluster, distance-based and density-based clustering techniques can be used in anomaly detection tasks.

In data science, the objective is to develop a representative model to generalize the relationship between input attributes and target attributes, so that we can predict the value or class of the target variables. Chapter 14, Feature Selection, introduces a preprocessing step that is often critical for a successful predictive modeling exercise: *feature selection*. Feature selection is known by several alternative terms such as attribute weighting, dimension reduction, and so on. There are two main styles of feature selection: filtering the key attributes before modeling (filter style) or selecting the attributes during the process of modeling (wrapper style). A few filter-based methods such as principal component analysis (PCA), information gain, and chi-square, and a couple of wrapper-type methods like forward selection and backward elimination will be discussed.

The first few sections of Chapter 15, Getting Started with RapidMiner, should provide a good overview for getting familiar with RapidMiner, while the latter sections of this chapter discuss some of the commonly used productivity tools and techniques such as data transformation, missing value handling, and process optimizations using RapidMiner.

References

Breiman, L. (2001). Statistical modeling: Two cultures. *Statistical Science, 6*(3), 199–231.

Fayyad, U., Piatetsky-shapiro, G., & Smyth, P. (1996). From data science to knowledge discovery in databases. *AI Magazine, 17*(3), 37–54.

Parr Rud, O. (2001). *Data science Cookbook*. New York: John Wiley and Sons.

Piatetsky, G. (2018). Top Software for Analytics, Data Science, Machine Learning in 2018: Trends and Analysis. Retrieved July 7, 2018, from https://www.kdnuggets.com/2018/05/poll-tools-analytics-data-science-machine-learning-results.html.

Piatetsky-Shapiro, G., Brachman, R., Khabaza, T., Kloesgen, W., & Simoudis, E. (1996). An overview of issues in developing industrial data science and knowledge discovery applications. In: *KDD-96 conference proceedings. KDD-96 conference proceedings*.

Rexer, K. (2013). *2013 Data miner survey summary report*. Winchester, MA: Rexer Analytics. <www.rexeranalytics.com>.

Tan, P.-N., Michael, S., & Kumar, V. (2005). *Introduction to data science*. Boston, MA: Addison-Wesley.

Tuckey, J. (1980). We need exploratory and Confirmatory. *The American Statistician, 34*(1), 23–25.

CHAPTER 2

Data Science Process

The methodical discovery of useful relationships and patterns in data is enabled by a set of iterative activities collectively known as the data science process. The standard data science process involves (1) understanding the problem, (2) preparing the data samples, (3) developing the model, (4) applying the model on a dataset to see how the model may work in the real world, and (5) deploying and maintaining the models. Over the years of evolution of data science practices, different frameworks for the process have been put forward by various academic and commercial bodies. The framework put forward in this chapter is synthesized from a few data science frameworks and is explained using a simple example dataset. This chapter serves as a high-level roadmap to building deployable data science models, and discusses the challenges faced in each step and the pitfalls to avoid.

One of the most popular data science process frameworks is Cross Industry Standard Process for Data Mining (CRISP-DM), which is an acronym for Cross Industry Standard Process for Data Mining. This framework was developed by a consortium of companies involved in data mining (Chapman et al., 2000). The CRISP-DM process is the most widely adopted framework for developing data science solutions. Fig. 2.1 provides a visual overview of the CRISP-DM framework. Other data science frameworks are SEMMA, an acronym for Sample, Explore, Modify, Model, and Assess, developed by the SAS Institute (SAS Institute, 2013); DMAIC, is an acronym for Define, Measure, Analyze, Improve, and Control, used in Six Sigma practice (Kubiak & Benbow, 2005); and the Selection, Preprocessing, Transformation, Data Mining, Interpretation, and Evaluation framework used in the knowledge discovery in databases process (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). All these frameworks exhibit common characteristics, and hence, a generic framework closely resembling the CRISP process will be used. As with any process framework, a data science process recommends the execution of a certain set of tasks to achieve optimal output. However, the process of extracting information and knowledge from the data is *iterative*. The steps within the data science process are not linear and have to undergo many

Data Science. DOI: https://doi.org/10.1016/B978-0-12-814761-0.00002-2

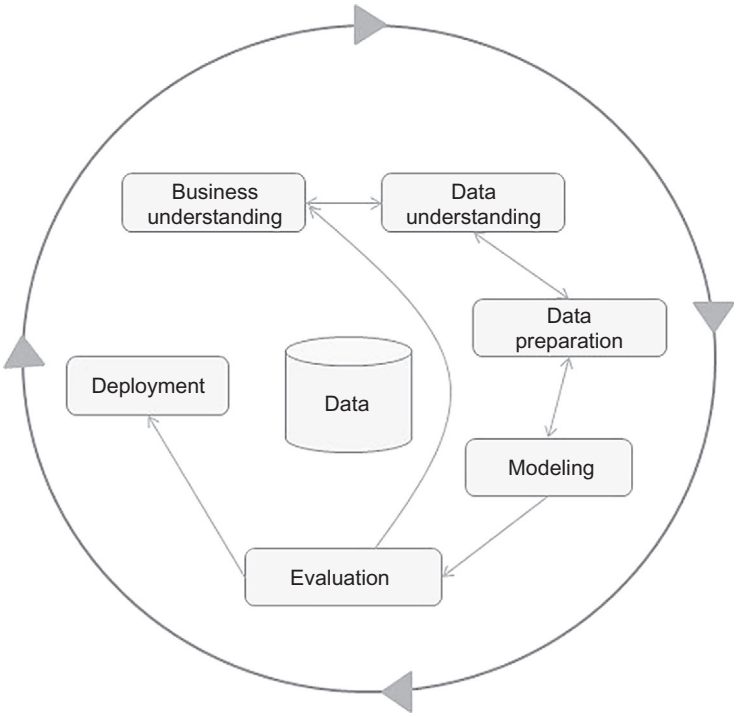

FIGURE 2.1
CRISP data mining framework.

loops, go back and forth between steps, and at times go back to the first step to redefine the data science problem statement.

The data science process presented in Fig. 2.2 is a generic set of steps that is problem, algorithm, and, data science tool agnostic. The fundamental objective of any process that involves data science is to address the analysis question. The problem at hand could be a segmentation of customers, a prediction of climate patterns, or a simple data exploration. The learning algorithm used to solve the business question could be a decision tree, an artificial neural network, or a scatterplot. The software tool to develop and implement the data science algorithm used could be custom coding, RapidMiner, R, Weka, SAS, Oracle Data Miner, Python, etc., (Piatetsky, 2018) to mention a few.

Data science, specifically in the context of big data, has gained importance in the last few years. Perhaps the most visible and discussed part of data science is the third step: *modeling*. It is the process of building representative models that can be inferred from the sample dataset which can be used for either predicting (*predictive modeling*) or describing the underlying pattern in the data (*descriptive or explanatory modeling*). Rightfully so, there is plenty of

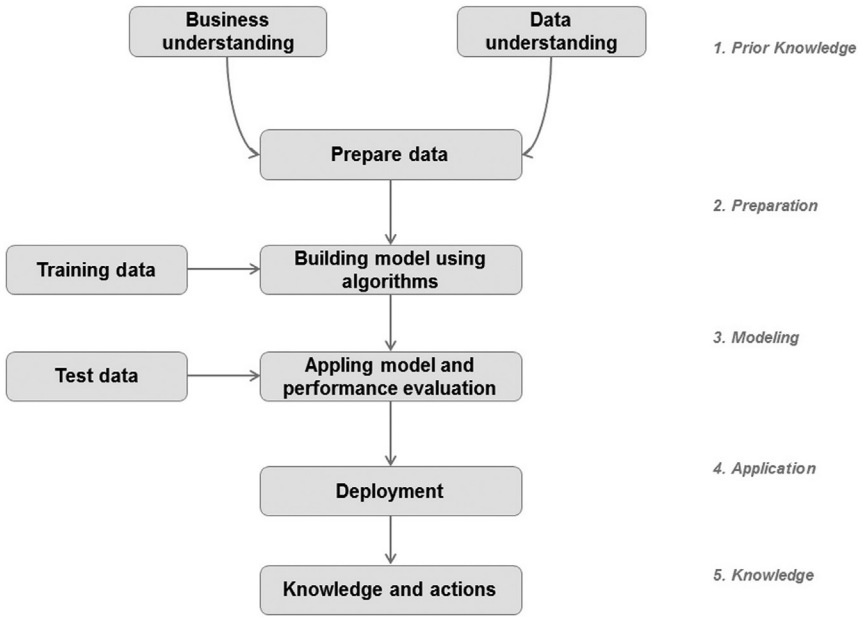

FIGURE 2.2
Data science process.

academic and business research in the modeling step. Most of this book has been dedicated to discussing various algorithms and the quantitative foundations that go with it. However, emphasis should be placed on considering data science as an end-to-end, multi-step, iterative process instead of just a model building step. Seasoned data science practitioners can attest to the fact that the most time-consuming part of the overall data science process is not the model building part, but the preparation of data, followed by data and business understanding. There are many data science tools, both open source and commercial, available on the market that can automate the model building. Asking the right business question, gaining in-depth business understanding, sourcing and preparing the data for the data science task, mitigating implementation considerations, integrating the model into the business process, and, most useful of all, gaining knowledge from the dataset, remain crucial to the success of the data science process. It's time to get started with Step 1: Framing the data science question and understanding the context.

2.1 PRIOR KNOWLEDGE

Prior knowledge refers to information that is already known about a subject. The data science problem doesn't emerge in isolation; it always develops on

top of existing subject matter and contextual information that is already known. The prior knowledge step in the data science process helps to define what problem is being solved, how it fits in the business context, and what data is needed in order to solve the problem.

2.1.1 Objective

The data science process starts with a need for analysis, a question, or a business objective. This is possibly the most important step in the data science process (Shearer, 2000). Without a well-defined statement of the problem, it is impossible to come up with the right dataset and pick the right data science algorithm. As an iterative process, it is common to go back to previous data science process steps, revise the assumptions, approach, and tactics. However, it is imperative to get the first step—the objective of the whole process—right.

The data science process is going to be explained using a hypothetical use case. Take the consumer loan business for example, where a loan is provisioned for individuals against the collateral of assets like a home or car, that is, a mortgage or an auto loan. As many homeowners know, an important component of the loan, for the borrower and the lender, is the interest rate at which the borrower repays the loan on top of the principal. The interest rate on a loan depends on a gamut of variables like the current federal funds rate as determined by the central bank, borrower's credit score, income level, home value, initial down payment amount, current assets and liabilities of the borrower, etc. The key factor here is whether the lender sees enough reward (interest on the loan) against the risk of losing the principal (borrower's default on the loan). In an individual case, the status of default of a loan is Boolean; either one defaults or not, during the period of the loan. But, in a group of tens of thousands of borrowers, the default rate can be found—a continuous numeric variable that indicates the percentage of borrowers who default on their loans. All the variables related to the borrower like credit score, income, current liabilities, etc., are used to assess the default risk in a related group; based on this, the interest rate is determined for a loan. The business objective of this hypothetical case is: *If the interest rate of past borrowers with a range of credit scores is known, can the interest rate for a new borrower be predicted?*

2.1.2 Subject Area

The process of data science uncovers hidden patterns in the dataset by exposing relationships between attributes. But the problem is that it uncovers a lot of patterns. The false or spurious signals are a major problem in the data science process. It is up to the practitioner to sift through the exposed patterns and accept the ones that are valid and relevant to the answer of the objective

question. Hence, it is essential to know the subject matter, the context, and the business process generating the data.

The lending business is one of the oldest, most prevalent, and complex of all the businesses. If the objective is to predict the lending interest rate, then it is important to know how the lending business works, why the prediction matters, what happens after the rate is predicted, what data points can be collected from borrowers, what data points cannot be collected because of the external regulations and the internal policies, what other external factors can affect the interest rate, how to verify the validity of the outcome, and so forth. Understanding current models and business practices lays the foundation and establishes known knowledge. Analysis and mining the data provides the new knowledge that can be built on top of the existing knowledge (Lidwell, Holden, & Butler, 2010).

2.1.3 Data

Similar to the prior knowledge in the subject area, prior knowledge in the data can also be gathered. Understanding how the data is collected, stored, transformed, reported, and used is essential to the data science process. This part of the step surveys all the data available to answer the business question and narrows down the new data that need to be sourced. There are quite a range of factors to consider: quality of the data, quantity of data, availability of data, gaps in data, does lack of data compel the practitioner to change the business question, etc. The objective of this step is to come up with a dataset to answer the business question through the data science process. It is critical to recognize that an inferred model is only as good as the data used to create it.

For the lending example, a sample dataset of ten data points with three attributes has been put together: identifier, credit score, and interest rate. First, some of the terminology used in the data science process are discussed.

- A *dataset* (*example set*) is a collection of data with a defined structure. Table 2.1 shows a dataset. It has a well-defined structure with 10 rows and 3 columns along with the column headers. This structure is also sometimes referred to as a "data frame".
- A *data point* (*record*, *object* or *example*) is a single instance in the dataset. Each row in Table 2.1 is a data point. Each instance contains the same structure as the dataset.
- An *attribute* (*feature*, *input*, *dimension*, *variable*, or *predictor*) is a single property of the dataset. Each column in Table 2.1 is an attribute. Attributes can be numeric, categorical, date-time, text, or Boolean *data types*. In this example, both the credit score and the interest rate are numeric attributes.

Table 2.1 Dataset

Borrower ID	Credit Score	Interest Rate (%)
01	500	7.31
02	600	6.70
03	700	5.95
04	700	6.40
05	800	5.40
06	800	5.70
07	750	5.90
08	550	7.00
09	650	6.50
10	825	5.70

Table 2.2 New Data With Unknown Interest Rate

Borrower ID	Credit Score	Interest Rate
11	625	?

- A *label* (*class label*, *output*, *prediction*, *target*, or *response*) is the special attribute to be predicted based on all the input attributes. In Table 2.1, the interest rate is the output variable.
- *Identifiers* are special attributes that are used for locating or providing context to individual records. For example, common attributes like names, account numbers, and employee ID numbers are identifier attributes. Identifiers are often used as lookup keys to join multiple datasets. They bear no information that is suitable for building data science models and should, thus, be excluded for the actual modeling step. In Table 2.1, the attribute ID is the identifier.

2.1.4 Causation Versus Correlation

Suppose the business question is inverted: *Based on the data in* Table 2.1, *can the credit score of the borrower be predicted based on interest rate?* The answer is yes—but it does not make business sense. From the existing domain expertise, it is known that credit score *influences* the loan interest rate. Predicting credit score based on interest rate inverses the direction of the causal relationship. This question also exposes one of the key aspects of model building. The correlation between the input and output attributes doesn't guarantee causation. Hence, it is important to frame the data science question correctly using the existing domain and data knowledge. In this data science example, the interest rate of the new borrower with an unknown interest rate will be predicted (Table 2.2) based on the pattern learned from known data in Table 2.1.

2.2 DATA PREPARATION

Preparing the dataset to suit a data science task is the most time-consuming part of the process. It is extremely rare that datasets are available in the form required by the data science algorithms. Most of the data science algorithms would require data to be structured in a tabular format with records in the rows and attributes in the columns. If the data is in any other format, the data would need to be transformed by applying pivot, type conversion, join, or transpose functions, etc., to condition the data into the required structure.

2.2.1 Data Exploration

Data preparation starts with an in-depth exploration of the data and gaining a better understanding of the dataset. Data exploration, also known as *exploratory data analysis*, provides a set of simple tools to achieve basic understanding of the data. Data exploration approaches involve computing descriptive statistics and visualization of data. They can expose the structure of the data, the distribution of the values, the presence of extreme values, and highlight the inter-relationships within the dataset. Descriptive statistics like mean, median, mode, standard deviation, and range for each attribute provide an easily readable summary of the key characteristics of the distribution of data. On the other hand, a visual plot of data points provides an instant grasp of all the data points condensed into one chart. Fig. 2.3 shows the scatterplot of credit score vs. loan interest rate and it can be observed that as credit score increases, interest rate decreases.

2.2.2 Data Quality

Data quality is an ongoing concern wherever data is collected, processed, and stored. In the interest rate dataset (Table 2.1), how does one know if the credit score and interest rate data are accurate? What if a credit score has a recorded value of 900 (beyond the theoretical limit) or if there was a data entry error? Errors in data will impact the representativeness of the model. Organizations use data alerts, cleansing, and transformation techniques to improve and manage the quality of the data and store them in companywide repositories called *data warehouses*. Data sourced from well-maintained data warehouses have higher quality, as there are proper controls in place to ensure a level of data accuracy for new and existing data. The data cleansing practices include elimination of duplicate records, quarantining outlier records that exceed the bounds, standardization of attribute values, substitution of missing values, etc. Regardless, it is critical to check the data using data exploration techniques in addition to using prior knowledge of the data and business before building models.

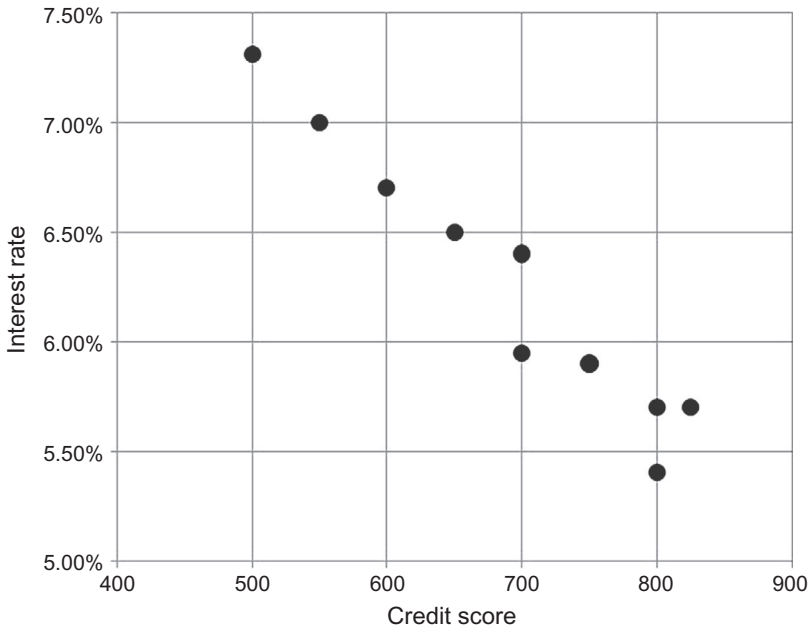

FIGURE 2.3
Scatterplot for interest rate dataset.

2.2.3 Missing Values

One of the most common data quality issues is that some records have missing attribute values. For example, a credit score may be missing in one of the records. There are several different mitigation methods to deal with this problem, but each method has pros and cons. The first step of managing missing values is to understand the reason behind why the values are missing. Tracking the data lineage (provenance) of the data source can lead to the identification of systemic issues during data capture or errors in data transformation. Knowing the source of a missing value will often guide which mitigation methodology to use. The missing value can be substituted with a range of artificial data so that the issue can be managed with marginal impact on the later steps in the data science process. Missing credit score values can be replaced with a credit score derived from the dataset (mean, minimum, or maximum value, depending on the characteristics of the attribute). This method is useful if the missing values occur randomly and the frequency of occurrence is quite rare. Alternatively, to build the representative model, all the data records with missing values or records with poor data quality can be ignored. This method reduces the size of the dataset. Some data science algorithms are good at handling records with missing values, while others expect the data preparation step to handle it before the model is

inferred. For example, k-nearest neighbor (k-NN) algorithm for classification tasks are often robust with missing values. Neural network models for classification tasks do not perform well with missing attributes, and thus, the data preparation step is essential for developing neural network models.

2.2.4 Data Types and Conversion

The attributes in a dataset can be of different types, such as continuous numeric (interest rate), integer numeric (credit score), or categorical. For example, the credit score can be expressed as categorical values (poor, good, excellent) or numeric score. Different data science algorithms impose different restrictions on the attribute data types. In case of linear regression models, the input attributes have to be numeric. If the available data are categorical, they must be converted to continuous numeric attribute. A specific numeric score can be encoded for each category value, such as poor = 400, good = 600, excellent = 700, etc. Similarly, numeric values can be converted to categorical data types by a technique called *binning*, where a range of values are specified for each category, for example, a score between 400 and 500 can be encoded as "low" and so on.

2.2.5 Transformation

In some data science algorithms like k-NN, the input attributes are expected to be numeric and *normalized*, because the algorithm compares the values of different attributes and calculates distance between the data points. Normalization prevents one attribute dominating the distance results because of large values. For example, consider income (expressed in USD, in thousands) and credit score (in hundreds). The distance calculation will always be dominated by slight variations in income. One solution is to convert the range of income and credit score to a more uniform scale from 0 to 1 by normalization. This way, a consistent comparison can be made between the two different attributes with different units.

2.2.6 Outliers

Outliers are anomalies in a given dataset. Outliers may occur because of correct data capture (few people with income in tens of millions) or erroneous data capture (human height as 1.73 cm instead of 1.73 m). Regardless, the presence of outliers needs to be understood and will require special treatments. The purpose of creating a representative model is to generalize a pattern or a relationship within a dataset and the presence of outliers skews the representativeness of the inferred model. Detecting outliers may be the primary purpose of some data science applications, like fraud or intrusion detection.

2.2.7 Feature Selection

The example dataset shown in Table 2.1 has one *attribute* or *feature*—the credit score—and one *label*—the interest rate. In practice, many data science problems involve a dataset with hundreds to thousands of attributes. In text mining applications, every distinct word in a document forms a distinct attribute in the dataset. Not all the attributes are equally important or useful in predicting the target. The presence of some attributes might be counterproductive. Some of the attributes may be highly correlated with each other, like annual income and taxes paid. A large number of attributes in the dataset significantly increases the complexity of a model and may degrade the performance of the model due to the *curse of dimensionality*. In general, the presence of more detailed information is desired in data science because discovering nuggets of a pattern in the data is one of the attractions of using data science techniques. But, as the number of dimensions in the data increase, data becomes sparse in high-dimensional space. This condition degrades the reliability of the models, especially in the case of clustering and classification (Tan, Steinbach, & Kumar, 2005).

Reducing the number of attributes, without significant loss in the performance of the model, is called feature selection. It leads to a more simplified model and helps to synthesize a more effective explanation of the model.

2.2.8 Data Sampling

Sampling is a process of selecting a subset of records as a representation of the original dataset for use in data analysis or modeling. The sample data serve as a representative of the original dataset with similar properties, such as a similar mean. Sampling reduces the amount of data that need to be processed and speeds up the build process of the modeling. In most cases, to gain insights, extract the information, and to build representative predictive models it is sufficient to work with samples. Theoretically, the error introduced by sampling impacts the relevancy of the model, but their benefits far outweigh the risks.

In the build process for data science applications, it is necessary to segment the datasets into training and test samples. The training dataset is sampled from the original dataset using simple sampling or class label specific sampling. Consider the example cases for predicting anomalies in a dataset (e.g., predicting fraudulent credit card transactions). The objective of anomaly detection is to classify the outliers in the data. These are rare events and often the dataset does not have enough examples of the outlier class. *Stratified sampling* is a process of sampling where each class is equally represented in the sample; this allows the model to focus on the difference between the patterns of each class that is, normal and outlier records. In classification applications,

sampling is used create multiple base models, each developed using a different set of sampled training datasets. These base models are used to build one meta model, called the *ensemble model*, where the error rate is improved when compared to that of the base models.

2.3 MODELING

A model is the abstract representation of the data and the relationships in a given dataset. A simple rule of thumb like "*mortgage interest rate reduces with increase in credit score*" is a model; although there is not enough quantitative information to use in a production scenario, it provides directional information by abstracting the relationship between credit score and interest rate.

There are a few hundred data science algorithms in use today, derived from statistics, machine learning, pattern recognition, and the body of knowledge related to computer science. Fortunately, there are many viable commercial and open source data science tools on the market to automate the execution of these learning algorithms. As a data science practitioner, it is sufficient to an overview of the learning algorithm, how it works, and determining what parameters need to be configured based on the understanding of the business and data. Classification and regression tasks are predictive techniques because they predict an outcome variable based on one or more input variables. Predictive algorithms require a prior known dataset to learn the model. Fig. 2.4 shows the steps in the modeling phase of predictive data science. Association analysis and clustering are descriptive data science techniques where there is no target variable to predict; hence, there is no test dataset. However, both predictive and descriptive models have an evaluation step.

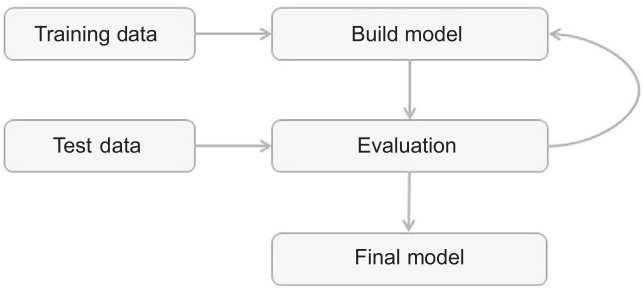

FIGURE 2.4
Modeling steps.

Table 2.3 Training Dataset

Borrower	Credit Score (*X*)	Interest Rate (*Y*) (%)
01	500	7.31
02	600	6.70
03	700	5.95
05	800	5.40
06	800	5.70
08	550	7.00
09	650	6.50

Table 2.4 Test Dataset

Borrower	Credit Score (*X*)	Interest Rate (*Y*)
04	700	6.40
07	750	5.90
10	825	5.70

2.3.1 Training and Testing Datasets

The modeling step creates a representative model inferred from the data. The dataset used to create the model, with known attributes and target, is called the *training dataset*. The validity of the created model will also need to be checked with another known dataset called the *test dataset* or *validation dataset*. To facilitate this process, the overall known dataset can be split into a training dataset and a test dataset. A standard rule of thumb is two-thirds of the data are to be used as training and one-third as a test dataset. Tables 2.3 and 2.4 show the random split of training and test data, based on the example dataset shown in Table 2.1. Fig. 2.5 shows the scatterplot of the entire example dataset with the training and test datasets marked.

2.3.2 Learning Algorithms

The business question and the availability of data will dictate what data science task (association, classification, regression, etc.,) can to be used. The practitioner determines the appropriate data science algorithm within the chosen category. For example, within a classification task many algorithms can be chosen from: decision trees, rule induction, neural networks, Bayesian models, k-NN, etc. Likewise, within decision tree techniques, there are quite a number of variations of learning algorithms like classification and regression tree (CART), CHi-squared Automatic Interaction Detector (CHAID) etc.

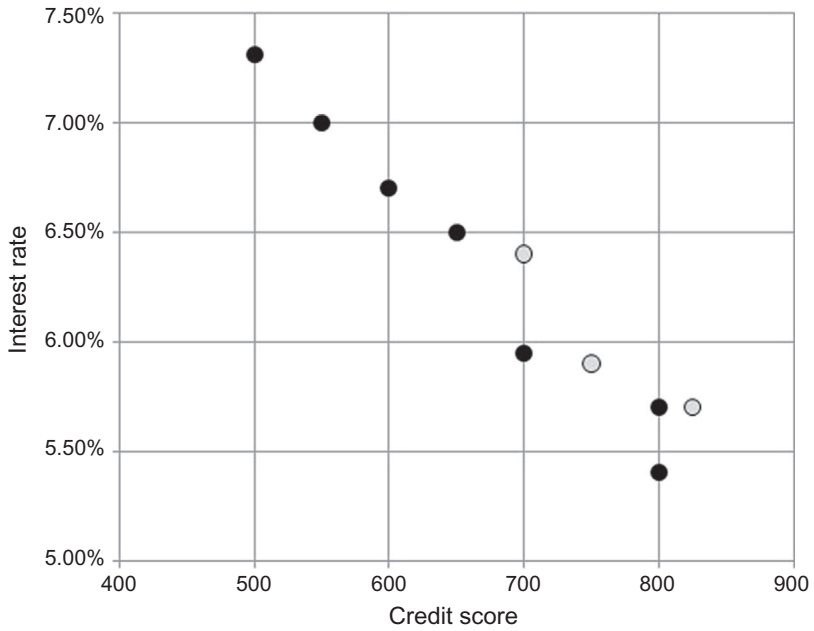

FIGURE 2.5
Scatterplot of training and test data.

These algorithms will be reviewed in detail in later chapters. It is not uncommon to use multiple data science tasks and algorithms to solve a business question.

Interest rate prediction is a regression problem. A simple linear regression technique will be used to model and generalize the relationship between credit score and interest rate. The training set of seven records is used to create the model and the test set of three records is used to evaluate the validity of the model.

The objective of simple linear regression can be visualized as fitting a straight line through the data points in a scatterplot (Fig. 2.6). The line has to be built in such a way that the sum of the squared distance from the data points to the line is minimal. The line can be expressed as:

$$y = a * x + b \tag{2.1}$$

where y is the output or dependent variable, x is the input or independent variable, b is the y-intercept, and a is the coefficient of x. The values of a and b can be found in such a way so as to minimize the sum of the squared residuals of the line.

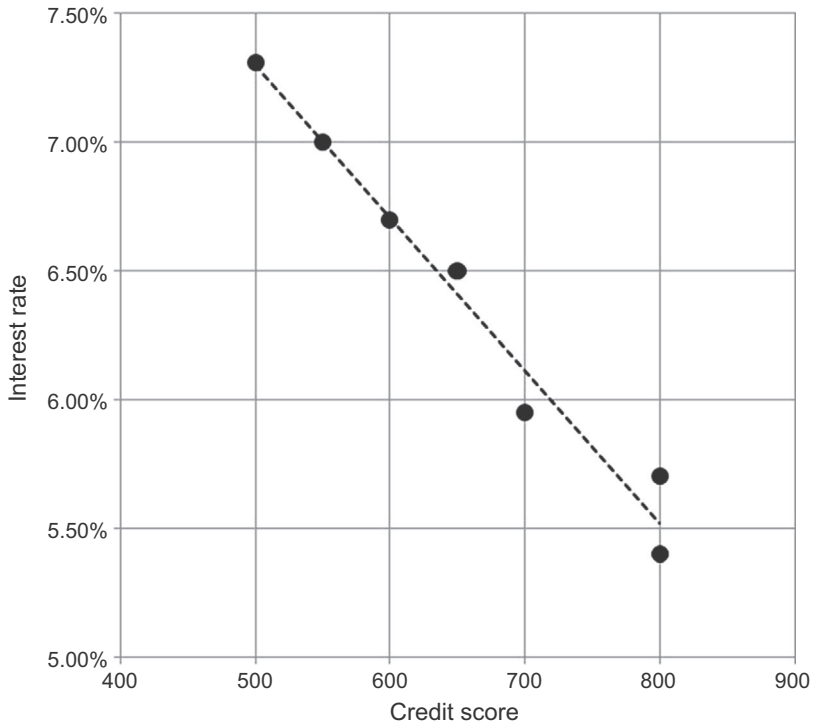

FIGURE 2.6
Regression model.

The line shown in Eq. (2.1) serves as a model to predict the outcome of new unlabeled datasets. For the interest rate dataset, the simple linear regression for the interest rate (y) has been calculated as (details in Chapter 5: Regression):

$$y = 0.1 + \frac{6}{100,000}x$$

$$\text{Interest rate} = 10 - \frac{6 \times \text{credit score}}{1000}$$

Using this model, the interest rate for a new borrower with a specific credit score can be calculated.

2.3.3 Evaluation of the Model

The model generated in the form of an equation is generalized and synthesized from seven training records. The credit score in the equation can be substituted to see if the model estimates the interest rate for each of the

Table 2.5 Evaluation of Test Dataset

Borrower	Credit Score (*X*)	Interest Rate (*Y*) (%)	Model Predicted (*Y*) (%)	Model Error (%)
04	700	6.40	6.11	− 0.29
07	750	5.90	5.81	− 0.09
10	825	5.70	5.37	− 0.33

seven training records. The estimation may not be exactly the same as the values in the training records. A model should not memorize and output the same values that are in the training records. The phenomenon of a model memorizing the training data is called *overfitting*. An overfitted model just memorizes the training records and will underperform on real unlabeled new data. The model should generalize or *learn* the relationship between credit score and interest rate. To evaluate this relationship, the validation or test dataset, which was not previously used in building the model, is used for evaluation, as shown in Table 2.5.

Table 2.5 provides the three testing records where the value of the interest rate is known; these records were not used to build the model. The actual value of the interest rate can be compared against the predicted value using the model, and thus, the *prediction error* can be calculated. As long as the error is acceptable, this model is ready for deployment. The error rate can be used to compare this model with other models developed using different algorithms like neural networks or Bayesian models, etc.

2.3.4 Ensemble Modeling

Ensemble modeling is a process where multiple diverse base models are used to predict an outcome. The motivation for using ensemble models is to reduce the generalization error of the prediction. As long as the base models are diverse and *independent*, the prediction error decreases when the ensemble approach is used. The approach seeks the wisdom of crowds in making a prediction. Even though the ensemble model has multiple base models within the model, it acts and performs as a single model. Most of the practical data science applications utilize ensemble modeling techniques.

At the end of the modeling stage of the data science process, one has (1) analyzed the business question; (2) sourced the data relevant to answer the question; (3) selected a data science technique to answer the question; (4) picked a data science algorithm and prepared the data to suit the algorithm; (5) split the data into training and test datasets; (6) built a generalized model from the training dataset; and (7) validated the model against the test

dataset. This model can now be used to predict the interest rate of new borrowers by integrating it in the actual loan approval process.

2.4 APPLICATION

Deployment is the stage at which the model becomes production ready or *live*. In business applications, the results of the data science process have to be assimilated into the business process—usually in software applications. The model deployment stage has to deal with: assessing model readiness, technical integration, response time, model maintenance, and assimilation.

2.4.1 Production Readiness

The production readiness part of the deployment determines the critical qualities required for the deployment objective. Consider two business use cases: determining whether a consumer qualifies for a loan and determining the groupings of customers for an enterprise by marketing function.

The consumer credit approval process is a real-time endeavor. Either through a consumer-facing website or through a specialized application for frontline agents, the credit decisions and terms need to be provided in real-time as soon as prospective customers provide the relevant information. It is optimal to provide a quick decision while also proving accurate. The decision-making model has to collect data from the customer, integrate third-party data like credit history, and make a decision on the loan approval and terms in a matter of seconds. The critical quality of this model deployment is real-time prediction.

Segmenting customers based on their relationship with the company is a thoughtful process where signals from various customer interactions are collected. Based on the patterns, similar customers are put in cohorts and campaign strategies are devised to best engage the customer. For this application, batch processed, time lagged data would suffice. The critical quality in this application is the ability to find unique patterns amongst customers, not the response time of the model. The business application informs the choices that need to be made in the data preparation and modeling steps.

2.4.2 Technical Integration

Currently, it is quite common to use data science automation tools or coding using R or Python to develop models. Data science tools save time as they do not require the writing of custom codes to execute the algorithm. This allows the analyst to focus on the data, business logic, and exploring patterns

from the data. The models created by data science tools can be ported to production applications by utilizing the Predictive Model Markup Language (PMML) (Guazzelli, Zeller, Lin, & Williams, 2009) or by invoking data science tools in the production application. PMML provides a portable and consistent format of model description which can be read by most data science tools. This allows the flexibility for practitioners to develop the model with one tool (e.g., RapidMiner) and deploy it in another tool or application. Some models such as simple regression, decision trees, and induction rules for predictive analytics can be incorporated directly into business applications and business intelligence systems easily. These models are represented by simple equations and the "if-then" rule, hence, they can be ported easily to most programming languages.

2.4.3 Response Time

Data science algorithms, like k-NN, are easy to build, but quite slow at predicting the unlabeled records. Algorithms such as the decision tree take time to build but are fast at prediction. There are trade-offs to be made between production responsiveness and modeling build time. The quality of prediction, accessibility of input data, and the response time of the prediction remain the critical quality factors in business application.

2.4.4 Model Refresh

The key criterion for the ongoing relevance of the model is the representativeness of the dataset it is processing. It is quite normal that the conditions in which the model is built change after the model is sent to deployment. For example, the relationship between the credit score and interest rate change frequently based on the prevailing macroeconomic conditions. Hence, the model will have to be refreshed frequently. The validity of the model can be routinely tested by using the new known test dataset and calculating the prediction error rate. If the error rate exceeds a particular threshold, then the model has to be refreshed and redeployed. Creating a maintenance schedule is a key part of a deployment plan that will sustain a relevant model.

2.4.5 Assimilation

In the descriptive data science applications, deploying a model to live systems may not be the end objective. The objective may be to assimilate the knowledge gained from the data science analysis to the organization. For example, the objective may be finding logical clusters in the customer database so that separate marketing approaches can be developed for each customer cluster. Then the next step may be a classification task for new customers to bucket them in one of known clusters. The association analysis

provides a solution for the market basket problem, where the task is to find which two products are purchased together most often. The challenge for the data science practitioner is to articulate these findings, establish relevance to the original business question, quantify the risks in the model, and quantify the business impact. This is indeed a challenging task for data science practitioners. The business user community is an amalgamation of different points of view, different quantitative mindsets, and skill sets. Not everyone is aware about the process of data science and what it can and cannot do. Some aspects of this challenge can be addressed by focusing on the end result, the impact of knowing the discovered information, and the follow-up actions, instead of the technical process of extracting the information through data science.

2.5 KNOWLEDGE

The data science process provides a framework to extract nontrivial information from data. With the advent of massive storage, increased data collection, and advanced computing paradigms, the available datasets to be utilized are only increasing. To extract knowledge from these massive data assets, advanced approaches need to be employed, like data science algorithms, in addition to standard business intelligence reporting or statistical analysis. Though many of these algorithms can provide valuable knowledge, it is up to the practitioner to skillfully transform a business problem to a data problem and apply the right algorithm. Data science, like any other technology, provides various options in terms of algorithms and parameters within the algorithms. Using these options to extract the right information from data is a bit of an art and can be developed with practice.

The data science process starts with prior knowledge and ends with posterior knowledge, which is the incremental insight gained. As with any quantitative technique, the data science process can bring up spurious irrelevant patterns from the dataset. Not all discovered patterns lead to incremental knowledge. Again, it is up to the practitioner to invalidate the irrelevant patterns and identify the meaningful information. The impact of the information gained through data science can be measured in an application. It is the difference between gaining the information through the data science process and the insights from basic data analysis. Finally, the whole data science process is a framework to invoke the right questions (Chapman et al., 2000) and provide guidance, through the right approaches, to solve a problem. It is not meant to be used as a set of rigid rules, but as a set of iterative, distinct steps that aid in knowledge discovery.

In the upcoming chapters, the details of key data science concepts along with their implementation will be explored. Exploring data using basic statistical and visual techniques are an important first step in preparing the data for data science. The next chapter on data exploration provides a practical tool kit to explore and understand data. The techniques of data preparation are explained in the context of individual data science algorithms in the chapters on classification, association analysis, clustering, text mining, time series, and anomaly detection.

References

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0: Step-by-step data mining guide*. SPSS Inc. Retrieved from <ftp://ftp.software.ibm.com/software/analytics/spss/support/Modeler/Documentation/14/UserManual/CRISP-DM.pdf>.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, *17*(3), 37–54.

Guazzelli, A., Zeller, M., Lin, W., & Williams, G. (2009). PMML: An open standard for sharing models. *The R Journal*, *1*(1), 60–65.

Kubiak, T., & Benbow, D. W. (2005). *The certified six sigma black belt handbook*. Milwaukee, WI: ASQQuality Press.

Lidwell, W., Holden, K., & Butler, J. (2010). *Universal principles of design, revised and updated: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design*. Beverly, MA: Rockport Publishers.

Piatetsky, G. (2018). *Top software for analytics, data science, machine learning in 2018: Trends and analysis*. Retrieved from <https://www.kdnuggets.com/2018/05/poll-tools-analytics-data-science-machine-learning-results.html> Accessed 07.07.18.

SAS Institute. (2013). *Getting started with SAS enterprise miner 12.3*.

Shearer, C. (2000). The CRISP-DM model: The new blueprint for data mining. *Journal of Data Warehousing*, *5*(4), 13–22.

Tan, P.-N., Steinbach, M., & Kumar, V. (2005). Introduction to data mining. *Journal of School Psychology*, *19*, 51–56. Available from https://doi.org/10.1016/0022-4405(81)90007-8.

Weisstein, E. W. (2013). Retrieved from <http://mathworld.wolfram.com/LeastSquaresFitting.html> *Least squares fitting*. Champaign, Illinois: MathWorld—Wolfram Research, Inc.

CHAPTER 3

Data Exploration

The word "data" is derived from the Latin word *dare*, which means "something given"—an observation or a fact about a subject. (Interestingly, the Sanskrit word *dAta* also means "given"). Data science helps decipher the hidden useful relationships within data. Before venturing into any advanced analysis of data using statistical, machine learning, and algorithmic techniques, it is essential to perform basic data exploration to study the basic characteristics of a dataset. Data exploration helps with understanding data better, to prepare the data in a way that makes advanced analysis possible, and sometimes to get the necessary insights from the data faster than using advanced analytical techniques.

Simple pivot table functions, computing statistics like mean and deviation, and plotting data as a line, bar, and scatter charts are part of data exploration techniques that are used in everyday business settings. Data exploration, also known as exploratory data analysis, provides a set of tools to obtain fundamental understanding of a dataset. The results of data exploration can be extremely powerful in grasping the structure of the data, the distribution of the values, and the presence of extreme values and the interrelationships between the attributes in the dataset. Data exploration also provides guidance on applying the right kind of further statistical and data science treatment.

Data exploration can be broadly classified into two types—descriptive statistics and data visualization. Descriptive statistics is the process of condensing key characteristics of the dataset into simple numeric metrics. Some of the common quantitative metrics used are mean, standard deviation, and correlation. Visualization is the process of projecting the data, or parts of it, into multi-dimensional space or abstract images. All the useful (and adorable) charts fall under this category. Data exploration in the context of data science uses both descriptive statistics and visualization techniques.

Data Science. DOI: https://doi.org/10.1016/B978-0-12-814761-0.00003-4

3.1 OBJECTIVES OF DATA EXPLORATION

In the data science process, data exploration is leveraged in many different steps including preprocessing or data preparation, modeling, and interpretation of the modeling results.

1. *Data understanding*: Data exploration provides a high-level overview of each attribute (also called variable) in the dataset and the interaction between the attributes. Data exploration helps answers the questions like what is the typical value of an attribute or how much do the data points differ from the typical value, or presence of extreme values.
2. *Data preparation*: Before applying the data science algorithm, the dataset has to be prepared for handling any of the anomalies that may be present in the data. These anomalies include outliers, missing values, or highly correlated attributes. Some data science algorithms do not work well when input attributes are correlated with each other. Thus, correlated attributes need to be identified and removed.
3. *Data science tasks*: Basic data exploration can sometimes substitute the entire data science process. For example, scatterplots can identify clusters in low-dimensional data or can help develop regression or classification models with simple visual rules.
4. *Interpreting the results*: Finally, data exploration is used in understanding the prediction, classification, and clustering of the results of the data science process. Histograms help to comprehend the distribution of the attribute and can also be useful for visualizing numeric prediction, error rate estimation, etc.

3.2 DATASETS

Throughout the rest of this chapter (and the book) a few classic datasets, which are simple to understand, easy to explain, and can be used commonly across many different data science techniques, will be introduced. The most popular datasets used to learn data science is probably the *Iris dataset*, introduced by Ronald Fisher, in his seminal work on discriminant analysis, "The use of multiple measurements in taxonomic problems" (Fisher, 1936). Iris is a flowering plant that is found widely, across the world. The genus of Iris contains more than 300 different species. Each species exhibits different physical characteristics like shape and size of the flowers and leaves. The *Iris dataset* contains 150 observations of three different species, *Iris setosa*, *Iris virginica*, and *I. versicolor*, with 50 observations each. Each observation consists of four attributes: sepal length, sepal width, petal length, and petal width. The fifth attribute, the label, is the name of the species observed, which takes the values *I. setosa*, *I. virginica*, and *I. versicolor*. The petals are the brightly

colored inner part of the flowers and the sepals form the outer part of the flower and are usually green in color. However, in an Iris flower, both sepals and petals are bright purple in color, but can be distinguished from each other by differences in the shape (Fig. 3.1).

All four attributes in the Iris dataset are numeric continuous values measured in centimeters. One of the species, *I. setosa*, can be easily distinguished from the other two using simple rules like the petal length is less than 2.5 cm. Separating the *virginica* and *versicolor* classes requires more complex rules that involve more attributes. The dataset is available in all standard data science tools, such as RapidMiner, or can be downloaded from public websites such as the University of California Irvine—Machine Learning repository[2] (Bache & Lichman, 2013). This dataset and other datasets used in this book can be accessed from the book companion website: www.IntroDataScience.com.

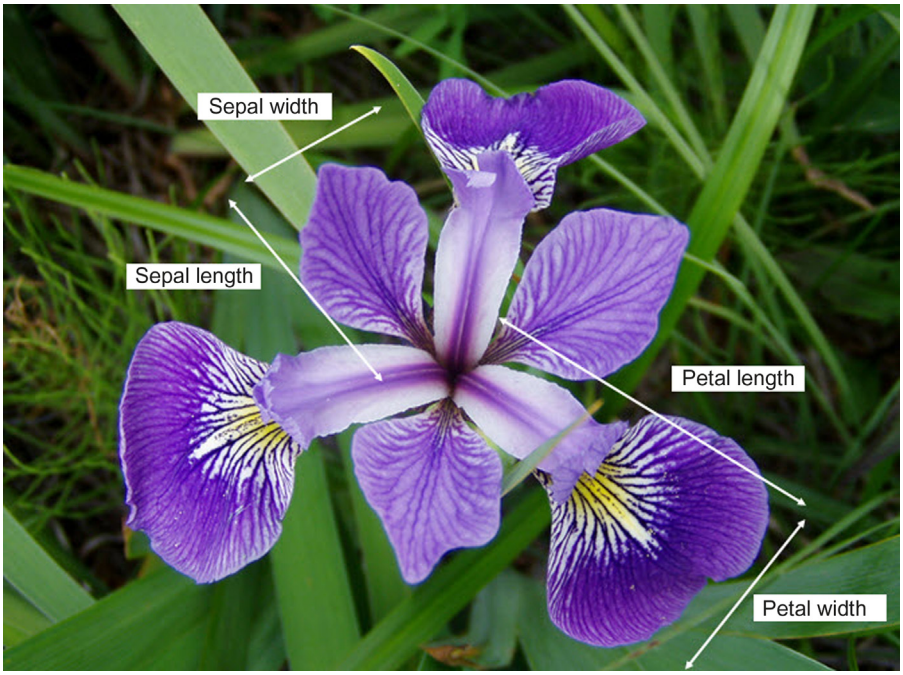

FIGURE 3.1

Iris versicolor. Source: *Photo by Danielle Langlois. July 2005 (Image modified from original by marking parts. "*Iris versicolor *3." Licensed under Creative Commons Attribution-Share Alike 3.0 via Wikimedia Commons.*[1]*)*

[1] http://commons.wikimedia.org/wiki/File:Iris_versicolor_3.jpg#mediaviewer/File:Iris_versicolor_3.jpg.
[2] https://archive.ics.uci.edu/ml/datasets.html.

The Iris dataset is used for learning data science mainly because it is simple to understand, explore, and can be used to illustrate how different data science algorithms approach the problem on the same standard dataset. The dataset extends beyond two dimensions, with three class labels, of which one class is easily separable (*I. setosa*) just by visual exploration, while classifying the other two classes is slightly more challenging. It helps to reaffirm the classification results that can be derived based on visual rules, and at the same time sets the stage for data science to build new rules beyond the limits of visual exploration.

3.2.1 Types of Data

Data come in different formats and types. Understanding the properties of each attribute or feature provides information about what kind of operations can be performed on that attribute. For example, the temperature in weather data can be expressed as any of the following formats:

- Numeric centigrade (31°C, 33.3°C) or Fahrenheit (100°F, 101.45°F) or on the Kelvin scale
- Ordered labels as in hot, mild, or cold
- Number of days within a year below 0°C (10 days in a year below freezing)

All of these attributes indicate temperature in a region, but each have different data types. A few of these data types can be converted from one to another.

Numeric or Continuous

Temperature expressed in Centigrade or Fahrenheit is numeric and continuous because it can be denoted by numbers and take an infinite number of values between digits. Values are ordered and calculating the difference between the values makes sense. Hence, additive and subtractive mathematical operations and logical comparison operators like greater than, less than, and equal to, operations can be applied.

An integer is a special form of the numeric data type which does not have decimals in the value or more precisely does not have infinite values between consecutive numbers. Usually, they denote a count of something, number of days with temperature less than 0°C, number of orders, number of children in a family, etc.

If a zero point is defined, numeric data become a *ratio* or *real* data type. Examples include temperature in Kelvin scale, bank account balance, and income. Along with additive and logical operations, ratio operations can be performed with this data type. Both integer and ratio data types are categorized as a *numeric* data type in most data science tools.

Categorical or Nominal

Categorical data types are attributes treated as distinct symbols or just names. The color of the iris of the human eye is a categorical data type because it takes a value like black, green, blue, gray, etc. There is no direct relationship among the data values, and hence, mathematical operators except the logical or "is equal" operator cannot be applied. They are also called a nominal or polynominal data type, derived from the Latin word for *name*.

An ordered nominal data type is a special case of a categorical data type where there is some kind of order among the values. An example of an ordered data type is temperature expressed as hot, mild, cold.

Not all data science tasks can be performed on all data types. For example, the neural network algorithm does not work with categorical data. However, one data type can be converted to another using a type conversion process, but this may be accompanied with possible loss of information. For example, credit scores expressed in poor, average, good, and excellent categories can be converted to either 1, 2, 3, and 4 or average underlying numerical scores like 400, 500, 600, and 700 (scores here are just an example). In this type conversion, there is no loss of information. However, conversion from numeric credit score to categories (poor, average, good, and excellent) does incur loss of information.

3.3 DESCRIPTIVE STATISTICS

Descriptive statistics refers to the study of the aggregate quantities of a dataset. These measures are some of the commonly used notations in everyday life. Some examples of descriptive statistics include average annual income, median home price in a neighborhood, range of credit scores of a population, etc. In general, descriptive analysis covers the following characteristics of the sample or population dataset (Kubiak & Benbow, 2006):

Characteristics of the Dataset	Measurement Technique
Center of the dataset	Mean, median, and mode
Spread of the dataset	Range, variance, and standard deviation
Shape of the distribution of the dataset	Symmetry, skewness, and kurtosis

The definition of these metrics will be explored shortly. Descriptive statistics can be broadly classified into univariate and multivariate exploration depending on the number of attributes under analysis.

3.3.1 Univariate Exploration

Univariate data exploration denotes analysis of one attribute at a time. The example Iris dataset for one species, *I. setosa*, has 50 observations and 4 attributes, as shown in Table 3.1. Here some of the descriptive statistics for sepal length attribute are explored.

Measure of Central Tendency

The objective of finding the central location of an attribute is to quantify the dataset with one central or most common number.

- *Mean*: The mean is the arithmetic average of all observations in the dataset. It is calculated by summing all the data points and dividing by the number of data points. The mean for sepal length in centimeters is 5.0060.
- *Median*: The median is the value of the central point in the distribution. The median is calculated by sorting all the observations from small to large and selecting the mid-point observation in the sorted list. If the number of data points is even, then the average of the middle two data points is used as the median. The median for sepal length is in centimeters is 5.0000.
- *Mode*: The mode is the most frequently occurring observation. In the dataset, data points may be repetitive, and the most repetitive data point is the mode of the dataset. In this example, the mode in centimeters is 5.1000.

Table 3.1 Iris Dataset and Descriptive Statistics (Fisher, 1936)

Observation	Sepal Length	Sepal Width	Petal Length	Petal Width
1	5.1	3.5	1.4	0.2
2	4.9	3.1	1.5	0.1
...	...	...	...	...
49	5	3.4	1.5	0.2
50	4.4	2.9	1.4	0.2
Statistics	**Sepal Length**	**Sepal Width**	**Petal Length**	**Petal Width**
Mean	5.006	3.418	1.464	0.244
Median	5.000	3.400	1.500	0.200
Mode	5.100	3.400	1.500	0.200
Range	1.500	2.100	0.900	0.500
Standard deviation	0.352	0.381	0.174	0.107
Variance	0.124	0.145	0.030	0.011

In an attribute, the mean, median, and mode may be different numbers, and this indicates the shape of the distribution. If the dataset has outliers, the mean will get affected while in most cases the median will not. The mode of the distribution can be different from the mean or median, if the underlying dataset has more than one natural normal distribution.

Measure of Spread

In desert regions, it is common for the temperature to cross above 110°F during the day and drop below 30°F during the night while the average temperature for a 24-hour period is around 70°F. Obviously, the experience of living in the desert is not the same as living in a tropical region with the same average daily temperature around 70°F, where the temperature within the day is between 60°F and 80°F. What matters here is not just the central location of the temperature, but the *spread* of the temperature. There are two common metrics to quantify spread.

- *Range:* The range is the difference between the maximum value and the minimum value of the attribute. The range is simple to calculate and articulate but has shortcomings as it is severely impacted by the presence of outliers and fails to consider the distribution of all other data points in the attributes. In the example, the range for the temperature in the desert is 80°F and the range for the tropics is 20°F. The desert region experiences larger temperature swings as indicated by the range.
- *Deviation:* The variance and standard deviation measures the spread, by considering all the values of the attribute. Deviation is simply measured as the difference between any given value (x_i) and the mean of the sample (μ). The variance is the sum of the squared deviations of all data points divided by the number of data points. For a dataset with N observations, the variance is given by the following equation:

$$\text{Variance} = s^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 \tag{3.1}$$

Standard deviation is the square root of the variance. Since the standard deviation is measured in the same units as the attribute, it is easy to understand the magnitude of the metric. High standard deviation means the data points are spread widely around the central point. Low standard deviation means data points are closer to the central point. If the distribution of the data aligns with the *normal distribution*, then 68% of the data points lie within one standard deviation from the mean. Fig. 3.2 provides the univariate summary of the Iris dataset with all 150 observations, for each of the four numeric attributes.

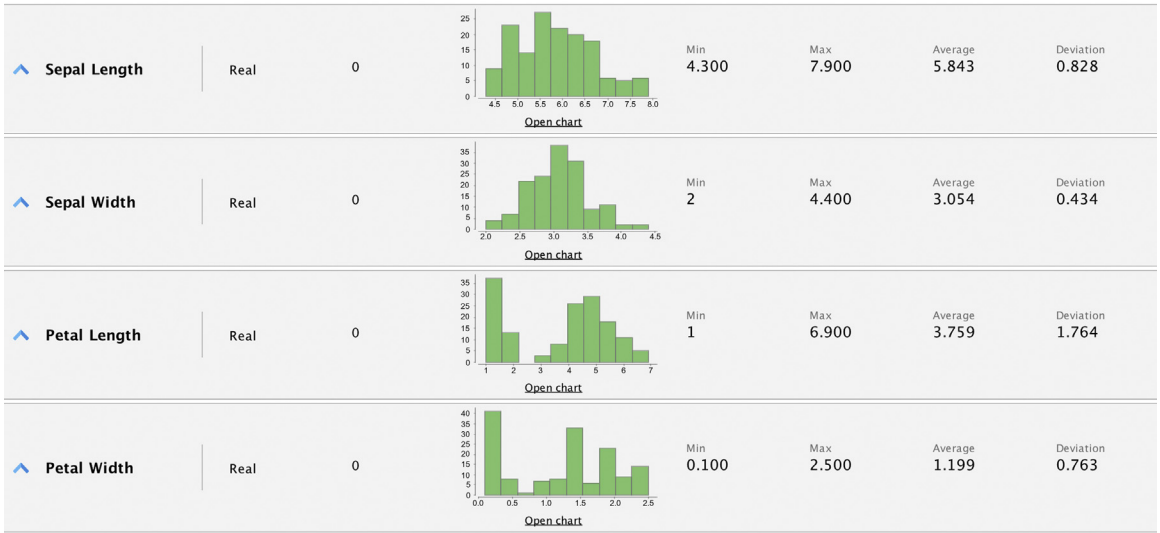

FIGURE 3.2
Descriptive statistics for the Iris dataset.

3.3.2 Multivariate Exploration

Multivariate exploration is the study of more than one attribute in the dataset simultaneously. This technique is critical to understanding the relationship between the attributes, which is central to data science methods. Similar to univariate explorations, the measure of central tendency and variance in the data will be discussed.

Central Data Point

In the Iris dataset, each data point as a set of all the four attributes can be expressed:

observation *i*: {sepal length, sepal width, petal length, petal width}

For example, observation one: {5.1, 3.5, 1.4, 0.2}. This observation point can also be expressed in four-dimensional Cartesian coordinates and can be plotted in a graph (although plotting more than three dimensions in a visual graph can be challenging). In this way, all 150 observations can be expressed in Cartesian coordinates. If the objective is to find the most "typical" observation point, it would be a data point made up of the mean of each attribute in the dataset independently. For the Iris dataset shown in Table 3.1, the central mean point is {5.006, 3.418, 1.464, 0.244}. This data point may not be an actual observation. It will be a hypothetical data point with the most typical attribute values.

Correlation

Correlation measures the statistical relationship between two attributes, particularly dependence of one attribute on another attribute. When two attributes are highly correlated with each other, they both vary at the same rate with each other either in the same or in opposite directions. For example, consider average temperature of the day and ice cream sales. Statistically, the two attributes that are correlated are dependent on each other and one may be used to predict the other. If there are sufficient data, future sales of ice cream can be predicted if the temperature forecast is known. However, correlation between two attributes does not imply causation, that is, one doesn't necessarily cause the other. The ice cream sales and the shark attacks are correlated, however there is no causation. Both ice cream sales and shark attacks are influenced by the third attribute—the summer season. Generally, ice cream sales spikes as temperatures rise. As more people go to beaches during summer, encounters with sharks become more probable.

Correlation between two attributes is commonly measured by the Pearson correlation coefficient (r), which measures the strength of *linear* dependence (Fig. 3.3). Correlation coefficients take a value from $-1 \leq r \leq 1$. A value closer to 1 or -1 indicates the two attributes are highly correlated, with perfect correlation at 1 or -1. Perfect correlation also exists when the attributes are governed by formulas and laws. For example, observations of the values of gravitational force and the mass of the object (Newton's second law) or the quantity of the products sold and total revenue (price* volume = revenue). A correlation value of 0 means there is no linear relationship between two attributes.

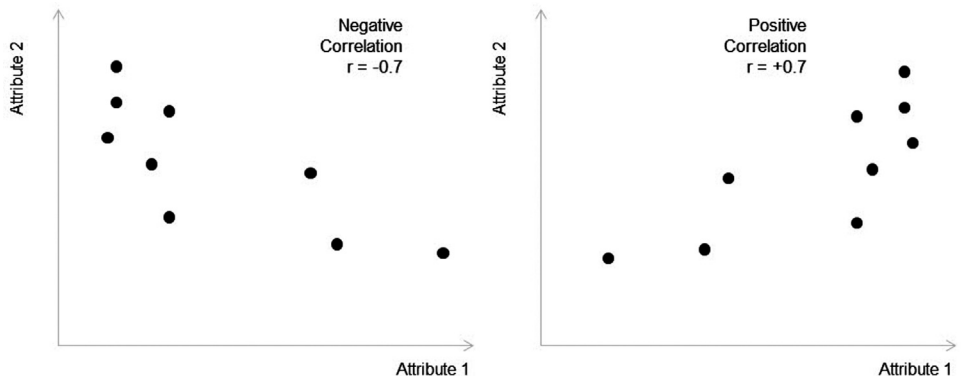

FIGURE 3.3
Correlation of attributes.

The Pearson correlation coefficient between two attributes x and y is calculated with the formula:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}} = \frac{\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{N\times s_x\times s_y} \tag{3.2}$$

where s_x and s_y are the standard deviations of random variables x and y, respectively. The Pearson correlation coefficient has some limitations in quantifying the strength of correlation. When datasets have more complex nonlinear relationships like quadratic functions, only the effects on linear relationships are considered and quantified using correlation coefficient. The presence of outliers can also skew the measure of correlation. Visually, correlation can be observed using scatterplots with the attributes in each Cartesian coordinate (Fig. 3.3). In fact, visualization should be the first step in understanding correlation because it can identify nonlinear relationships and show any outliers in the dataset. Anscombe's quartet (Anscombe, 1973) clearly illustrates the limitations of relying only on the correlation coefficient to understand the data (Fig. 3.4). The quartet consists of four different datasets, with two attributes (x, y). All four datasets have the same mean, the same variance for x and y, and the same correlation coefficient between x and y, but look drastically different when plotted on a chart. This evidence illustrates the necessity of visualizing the attributes instead of just calculating statistical metrics.

3.4 DATA VISUALIZATION

Visualizing data is one of the most important techniques of data discovery and exploration. Though visualization is not considered a data science technique, terms like visual mining or pattern discovery based on visuals are increasingly used in the context of data science, particularly in the business world. The discipline of data visualization encompasses the methods of expressing data in an abstract visual form. The visual representation of data provides easy comprehension of complex data with multiple attributes and their underlying relationships. The motivation for using data visualization includes:

- *Comprehension of dense information*: A simple visual chart can easily include thousands of data points. By using visuals, the user can see the big picture, as well as longer term trends that are extremely difficult to interpret purely by expressing data in numbers.

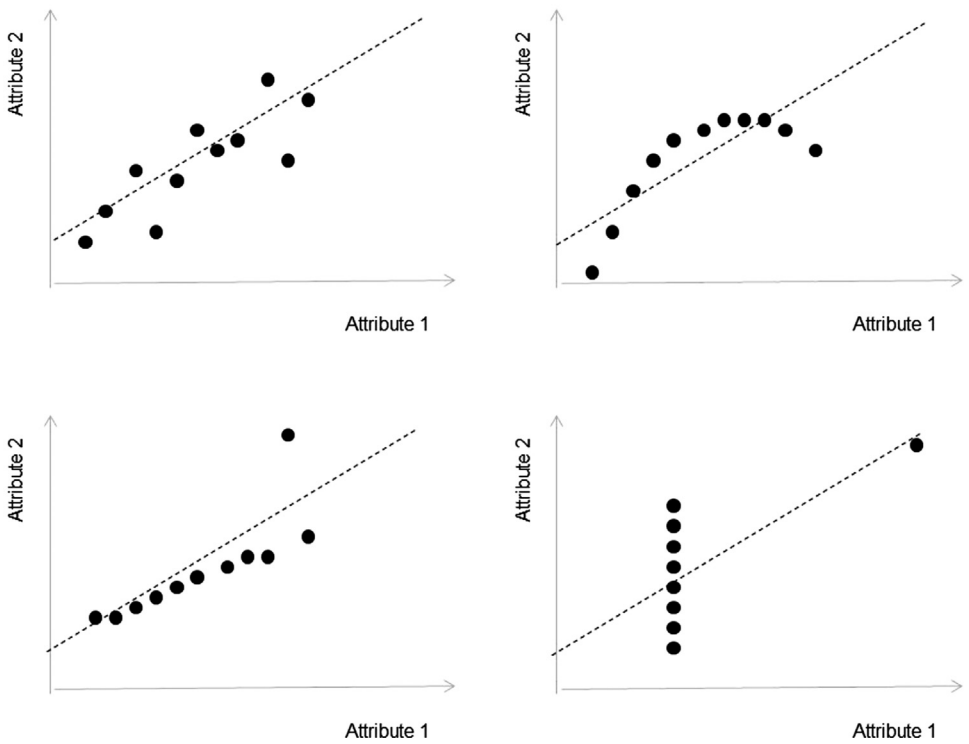

FIGURE 3.4

Anscombe's Quartet: descriptive statistics versus visualization. Source*: Adapted from: Anscombe, F. J., 1973. Graphs in statistical analysis*, American Statistician, 27*(1), pp. 19–20.*

- *Relationships*: Visualizing data in Cartesian coordinates enables exploration of the relationships between the attributes. Although representing more than three attributes on the *x*, *y*, and *z*-axes is not feasible in Cartesian coordinates, there are a few creative solutions available by changing properties like the size, color, and shape of data markers or using flow maps (Tufte, 2001), where more than two attributes are used in a two-dimensional medium.

Vision is one of the most powerful senses in the human body. As such, it is intimately connected with cognitive thinking (Few, 2006). Human vision is trained to discover patterns and anomalies even in the presence of a large volume of data. However, the effectiveness of the pattern detection depends on how effectively the information is visually presented. Hence, selecting suitable visuals to explore data is critically important in discovering and comprehending hidden patterns in the data (Ware, 2004). As with descriptive statistics, visualization techniques are also categorized into: univariate visualization, multivariate visualization and visualization of a large number of attributes using parallel dimensions.

Some of the common data visualization techniques used to analyze data will be reviewed. Most of these visualization techniques are available in commercial spreadsheet software like MS Excel. RapidMiner, like any other data science tool, offers a wide range of visualization tools. To maintain consistency with rest of the book, all further visualizations are output from RapidMiner using the Iris dataset. Please review Chapter 15, Getting Started With RapidMiner, to become familiar with RapidMiner.

3.4.1 Univariate Visualization

Visual exploration starts with investigating one attribute at a time using univariate charts. The techniques discussed in this section give an idea of how the attribute values are distributed and the shape of the distribution.

Histogram

A histogram is one of the most basic visualization techniques to understand the frequency of the occurrence of values. It shows the distribution of the data by plotting the frequency of occurrence in a range. In a histogram, the attribute under inquiry is shown on the horizontal axis and the frequency of occurrence is on the vertical axis. For a continuous numeric data type, the range or *binning* value to group a range of values need to be specified. For example, in the case of human height in centimeters, all the occurrences between 152.00 and 152.99 are grouped under 152. There is no optimal number of bins or bin width that works for all the distributions. If the bin width is too small, the distribution becomes more precise but reveals the noise due to sampling. A general rule of thumb is to have a number of bins equal to the square root or cube root of the number of data points.

Histograms are used to find the central location, range, and shape of distribution. In the case of the petal length attribute in the Iris dataset, the data is multimodal (Fig. 3.5), where the distribution does not follow the bell curve pattern. Instead, there are two peaks in the distribution. This is due to the fact that there are 150 observations of three *different* species (hence, distributions) in the dataset. A histogram can be *stratified* to include different classes in order to gain more insight. The enhanced histogram with class labels shows the dataset is made of three different distributions (Fig. 3.6). *I. setosa's* distribution stands out with a mean around 1.25 cm and ranges from 1−2 cm. *I. versicolor* and *I. virginica's* distributions overlap *I. setosa's* have separate means.

Quartile

A *box whisker* plot is a simple visual way of showing the distribution of a continuous variable with information such as quartiles, median, and outliers,

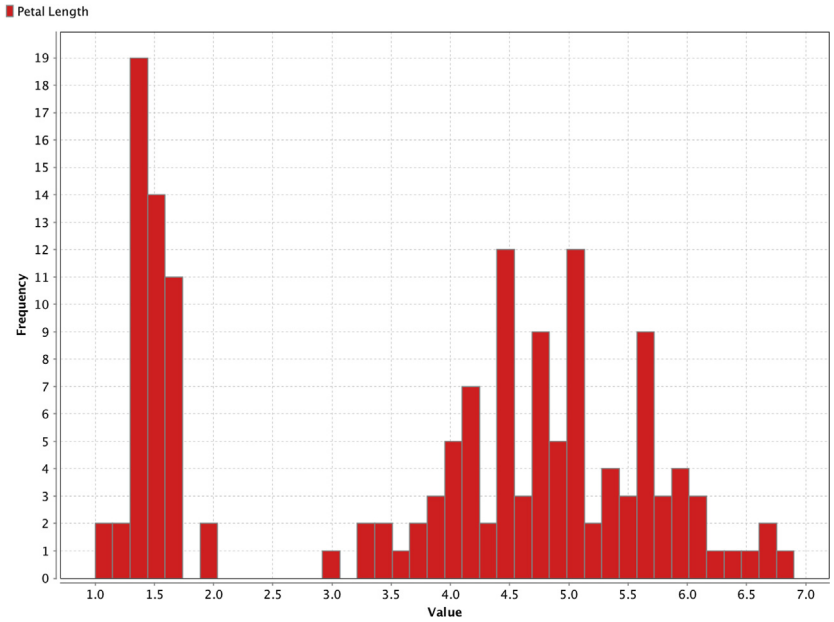

FIGURE 3.5
Histogram of petal length in Iris dataset.

overlaid by mean and standard deviation. The main attraction of box whisker or quartile charts is that distributions of multiple attributes can be compared side by side and the overlap between them can be deduced. The quartiles are denoted by Q1, Q2, and Q3 points, which indicate the data points with a 25% bin size. In a distribution, 25% of the data points will be below Q1, 50% will be below Q2, and 75% will be below Q3.

The Q1 and Q3 points in a box whisker plot are denoted by the edges of the box. The Q2 point, the median of the distribution, is indicated by a cross line within the box. The outliers are denoted by circles at the end of the whisker line. In some cases, the mean point is denoted by a solid dot overlay followed by standard deviation as a line overlay.

Fig. 3.7 shows that the quartile charts for all four attributes of the Iris dataset are plotted side by side. Petal length can be observed as having the broadest range and the sepal width has a narrow range, out of all of the four attributes.

One attribute can also be selected—petal length—and explored further using quartile charts by introducing a class label. In the plot in Fig. 3.8, we can see the distribution of three species for the petal length measurement. Similar to the previous comparison, the distribution of multiple species can be compared.

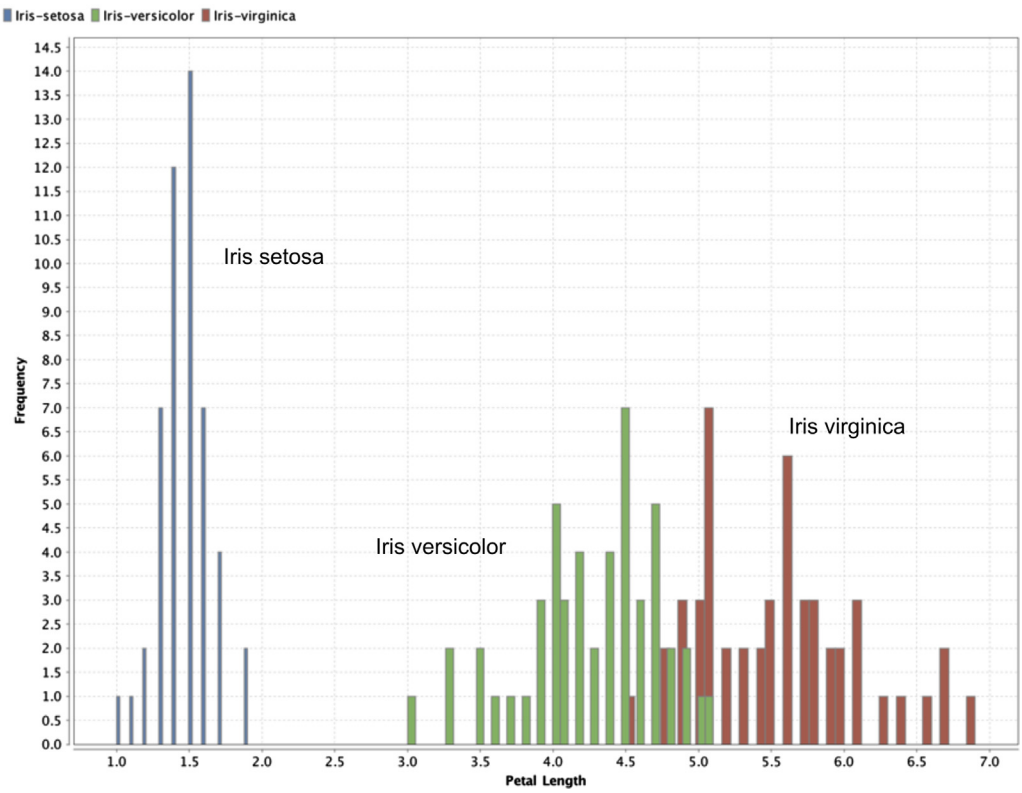

FIGURE 3.6
Class-stratified histogram of petal length in Iris dataset.

Distribution Chart

For continuous numeric attributes like petal length, instead of visualizing the actual data in the sample, its normal distribution function can be visualized instead. The normal distribution function of a continuous random variable is given by the formula:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{(x-\mu)^2/2\sigma^2} \tag{3.3}$$

where μ is the mean of the distribution and σ is the standard deviation of the distribution. Here an inherent assumption is being made that the measurements of petal length (or any continuous variable) follow the normal distribution, and hence, its distribution can be visualized instead of the actual values. The normal distribution is also called the *Gaussian distribution* or "bell curve" due to its bell shape. The normal distribution function

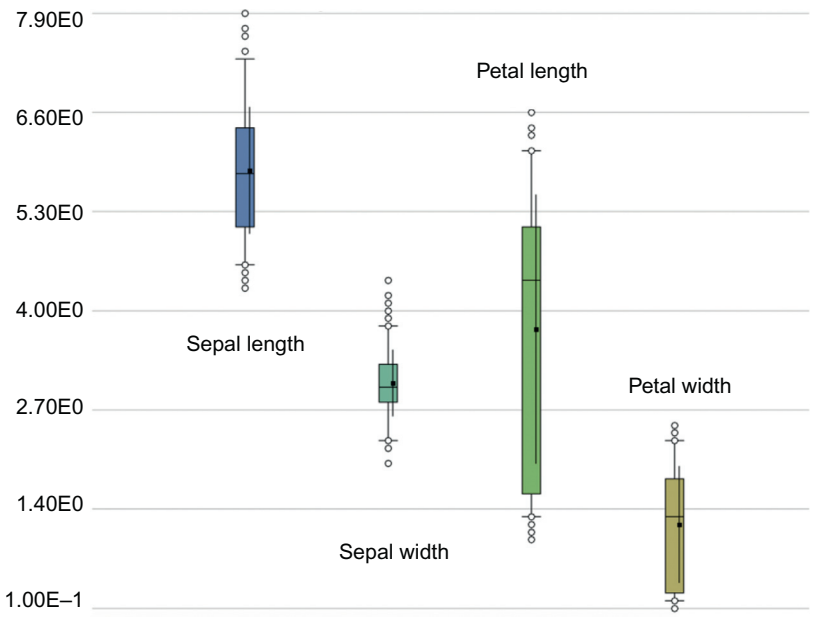

FIGURE 3.7
Quartile plot of Iris dataset.

shows the probability of occurrence of a data point within a range of values. If a dataset exhibits normal distribution, then 68.2% of data points will fall within one standard deviation from the mean; 95.4% of the points will fall within 2σ and 99.7% within 3σ of the mean. When the normal distribution curves are stratified by class type, more insight into the data can be gained. Fig. 3.9 shows the normal distribution curves for petal length measurement for each Iris species type. From the distribution chart, it can be inferred that the petal length for the *I. setosa* sample is more distinct and cohesive than *I. versicolor* and *I. virginica*. If there is an unlabeled measurement with a petal length of 1.5 cm, it can be predicted that the species is *I. setosa*. However, if the petal length measurement is 5.0 cm, there is no clear prediction, as the species could be either *Iris versicolor* and *I. virginica*.

3.4.2 Multivariate Visualization

The multivariate visual exploration considers more than one attribute in the same visual. The techniques discussed in this section focus on the relationship of one attribute with another attribute. These visualizations examine two to four attributes simultaneously.

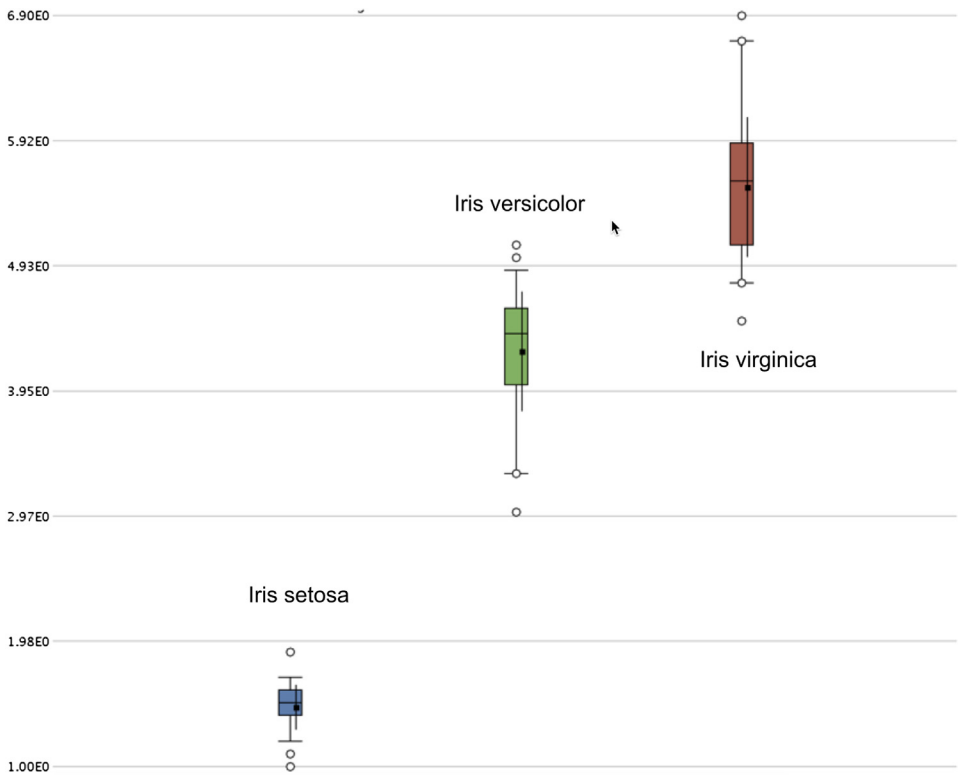

FIGURE 3.8
Class-stratified quartile plot of petal length in Iris dataset.

Scatterplot

A scatterplot is one of the most powerful yet simple visual plots available. In a scatterplot, the data points are marked in Cartesian space with attributes of the dataset aligned with the coordinates. The attributes are usually of continuous data type. One of the key observations that can be concluded from a scatterplot is the existence of a relationship between two attributes under inquiry. If the attributes are linearly correlated, then the data points align closer to an imaginary straight line; if they are not correlated, the data points are scattered. Apart from basic correlation, scatterplots can also indicate the existence of patterns or groups of clusters in the data and identify outliers in the data. This is particularly useful for low-dimensional datasets. Chapter 13: Anomaly detection, provides techniques for finding outliers in high-dimensional space.

Fig. 3.10 shows the scatterplot between petal length (x-axis) and petal width (y-axis). These two attributes are slightly correlated, because this is a measurement of the same part of the flower. When the data markers are colored to

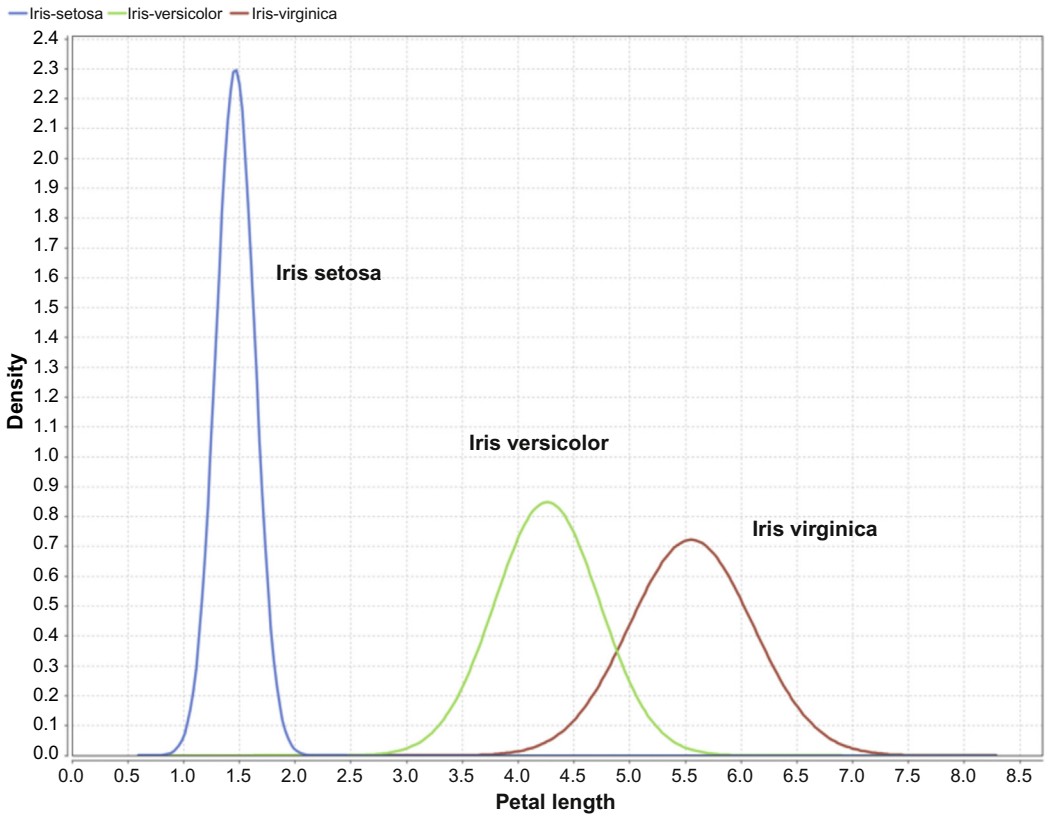

FIGURE 3.9
Distribution of petal length in Iris dataset.

indicate different species using class labels, more patterns can be observed. There is a cluster of data points, all belonging to species *I. setosa*, on the lower left side of the plot. *I. setosa* has much smaller petals. This feature can be used as a rule to predict the species of unlabeled observations. One of the limitations of scatterplots is that only two attributes can be used at a time, with an additional attribute possibly shown in the color of the data marker. However, the colors are usually reserved for class labels.

Scatter Multiple

A *scatter multiple* is an enhanced form of a simple scatterplot where more than two dimensions can be included in the chart and studied simultaneously. The primary attribute is used for the *x*-axis coordinate. The secondary axis is shared with more attributes or dimensions. In this example (Fig. 3.11), the values on the *y*-axis are shared between sepal length, sepal

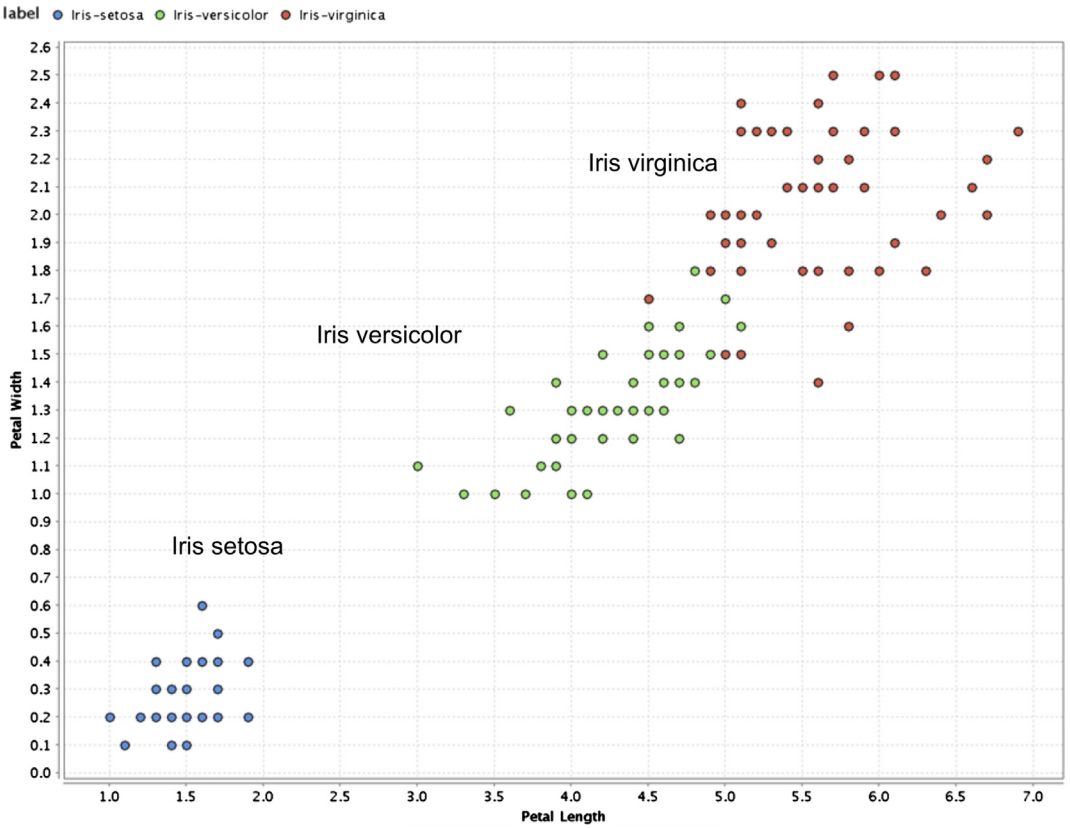

FIGURE 3.10
Scatterplot of Iris dataset.

width, and petal width. The name of the attribute is conveyed by colors used in data markers. Here, sepal length is represented by data points occupying the topmost part of the chart, sepal width occupies the middle portion, and petal width is in the bottom portion. Note that the data points are *duplicated for each attribute in the y-axis*. Data points are color-coded for each dimension in *y*-axis while the *x*-axis is anchored with one attribute—petal length. All the attributes sharing the y-axis should be of the same unit or normalized.

Scatter Matrix

If the dataset has more than two attributes, it is important to look at combinations of all the attributes through a scatterplot. A *scatter matrix* solves this need by comparing all combinations of attributes with individual scatterplots and arranging these plots in a matrix.

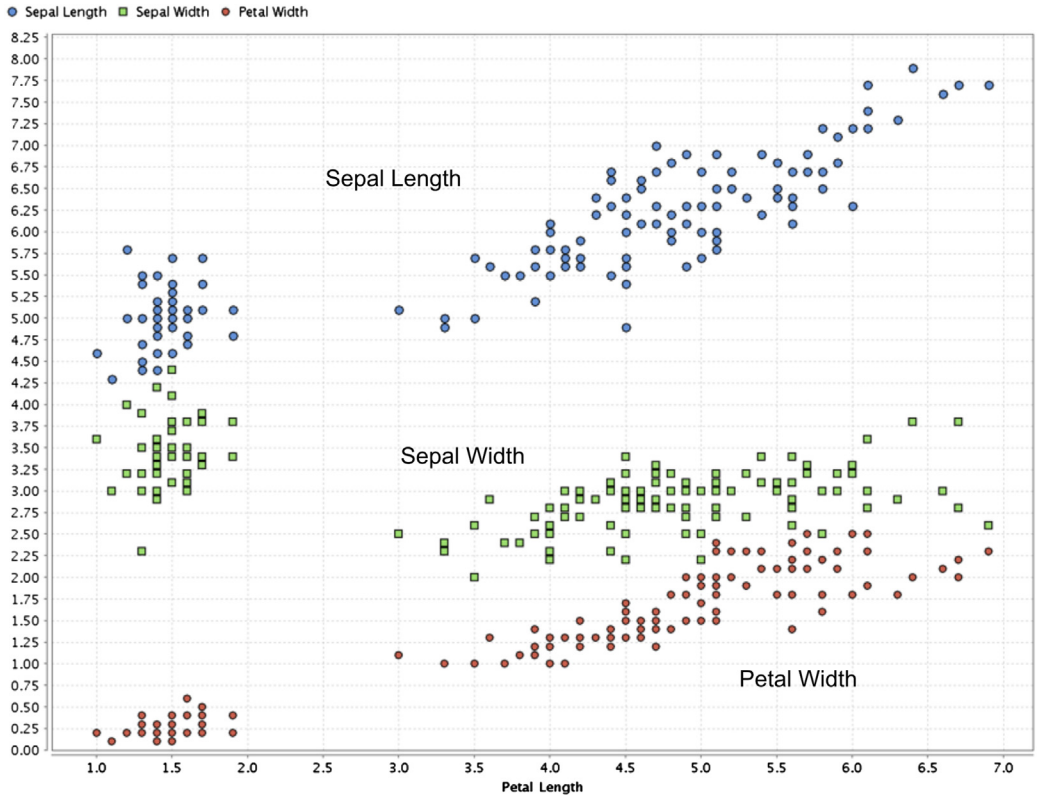

FIGURE 3.11
Scatter multiple plot of Iris dataset.

A scatter matrix for all four attributes in the Iris dataset is shown in Fig. 3.12. The color of the data point is used to indicate the species of the flower. Since there are four attributes, there are four rows and four columns, for a total of 16 scatter charts. Charts in the diagonal are a comparison of the attribute with itself; hence, they are eliminated. Also, the charts below the diagonal are mirror images of the charts above the diagonal. In effect, there are six distinct comparisons in scatter multiples of four attributes. Scatter matrices provide an effective visualization of comparative, multivariate, and high-density data displayed in small multiples of the similar scatterplots (Tufte, 2001).

Bubble Chart

A *bubble chart* is a variation of a simple scatterplot with the addition of one more attribute, which is used to determine the size of the data point. In the Iris dataset, petal length and petal width are used for x and y-axis, respectively

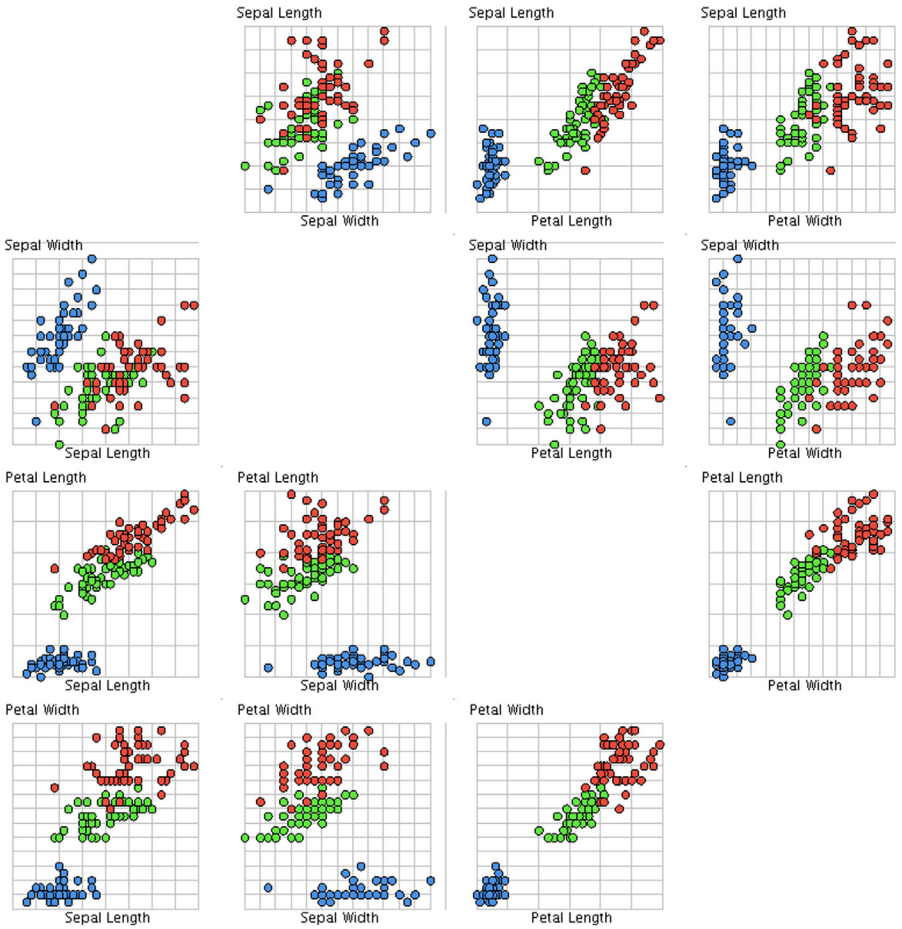

FIGURE 3.12
Scatter matrix plot of Iris dataset.

and sepal width is used for the size of the data point. The color of the data point represents a species class label (Fig. 3.13).

Density Chart

Density charts are similar to the scatterplots, with one more dimension included as a background color. The data point can also be colored to visualize one dimension, and hence, a total of four dimensions can be visualized in a density chart. In the example in Fig. 3.14, petal length is used for the *x*-axis, sepal length for the *y*-axis, sepal width for the background color, and class label for the data point color.

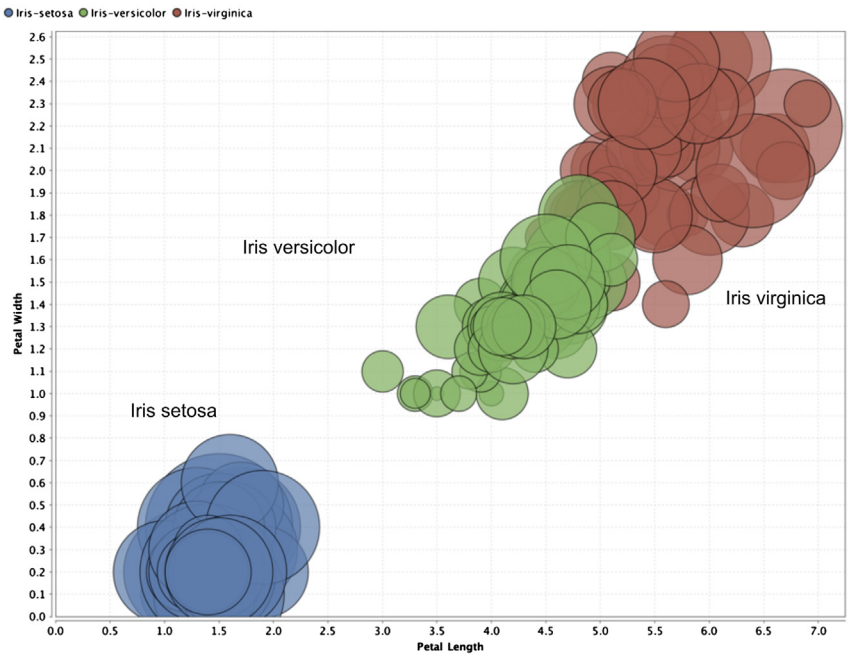

FIGURE 3.13
Bubble chart of Iris dataset.

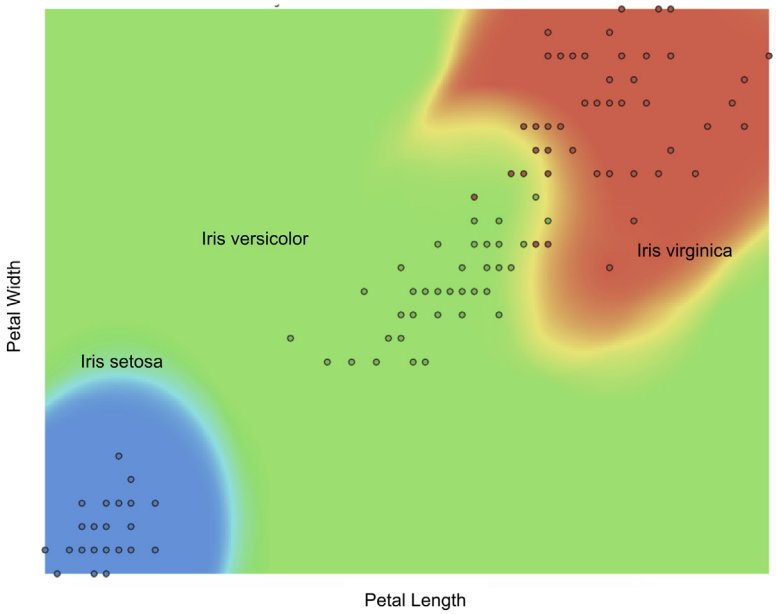

FIGURE 3.14
Density chart of a few attributes in the Iris dataset.

3.4.3 Visualizing High-Dimensional Data

Visualizing more than three attributes on a two-dimensional medium (like a paper or screen) is challenging. This limitation can be overcome by using transformation techniques to project the high-dimensional data points into parallel axis space. In this approach, a Cartesian axis is shared by more than one attribute.

Parallel Chart

A *parallel chart* visualizes a data point quite innovatively by transforming or projecting multi-dimensional data into a two-dimensional chart medium. In this chart, every attribute or dimension is linearly arranged in one coordinate (x-axis) and all the measures are arranged in the other coordinate (y-axis). Since the x-axis is multivariate, each data point is represented as a *line* in a parallel space.

In the case of the Iris dataset, all four attributes are arranged along the x-axis. The y-axis represents a generic distance and it is "shared" by all these attributes on the x-axis. Hence, parallel charts work only when attributes share a common unit of numerical measure or when the attributes are normalized. This visualization is called a *parallel axis* because all four attributes are represented in four parallel axes parallel to the y-axis.

In a parallel chart, a class label is used to color each data *line* so that one more dimension is introduced into the picture. By observing this parallel chart in Fig. 3.15, it can be noted that there is overlap between the three species on the sepal width attribute. So, sepal width cannot be the metric used to differentiate these three species. However, there is clear separation of species in petal length. No observation of *I. setosa* species has a petal length above 2.5 cm and there is little overlap between the *I. virginica* and *I. versicolor* species. Visually, just by knowing the petal length of an unlabeled observation, the species of Iris flower can be predicted. The relevance of this rule as a predictor will be discussed in the later chapter on Classification.

Deviation Chart

A *deviation chart* is very similar to a *parallel chart*, as it has parallel axes for all the attributes on the x-axis. Data points are extended across the dimensions as lines and there is one common y-axis. Instead of plotting all data lines, deviation charts only show the mean and standard deviation statistics. For each class, deviation charts show the mean line connecting the mean of each attribute; the standard deviation is shown as the band above and below the mean line. The mean line does not have to correspond to a data point (line). With this method, information is elegantly displayed, and the essence of a parallel chart is maintained.

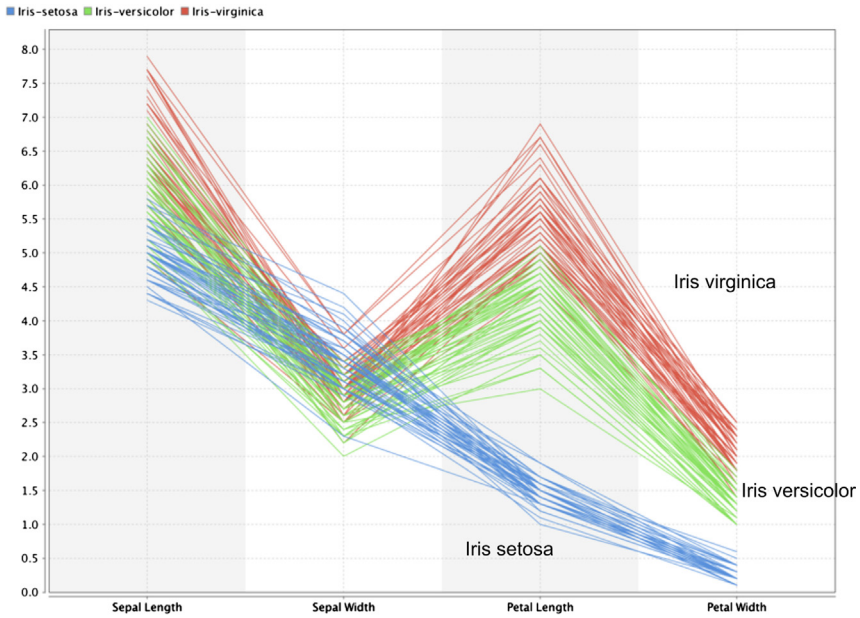

FIGURE 3.15
Parallel chart of Iris dataset.

In Fig. 3.16, a deviation chart for the Iris dataset stratified by species is shown. It can be observed that the petal length is a good predictor to classify the species because the mean line and the standard deviation bands for the species are well separated.

Andrews Curves

An *Andrews plot* belongs to a family of visualization techniques where the high-dimensional data are projected into a vector space so that each data point takes the form of a line or curve. In an Andrews plot, each data point X with d dimensions, $X = (x_1, x_2, x_3, \ldots, x_d)$, takes the form of a Fourier series:

$$f_x(t) = \frac{x_1}{\sqrt{2}} + x_2\sin(t) + x_3\cos(t) + x_4\sin(2t) + x_5\cos(2t) + \cdots \tag{3.4}$$

This function is plotted for $-\pi < t < \pi$ for each data point. Andrews plots are useful to determine if there are any outliers in the data and to identify potential patterns within the data points (Fig. 3.17). If two data points are similar, then the curves for the data points are closer to each other. If curves are far apart and belong to different classes, then this information can be used classify the data (Garcia-Osorio & Fyfe, 2005).

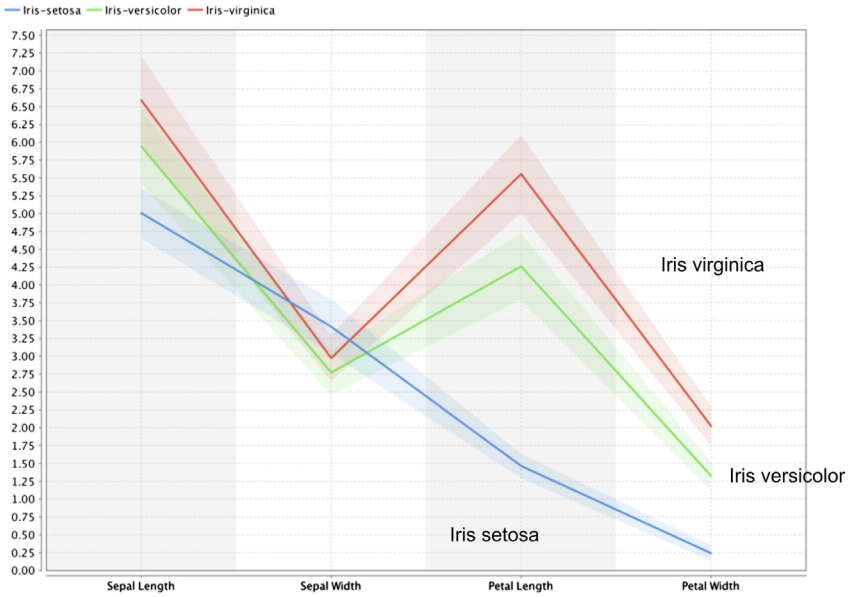

FIGURE 3.16
Deviation chart of Iris dataset.

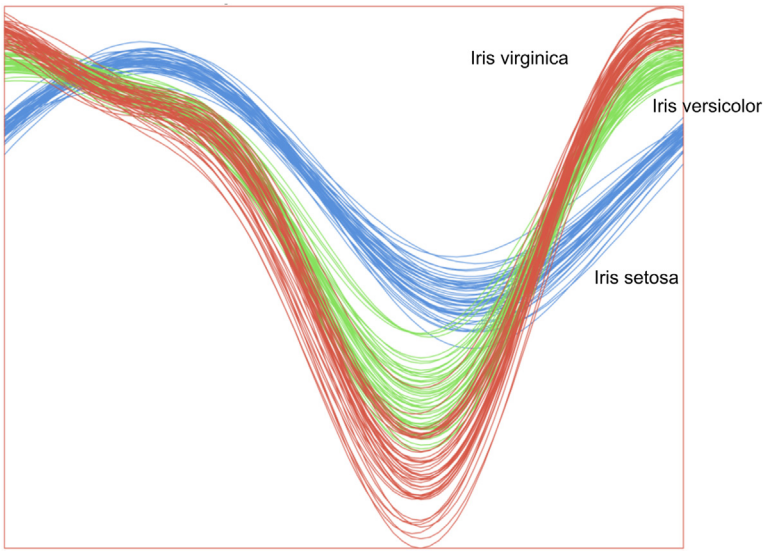

FIGURE 3.17
Andrews curves of Iris dataset.

Many of the charts and visuals discussed in this chapter explore the multivariate relationships within the dataset. They form the set of classic data visualizations used for data exploration, postprocessing, and understanding data science models. Some new developments in the area of visualization deals with networks and connections within the data objects (Lima, 2011). To better analyze data extracted from graph data, social networks, and integrated applications, connectivity charts are often used. Interactive exploration of data using visualization software provides an essential tool to observe multiple attributes at the same time but has limitations on the number of attributes used in visualizations. Hence, dimensional reduction using techniques discussed in Chapter 14, Feature Selection, can help in visualizing higher-dimensional data by reducing the dimensions to a critical few.

3.5 ROADMAP FOR DATA EXPLORATION

If there is a new dataset that has not been investigated before, having a structured way to explore and analyze the data will be helpful. Here is a roadmap to inquire a new dataset. Not all steps may be relevant for every dataset and the order may need to be adjusted for some sets, so this roadmap is intended as a guideline.

1. *Organize the dataset*: Structure the dataset with standard rows and columns. Organizing the dataset to have objects or instances in rows and dimensions or attributes in columns will be helpful for many data analysis tools. Identify the target or "class label" attribute, if applicable.
2. *Find the central point for each attribute*: Calculate *mean*, *median*, and *mode* for each attribute and the class label. If all three values are very different, it may indicate the presence of an outlier, or a multimodal or nonnormal distribution for an attribute.
3. *Understand the spread of each attribute*: Calculate the *standard deviation* and *range* for an attribute. Compare the standard deviation with the mean to understand the spread of the data, along with the max and min data points.
4. *Visualize the distribution of each attribute*: Develop the *histogram* and *distribution* plots for each attribute. Repeat the same for class-stratified histograms and distribution plots, where the plots are either repeated or color-coded for each class.
5. *Pivot the data*: Sometimes called dimensional slicing, a pivot is helpful to comprehend different values of the attributes. This technique can stratify by class and drill down to the details of any of the attributes. Microsoft Excel and Business Intelligence tools popularized this technique of data analysis for a wider audience.

6. *Watch out for outliers*: Use a scatterplot or quartiles to find outliers. The presence of outliers skews some measures like mean, variance, and range. Exclude outliers and rerun the analysis. Notice if the results change.
7. *Understand the relationship between attributes*: Measure the *correlation* between attributes and develop a correlation matrix. Notice what attributes are dependent on each other and investigate why they are dependent.
8. *Visualize the relationship between attributes*: Plot a quick scatter matrix to discover the relationship between multiple attributes at once. Zoom in on the attribute pairs with simple two-dimensional scatterplots stratified by class.
9. *Visualize high-dimensional datasets*: Create *parallel charts* and *Andrews curves* to observe the class differences exhibited by each attribute. *Deviation charts* provide a quick assessment of the spread of each class for each attribute.

References

Anscombe, F. J. (1973). Graphs in statistical analysis. *American Statistician, 27*(1), 17–21.

Bache, K., & Lichman, M. (2013) *University of California, School of Information and Computer Science*. Retrieved from UCI Machine Learning Repository <http://archive.ics.uci.edu/ml>.

Few, S. (2006). *Information dashboard design: The effective visual communication of data*. Sebastopol, CA: O'Reilly Media.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Human Genetics, 7*, 179–188, 10.1111/j.1469-1809.1936.tb02137.x.

Garcia-Osorio, C., & Fyfe, C. (2005). Visualization of high-dimensional data via orthogonal curves. *Journal of Universal Computer Science, 11*(11), 1806–1819.

Kubiak, T., & Benbow, D. W. (2006). *The certified six sigma black belt handbook*. Milwaukee, WI: ASQ Quality Press.

Lima, M. (2011). *Visual complexity: Mapping patterns of information*. New York: Princeton Architectural Press.

Tufte, E. R. (2001). *The visual display of quantitative information*. Cheshire, CT: Graphics Press.

Ware, C. (2004). *Information visualization: Perception for design*. Waltham, MA: Morgan Kaufmann.