2

WHAT ARE DATA, AND WHAT IS A DATA SET?

As its name suggests, data science is fundamentally dependent on data. In its most basic form, a datum or a piece of information is an abstraction of a real-world entity (person, object, or event). The terms *variable*, *feature*, and *attribute* are often used interchangeably to denote an individual abstraction. Each entity is typically described by a number of attributes. For example, a book might have the following attributes: author, title, topic, genre, publisher, price, date published, word count, number of chapters, number of pages, edition, ISBN, and so on.

A data set consists of the data relating to a collection of entities, with each entity described in terms of a set of attributes. In its most basic form,[1] a data set is organized in an $n * m$ data matrix called the *analytics record*, where n is the number of entities (rows) and m is the number of attributes (columns). In data science, the terms *data set* and

analytics record are often used interchangeably, with the analytics record being a particular representation of a data set. Table 1 illustrates an analytics record for a data set of classic books. Each row in the table describes one book. The terms *instance*, *example*, *entity*, *object*, *case*, *individual*, and *record* are used in data science literature to refer to a row. So a data set contains a set of instances, and each instance is described by a set of attributes.

The construction of the analytics record is a prerequisite of doing data science. In fact, the majority of the time and effort in data science projects is spent on creating, cleaning, and updating the analytics record. The analytics record is often constructed by merging information from many different sources: data may have to be extracted from multiple databases, data warehouses, or computer files in different formats (e.g., spreadsheets or csv files) or scraped from the web or social media streams.

Table 1 A Data Set of Classic Books

ID	Title	Author	Year	Cover	Edition	Price
1	*Emma*	Austen	1815	Paperback	20th	$5.75
2	*Dracula*	Stoker	1897	Hardback	15th	$12.00
3	*Ivanhoe*	Scott	1820	Hardback	8th	$25.00
4	*Kidnapped*	Stevenson	1886	Paperback	11th	$5.00

Four books are listed in the data set in table 1. Excluding the ID attribute—which is simply a label for each row and hence is not useful for analysis—each book is described using six attributes: title, author, year, cover, edition, and price. We could have included many more attributes for each book, but, as is typical of data science projects, we needed to make a choice when we were designing the data set. In this instance, we were constrained by the size of the page and the number of attributes we could fit onto it. In most data science projects, however, the constraints relate to what attributes we can actually gather and what attributes we believe, based on our domain knowledge, are relevant to the problem we are trying to solve. Including extra attributes in a data set does not come without cost. First, there is the extra time and effort in collecting and quality checking the attribute information for each instance in the data set and integrating these data into the analytics record. Second, including irrelevant or redundant attributes can have a negative effect on the performance of many of the algorithms used to analyze data. Including many attributes in a data set increases the probability that an algorithm will find irrelevant or spurious patterns in the data that appear to be statistically significant only because of the particular sample of instances in the data set. The problem of how to choose the correct attribute(s) is a challenge faced by all data science projects, and sometimes it comes down to an iterative process of

trial-and-error experiments where each iteration checks the results achieved using different subsets of attributes.

There are many different types of attributes, and for each attribute type different sorts of analysis are appropriate. So understanding and recognizing different attribute types is a fundamental skill for a data scientist. The standard types are *numeric*, *nominal*, and *ordinal*. Numeric attributes describe measurable quantities that are represented using integer or real values. Numeric attributes can be measured on either an *interval scale* or a *ratio scale*. Interval attributes are measured on a scale with a fixed but arbitrary interval and arbitrary origin—for example, date and time measurements. It is appropriate to apply ordering and subtraction operations to interval attributes, but other arithmetic operations (such as multiplication and division) are not appropriate. Ratio scales are similar to interval scales, but the scale of measurement possesses a true-zero origin. A value of zero indicates that none of the quantity is being measured. A consequence of a ratio scale having a true-zero origin is that we can describe a value on a ratio scale as being a multiple (or ratio) of another value. Temperature is a useful example for distinguishing between interval and ratio scales.[2] A temperature measurement on the Celsius or Fahrenheit scale is an interval measurement because a 0 value on either of these scales does not indicate zero heat. So although we can compute differences between temperatures on these scales and

compare these differences, we cannot say that a temperature of 20° Celsius is twice as warm as 10° Celsius. By contrast, a temperature measurement in Kelvins is on a ratio scale because 0 K (absolute zero) is the temperature at which all thermal motion ceases. Other common examples of ratio-scale measurements include money quantities, weight, height, and marks on an exam paper (scale 0–100). In table 1, the "year" attribute is an example of an interval-scale attribute, and the "price" attribute is an example of a ratio-scale attribute.

Nominal (also known as categorical) attributes take values from a finite set. These values are names (hence "nominal") for categories, classes, or states of things. Examples of nominal attributes include marital status (single, married, divorced) and beer type (ale, pale ale, pils, porter, stout, etc.). A binary attribute is a special case of a nominal attribute where the set of possible values is restricted to just two values. For example, we might have the binary attribute "spam," which describes whether an email is spam (true) or not spam (false), or the binary attribute "smoker," which describes whether an individual is a smoker (true) or not (false). Nominal attributes cannot have ordering or arithmetic operations applied to them. Note that a nominal attribute may be sorted alphabetically, but alphabetizing is a distinct operation from ordering. In table 1, "author" and "title" are examples of nominal attributes.

Ordinal attributes are similar to nominal attributes, with the difference that it is possible to apply a rank order over the categories of ordinal attributes. For example, an attribute describing the response to a survey question might take values from the domain "strongly dislike, dislike, neutral, like, and strongly like." There is a natural ordering over these values from "strongly dislike" to "strongly like" (or vice versa depending on the convention being used). However, an important feature of ordinal data is that there is no notion of equal distance between these values. For example, the cognitive distance between "dislike" and "neutral" may be different from the distance between "like" and "strongly like." As a result, it is not appropriate to apply arithmetic operations (such as averaging) on ordinal attributes. In table 1, the "edition" attribute is an example of an ordinal attribute. The distinction between nominal and ordinal data is not always clear-cut. For example, consider an attribute that describes the weather and that can take the values "sunny," "rainy," "overcast." One person might view this attribute as being nominal, with no natural order over the values, whereas another person might argue that the attribute is ordinal, with "overcast" being treated as an intermediate value between "sunny" and "rainy" (Hall, Witten, and Frank 2011).

The data type of an attribute (numeric, ordinal, nominal) affects the methods we can use to analyze and understand the data, including both the basic statistics we can

The data type of an attribute (numeric, ordinal, nominal) affects the methods we can use to analyze and understand the data.

use to describe the distribution of values that an attribute takes and the more complex algorithms we use to identify the patterns of relationships between attributes. At the most basic level of analysis, numeric attributes allow arithmetic operations, and the typical statistical analysis applied to numeric attributes is to measure the central tendency (using the mean value of the attribute) and the dispersion of the attributes values (using the variance or standard deviation statistics). However, it does not make sense to apply arithmetic operations to nominal or ordinal attributes. So the basic analysis of these types of attributes involves counting the number of times each of the values occurs in the data set or calculating the proportion of occurrence of each value or both.

Data are generated through a process of abstraction, so any data are the result of human decisions and choices. For every abstraction, somebody (or some set of people) will have made choices with regard to what to abstract from and what categories or measurements to use in the abstracted representation. The implication is that data are never an objective description of the world. They are instead always partial and biased. As Alfred Korzybski has observed, "A map is not the territory it represents, but, if correct, it has a similar structure to the territory which accounts for its usefulness" (1996, 58).

In other words, the data we use for data science are not a perfect representation of the real-world entities and

processes we are trying to understand, but if we are careful in how we design and gather the data that we use, then the results of our analysis will provide useful insights into our real-world problems. The moneyball story given in chapter 1 is a great example of how the determinant of success in many data science projects is figuring out the correct abstractions (attributes) to use for a given domain. Recall that the key to the moneyball story was that the Oakland A's figured out that a player's on-base percentage and slugging percentage are better attributes to use to predict a player's offensive success than traditional baseball statistics such as batting average. Using different attributes to describe players gave the Oakland A's a different and better model of baseball than the other teams had, which enabled it to identify undervalued players and to compete with larger franchises using a smaller budget.

The moneyball story illustrates that the old computer science adage "garbage in, garbage out" is true for data science: if the inputs to a computational process are incorrect, then the outputs from the process will be incorrect. Indeed, two characteristics of data science cannot be overemphasized: (*a*) for data science to be successful, we need to pay a great deal of attention to how we create our data (in terms of both the choices we make in designing the data abstractions and the quality of the data captured by our abstraction processes), and (*b*) we also need to "sense check" the results of a data science process—that

is, we need to understand that just because the computer identifies a pattern in the data this doesn't mean that it is identifying a real insight in the processes we are trying to analyze; the pattern may simply be based on the biases in our data design and capture.

Perspectives on Data

Other than type of data (numeric, nominal, and ordinal), a number of other useful distinctions can be made regarding data. One such distinction is between *structured* and *unstructured* data. Structured data are data that can be stored in a table, and every instance in the table has the same structure (i.e., set of attributes). As an example, consider the demographic data for a population, where each row in the table describes one person and consists of the same set of demographic attributes (name, age, date of birth, address, gender, education level, job status, etc.). Structured data can be easily stored, organized, searched, reordered, and merged with other structured data. It is relatively easy to apply data science to structured data because, by definition, it is already in a format that is suitable for integration into an analytics record. *Unstructured data* are data where each instance in the data set may have its own internal structure, and this structure is not necessarily the same in every instance. For example, imagine a

data set of webpages, with each webpage having a structure but this structure differing from one webpage to another. Unstructured data are much more common than structured data. For example, collections of human text (emails, tweets, text messages, posts, novels, etc.) can be considered unstructured data, as can collections of sound, image, music, video, and multimedia files. The variation in the structure between the different elements means that it is difficult to analyze unstructured data in its raw form. We can often extract structured data from unstructured data using techniques from artificial intelligence (such as natural-language processing and ML), digital signal processing, and computer vision. However, implementing and testing these data-transformation processes is expensive and time-consuming and can add significant financial overhead and time delays to a data science project.

Sometimes attributes are *raw* abstractions from an event or object—for example, a person's height, the number of words in an email, the temperature in a room, the time or location of an event. But data can also be *derived* from other pieces of data. Consider the average salary in a company or the variance in the temperature of a room across a period of time. In both of these examples, the resulting data are derived from an original set of data by applying a function to the original raw data (individual salaries or temperature readings). It is frequently the case that the real value of a data science project is the identification

It is frequently the case that the real value of a data science project is the identification of one or more important derived attributes that provide insight into a problem.

of one or more important derived attributes that provide insight into a problem. Imagine we are trying to get a better understanding of obesity within a population, and we are trying to understand the attributes of an individual that identify him as being obese. We would begin by examining the raw attributes of individuals, such as their height and weight, but after studying the problem for some time we might end up designing a more informative derived attribute such as the Body Mass Index (BMI). BMI is the ratio of a person's mass and height. Recognizing that the *interaction* between the raw attributes "mass" and "height" provides more information about obesity then either of these two attributes can when examined independently will help us to identify people in the population who are at risk of obesity. Obviously, BMI is a simple example that we use here to illustrate the importance of derived attributes. But consider situations where the insight into the problem is given through multiple derived attributes, where each attribute involves two (or potentially more) additional attributes. It is in contexts where multiple attributes interact together that data science provides us with real benefits because the algorithms we use can, in some cases, learn the derived attributes from the raw data.

There are generally two terms for gathered *raw data*: *captured data* and *exhaust data* (Kitchin 2014a). *Captured data* are collected through a direct measurement or observation that is designed to gather the data. For example,

the primary purpose of surveys and experiments is to gather specific data on a particular topic of interest. By contrast, exhaust data are a by-product of a process whose primary purpose is something other than data capture. For example, the primary purpose of many social media technologies is to enable users to connect with other people. However, for every image shared, blog posted, tweet retweeted, or post liked, a range of exhaust data is generated: who shared, who viewed, what device was used, what time of day, which device was used, how many people viewed/liked/retweeted, and so on. Similarly, the primary purpose of the Amazon website is to enable users to make purchases from the site. However, each purchase generates volumes of exhaust data: what items the user put into her basket, how long she stayed on the site, what other items she viewed, and so on.

One of the most common types of exhaust data is *metadata*—that is, data that describe other data. When Edward Snowden released documents about the US National Security Agency's surveillance program PRISM, he revealed that the agency was collecting a large amount of metadata about people's phone calls. This meant that the agency was not actually recording the content of peoples phone calls (it was not doing wiretapping) but rather collecting the data about the calls, such as when the call was made, who the recipient was, how long the call lasted, and so on (Pomerantz 2015). This type of data gathering may not appear ominous, but the MetaPhone study carried

out at Stanford highlighted the types of sensitive insights that phone-call metadata can reveal about an individual (Mayer and Mutchler 2014). The fact that many organizations have very specific purposes makes it relatively easy to infer sensitive information about a person based on his phone calls to these organizations. For example, some of the people in the MetaPhone study made calls to Alcoholics Anonymous, divorce lawyers, and medical clinics specializing in sexually transmitted diseases. Patterns in calls can also be revealing. The pattern analysis from the study showed how patterns of calls reveal potentially very sensitive information:

> Participant A communicated with multiple local neurology groups, a specialty pharmacy, a rare condition management service, and a hotline for a pharmaceutical used solely to treat relapsing multiple sclerosis. … In a span of three weeks, Participant D contacted a home improvement store, locksmiths, a hydroponics dealer, and a head shop. (Mayer and Mutchler 2014)

Data science has traditionally focused on captured data. However, as the MetaPhone study shows, exhaust data can be used to reveal hidden insight into situations. In recent years, exhaust data have become more and more useful, particularly in the realm of customer engagement, where the linking of different exhaust data sets has the

potential to provide a business with a richer profile of individual customers, thereby enabling the business to target its services and marketing to certain customers. In fact, one of the factors driving the growth in data science in business today is the recognition of the value of exhaust data and the potential that data science has to unlock this value for businesses.

Data Accumulates, Wisdom Doesn't!

The goal of data science is to use data to get insight and understanding. The Bible urges us to attain understanding by seeking wisdom: "wisdom is the principal thing, therefore get wisdom and with all thy getting get understanding" (Proverbs 4:7 [King James]). This advice is reasonable, but it does beg the question of how one should go about seeking wisdom. The following lines from T. S. Eliot's poem "Choruses from The Rock" describes a hierarchy of wisdom, knowledge, and information:

> Where is the wisdom we have lost in knowledge?
> Where is the knowledge we have lost in information?
> (Eliot 1934, 96)

Eliot's hierarchy mirrors the standard model of the structural relationships between wisdom, knowledge,

information, and data known as the *DIKW pyramid* (see figure 2). In the DIKW pyramid, data precedes information, which precedes knowledge, which precedes wisdom. Although the order of the layers in the hierarchy are generally agreed upon, the distinctions between the layers and the processes required to move from one layer to the next are often contested. Broadly speaking, however,

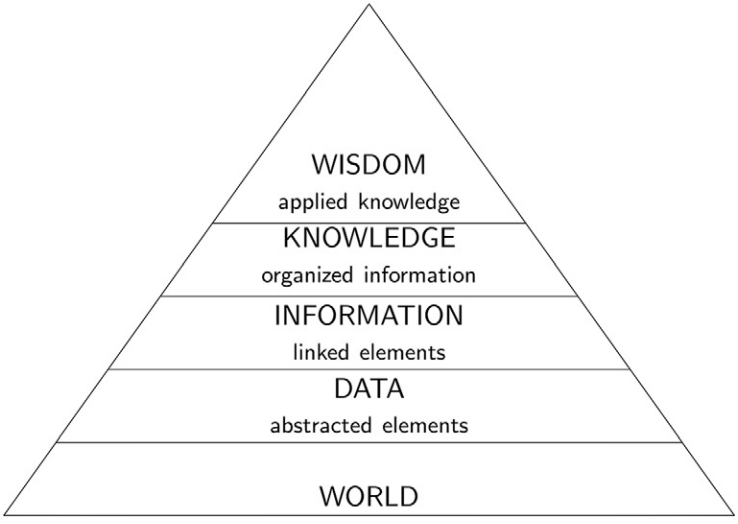

Figure 2 The DIKW pyramid (adapted from Kitchin 2014a).

- Data are created through abstractions or measurements taken from the world.
- Information is data that have been processed, structured, or contextualized so that it is meaningful to humans.
- Knowledge is information that has been interpreted and understood by a human so that she can act on it if required.
- Wisdom is acting on knowledge in an appropriate way.

The activities in the data science process can also be represented using a similar pyramid hierarchy where the width of the pyramid represents the amount of data being processed at each level and where the higher the layer in the pyramid, the more informative the results of the activities are for decision making. Figure 3 illustrates the hierarchy of data science activities from data capture and generation through data preprocessing and aggregation, data understanding and exploration, pattern discovery and model creation using ML, and decision support using data-driven models deployed in the business context.

The CRISP-DM Process

Many people and companies regularly put forward suggestions on the best process to follow to climb the data science pyramid. The most commonly used process is the

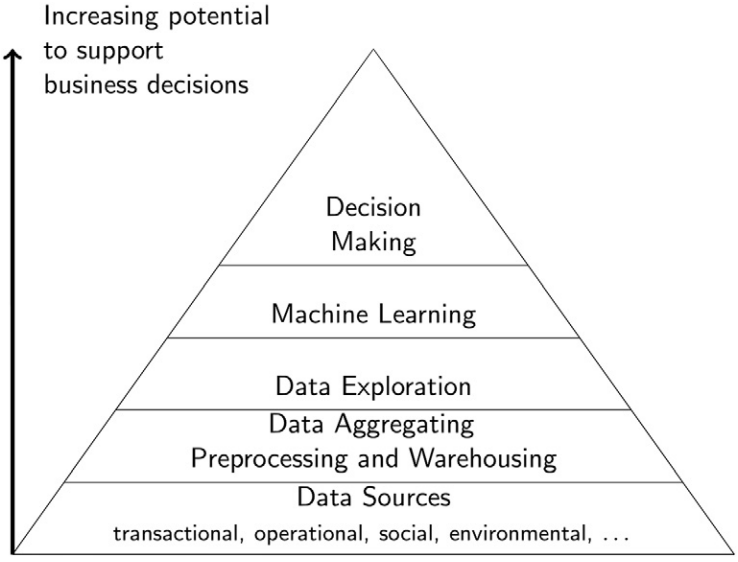

Figure 3 Data science pyramid (adapted from Han, Kamber, and Pei 2011).

Cross Industry Standard Process for Data Mining (CRISP-DM). In fact, the CRISP-DM has regularly been in the number-one spot in various industry surveys for a number of years. The primary advantage of CRISP-DM, the main reason why it is so widely used, is that it is designed to be independent of any software, vendor, or data-analysis technique.

CRISP-DM was originally developed by a consortium of organizations consisting of leading data science vendors, end users, consultancy companies, and researchers. The original CRISP-DM project was sponsored in part by the European Commission under the ESPRIT Program, and the process was first presented at a workshop in 1999. Since then, a number of attempts have been made to update the process, but the original version is still predominantly in use. For many years, there was a dedicated website for CRISP-DM, but in recent years this website is no longer available, and on occasion you might get redirected to the SPSS website by IBM, which was one of the original contributors to the project. The original consortium published a detailed (76-page) but readable step-by-step guide to the process that is freely available online (see Chapman et al. 1999), but the structure and major tasks of the process can be summarized in a few pages.

The CRISP-DM life cycle consists of six stages: *business understanding*, *data understanding*, *data preparation*, *modeling*, *evaluation*, and *deployment*, as shown in figure 4. Data are at the center of all data science activities, and that is why the CRISP-DM diagram has data at its center. The arrows between the stages indicate the typical direction of the process. The process is semistructured, which means that a data scientist doesn't always move through these six stages in a linear fashion. Depending on the outcome of a particular stage, a data scientist may go back to one of

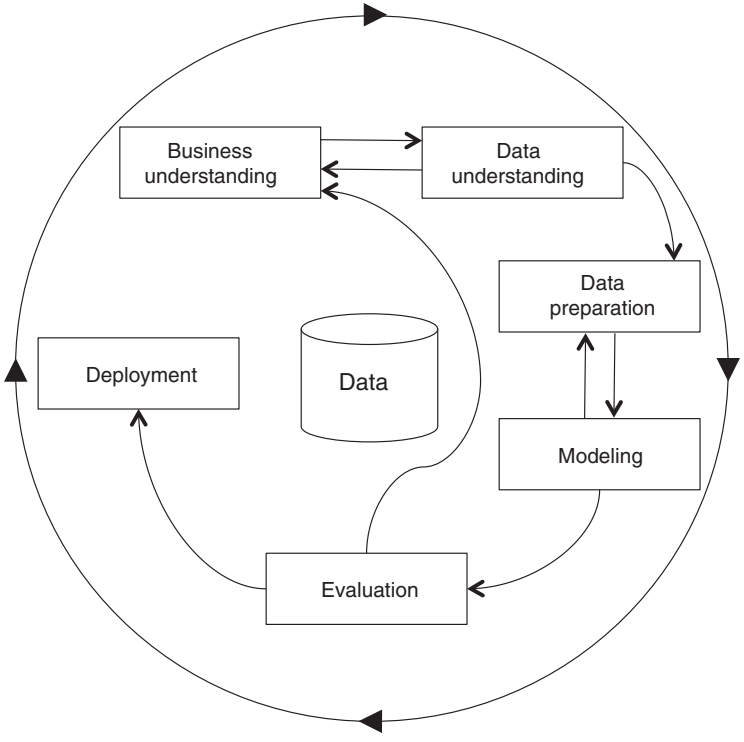

Figure 4 The CRISP-DM life cycle (based on figure 2 in Chapman, Clinton, Kerber, et al. 1999).

the previous stages, redo the current stage, or move on to the next stage.

In the first two stages, business understanding and data understanding, the data scientist is trying to define the goals of the project by understanding the business needs and the data that the business has available to it. In the early stages of a project, a data scientist will often iterate between focusing on the business and exploring what data are available. This iteration typically involves identifying a business problem and then exploring if the appropriate data are available to develop a data-driven solution to the problem. If the data are available, the project can proceed; if not, the data scientist will have to identify an alternative problem to tackle. During this stage of a project, a data scientist will spend a great deal of time in meetings with colleagues in the business-focused departments (e.g., sales, marketing, operations) to understand their problems and with the database administrators to get an understanding of what data are available.

Once the data scientist has clearly defined a business problem and is happy that the appropriate data are available, she moves on to the next phase of the CRISP-DM: data preparation. The focus of the data-preparation stage is the creation of a data set that can be used for the data analysis. In general, creating this data set involves integrating data sources from a number of databases. When an organization has a data warehouse, this data

integration can be relatively straightforward. Once a data set has been created, the quality of the data needs to be checked and fixed. Typical data-quality problems include outliers and missing values. Checking the quality of the data is very important because errors in the data can have a serious effect on the performance of the data-analysis algorithms.

The next stage of CRISP-DM is the modeling stage. This is the stage where automatic algorithms are used to extract useful patterns from the data and to create models that encode these patterns. Machine learning is the field of computer science that focuses on the design of these algorithms. In the modeling stage, a data scientist will normally use a number of different ML algorithms to train a number of different models on the data set. A model is trained on a data set by running an ML algorithm on the data set so as to identify useful patterns in the data and to return a model that encodes these patterns. In some cases an ML algorithm works by fitting a template model structure to a data set by setting the parameters of the template to good values for that data set (e.g., fitting a linear regression or neural network model to a data set). In other cases an ML algorithm builds a model in a piecewise fashion (e.g. growing a decision tree one node at a time beginning at the root node of the tree). In most data science projects it is a model generated by an ML algorithm that is ultimately the software that is deployed by an organization to help it

solve the problem the data science project is addressing. Each model is trained by a different type of ML algorithm, and each algorithm looks for different types of patterns in the data. At this stage in the project, the data scientist typically doesn't know which patterns are the best ones to look for in the data, so in this context it makes sense to experiment with a number of different algorithms and see which algorithm returns the most accurate models when run on the data set. In chapter 4 we will introduce ML algorithms and models in much more detail and explain how to create a test plan to evaluate model accuracy.

In the majority of data science projects, the initial model test results will uncover problems in the data. These data errors sometimes come to light when the data scientist investigates why the performance of a model is lower than expected or notices that maybe the model's performance is suspiciously good. Or by examining the structure of the models, the data scientist may find that the model is reliant on attributes that she would not expect, and as a result she revisits the data to check that these attributes are correctly encoded. It is thus not uncommon for a project to go through several rounds of these two stages of the process: modeling, data preparation; modeling, data preparation; and so on. For example, Dan Steinberg and his team reported that during one data science project, they rebuilt their data set 10 times over a six-week period, and in week five, having gone through a number of iterations of data

cleaning and preparation, they uncovered a major error in the data (Steinberg 2013). If this error had not been identified and fixed, the project would not have succeeded.

The last two stages of the CRISP-DM process, evaluation and deployment, are focused on how the models fit the business and its processes. The tests run during the modeling stage are focused purely on the accuracy of the models for the data set. The evaluation phase involves assessing the models in the broader context defined by the business needs. Does a model meet the business objectives of the process? Is there any business reason why a model is inadequate? At this point in the process, it is also useful for the data scientist to do a general quality-assurance review on the project activities: Was anything missed? Could anything have been done better? Based on the general assessment of the models, the main decision made during the evaluation phase is whether any of the models should be deployed in the business or another iteration of the CRISP-DM process is required to create adequate models. Assuming the evaluation process approves a model or models, the project moves into the final stage of the process: deployment. The deployment phase involves examining how to deploy the selected models into the business environment. This involves planning how to integrate the models into the organization's technical infrastructure and business processes. The best models are the ones that fit smoothly into current practices.

Models that fit current practices have a natural set of users who have a clearly defined problem that the model helps them to solve. Another aspect of deployment is putting a plan in place to periodically review the performance of the model.

The outer circle of the CRISP-DM diagram (figure 4) highlights how the whole process is iterative. The iterative nature of data science projects is perhaps the aspect of these projects that is most often overlooked in discussions of data science. After a project has developed and deployed a model, the model should be regularly reviewed to check that it still fits the business's needs and that it hasn't become obsolete. There are many reasons why a data-driven model can become obsolete: the business's needs might have changed; the process the model emulates and provides insight into might have changed (for example, customer behavior changes, spam email changes, etc.); or the data streams the model uses might have changed (for example, a sensor that feeds information into a model may have been updated, and the new version of the sensor provides slightly different readings, causing the model to be less accurate). The frequency of this review is dependent on how quickly the business ecosystem and the data that the model uses evolve. Constant monitoring is needed to determine the best time to go through the process again. This is what the outer circle of the CRISP-DM process shown in figure 4 represents. For example, depending on

the data, the business question, and the domain, you may have go through this iterative process on a yearly, quarterly, monthly, weekly, or even daily basis. Figure 5 gives a summary of the different stages of the data science project process and the major tasks involved in each phase.

A frequent mistake that many inexperienced data scientists make is to focus their efforts on the modeling stage of the CRISP-DM and to rush through the other stages. They may think that the really important deliverable from a project is the model, so the data scientist should devote most of his time to building and finessing the model. However, data science veterans will spend more time on ensuring that the project has a clearly defined focus and that it has the right data. For a data science project to succeed, a data scientist needs to have a clear understanding of the business need that the project is trying to solve. So the business understanding stage of the process is really important. With regard to getting the right data for a project, a survey of data scientists in 2016 found that 79 percent of their time is spent on data preparation. The time spent across the major tasks in the project was distributed as follows: collecting data sets, 19 percent; cleaning and organizing data, 60 percent; building training sets, 3 percent; mining data for patterns, 9 percent; refining algorithms, 4 percent; and performing other tasks, 5 percent (CrowdFlower 2016). The 79 percent figure for preparation comes from summing the time spent on collecting, cleaning, and

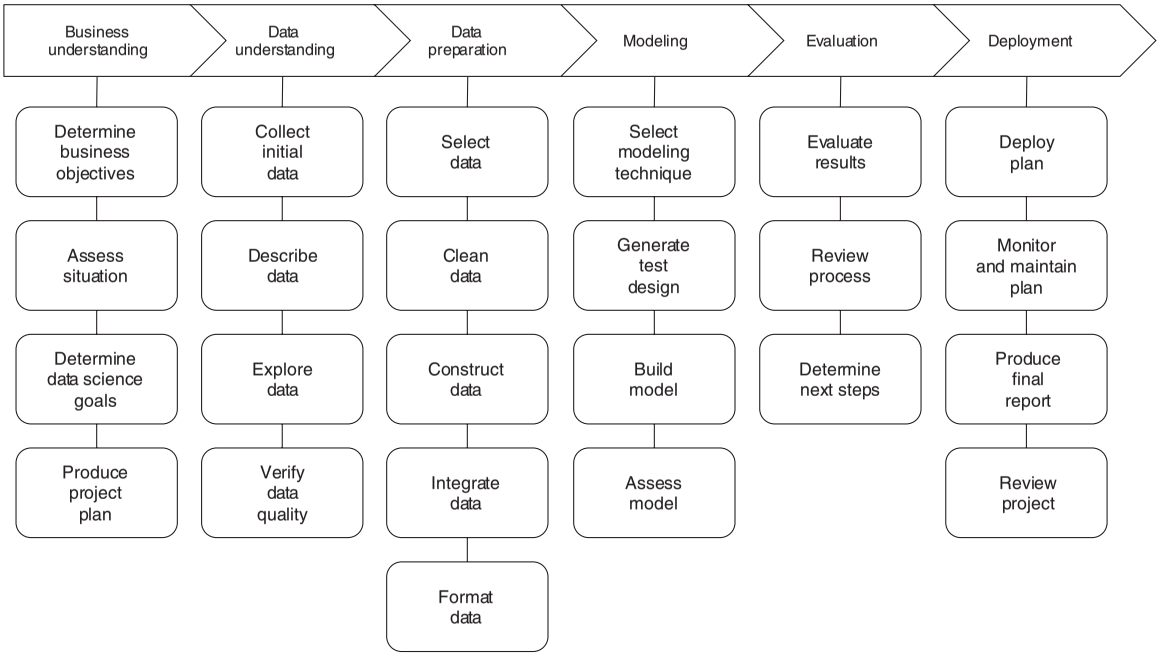

Figure 5 The CRISP-DM stages and tasks (based on figure 3 in Chapman, Clinton, Kerber, et al. 1999).

organizing the data. That around 80 percent of project time is spent on gathering and preparing data has been a consistent finding in industry surveys for a number of years. Sometimes this finding surprises people because they imagine data scientists spend their time building complex models to extract insight from the data. But the simple truth is that no matter how good your data analysis is, it won't identify useful patterns unless it is applied to the right data.