
Chapter 21 Introduction to Big Data and Data Mining



21.1 Data Mining and the Big Data Revolution

21.2 The Data Mining Process

21.3 Data Mining Algorithms: A Sample

21.4 Models Built from Combining Other Models

21.5 Comparing Models

21.6 Summary

Paralyzed Veterans of America

The Paralyzed Veterans of America (PVA) is a philanthropic and service organization chartered by the U.S. government to serve the needs of U.S. veterans who suffer from spinal cord injury or disease. Since 1946, PVA has raised money to support a variety of activities, including advocacy for veterans'

health care, research and education in spinal cord injury and disease, and support for veterans' benefits and rights.

PVA raises money in a variety of ways, but the majority of their fund-raising comes from direct mail campaigns. They send free address labels or greeting cards to potential donors on their mailing list and request a donation for receiving these gifts.

PVA sends out regular solicitations to a list of more than four million donors. In 2016, PVA received about \$100 million in donations, but the effort cost them over \$40 million in postage, administrative, and gift expenses. Much of that money was spent on mailings to people who never responded. In fact, in any one solicitation, groups like PVA are lucky to get responses from more than a few percent of the people they contact. Response rates for commercial companies such as large credit card banks are so low that they often measure responses not in percentage points, but in basis points—hundredths of a percent. If the PVA could avoid mailing to just half of the people who won't respond, they could save \$20 million a year and produce only half as much wasted paper. Can statistical and data mining methods help them decide who should get what mail?

21.1 Data Mining and the Big Data Revolution

You learned about Big Data in [Chapter 1](#), and you hear about it on the news almost every day. The technological advances of the past few decades have enabled us to capture information virtually everywhere and automatically. When you visit a social networking site, data about you, your likes, your photos, your friends, and your habits are uploaded to servers, waiting for analysis. Your smartphone sends information about the calls you make, the texts you send, and even your location to central servers. Whether this information is being used in a prudent way by the government and industry is the subject of recent news and citizen concerns,¹ but the collection of data by business, science, and government is exploding and will continue to explode into the foreseeable future. And these data will be used to solve problems using techniques that are based on methods similar to those you've learned to master in this course.

¹"Momentum Builds Against N.S.A. Surveillance," *New York Times*, July 28, 2013.

Terms like *data mining*, *data science*, and *analytics* are all used to describe different parts of the process of curating data and applying statistical methods to large datasets. In particular **Data science** has come to mean the entire field of collecting, managing, curating, and analyzing data and then communicating the results. In [Chapter 1](#), we defined **data mining** as the process that uses a variety of data analysis tools to discover patterns and relationships in data to help build useful models and make predictions. The more general term **business analytics** (or sometimes simply *analytics*) describes any use of statistical analysis to drive business decisions from data, whether the purpose is predictive or simply descriptive. **Big data** is a term for datasets so large and complex that it becomes difficult to use traditional methods to capture, store, visualize, and analyze them (see sidebar). But the definition of how "big" the database should be to qualify keeps changing with advancing computer speeds and storage capabilities.

What Makes Data "Big"? The Four V's

The size or "volume" of the dataset is just one of the aspects that make data "big," but complexity is equally important. Scientists at IBM have defined the 4 V's that make Big Data truly big: Volume, Variety, Velocity, and Veracity. So, not only is volume important, but also different types of data (variety), the speed and timeliness at which it is collected (velocity), and the quality (or lack thereof) of the data (veracity) make Big Data challenging to manage and analyze.

What can companies hope to learn from all these data, and how do they analyze them? With information about the transactions you make, the websites you visit, and the social media you interact with, retailers can customize offers to you and predict when and how to deliver those offers to maximize the chance that you will purchase. Advertisers can place their message in specific channels to maximize the impact for the group of consumers they are targeting.

The Obama presidential campaign used analytics rather than conventional wisdom to determine when and where to place ads during the fall of 2012. As part of their sophisticated targeting operations, they bought “detailed data on TV viewing by millions of cable subscribers, showing which channels they were watching, sometimes on a second-by-second basis.”² The campaign then “used a third-party company to match viewing data to its own internal list of voters and poll responses,” helping to make the most out of their advertising budget during the final weeks of the 2012 campaign. The *New York Times* reported that the system “allowed Mr. Obama’s team to direct advertising with a previously unheard-of level of efficiency,”³ with some analysts crediting this data mining with Obama’s victory.

²“Obama Campaign Took Unorthodox Approach to Ad Buying,” *Washington Post*, November 14, 2012.

³“Secret of the Obama Victory? Rerun Watchers, for One Thing,” *New York Times*, November 12, 2012.

What Makes Big Data Big? An Example

The U.S. Census Bureau collects information about nearly every person living in the United States. You might think that this would be Big Data. Although it is certainly voluminous, it lacks at least two crucial criteria: velocity and variety. Currently the Bureau collects demographic data only every 10 years on the entire population. It gathers economic data more frequently, but on smaller samples. According to Census Bureau Chief of Research and Methodology Tommy Wright, adding real-time information about individuals’ purchases through credit and loyalty card transactions and linking those to the demographics would go a long way to helping provide a more accurate and instantaneous picture of the state and health of our economy.

Most of the models used in analytics and data mining are based on the regression ideas that you’ve studied in previous chapters. But because data mining has benefited from work in other fields as well, it has a much richer set of tools. Some of the techniques enable more automatic re-expressions and more flexible fitting than the multiple regression and time series methods you saw in **Chapters 18**, **19**, and **20**, but they are based on the same ideas. The skills you’ve acquired by learning to fit models, diagnose them, and assess their limitations already prepare you to start entering the world of big data. The collection of models used is sometimes referred to as **machine learning algorithms** because of the role computers have in fitting these models (adding even more hype to a collection of models).

The Goals of Data Mining

The broad purpose of data mining is to extract useful information hidden in large databases. With a database as large as a typical data warehouse (see **Chapter 1**), that search can be like looking for a needle in a digital haystack. How can analysts hope to find what they’re looking for? They may start their search by using a sequence of queries to deduce facts about customer behavior, asking specific questions based on the data. Guided by their knowledge of the specific business, they may try a query-driven approach, asking a series of specific questions to deduce patterns. Such an approach is **online analytical processing** or **OLAP**. Analysts in sales, marketing, budgeting, inventory, and finance often use OLAP

to answer specific questions involving many variables. An OLAP question for the PVA might be: “What percentage of potential donors under the age of 65 with incomes between \$40,000 and \$60,000 in the Western region who have not donated in the previous two years gave more than \$25 to the most recent solicitation?” Although OLAP is efficient in answering such multivariable queries, it is question-specific. OLAP produces answers for specific queries, but does not build a predictive model, and so it is not possible to generalize using OLAP.



Data mining can address different types of problems, some of which we have already encountered. When the goal is to predict a quantitative response variable, the problem is generically called a **regression problem** ⓘ, regardless of whether linear regression is used (or even considered) as one of the models. When the response variable is categorical, the problem is referred to as a **classification problem** ⓘ because the model will either guess the most likely category for each or assign a probability to each class for that code. For example, for the classification problem of predicting *whether or not* a particular donor will give to the next campaign, a model will produce either the most probable class (*donate or not*) or the probabilities of each of these. Predicting the *amount* of money a donor will donate is a regression problem.

In contrast to an OLAP query, the outcome of a data mining analysis is a **predictive model** ⓘ—a model that uses predictor variables to predict a response. For a quantitative response variable (as in linear regression), the model will predict the *value* of the response, whereas for a categorical response, the model estimates the most probable category of the response, often with the probability that the response variable is in that class (as in logistic regression). Both linear and logistic multiple regressions are common tools of a data miner. Query-based methods like OLAP are restricted only to the data at hand. But data mining models, like all statistical models, are predictive and can be used to make decisions about future situations.

For example, an analyst using an OLAP query might find that customers in a certain age group responded to a recent product promotion. But without building a model, the analyst can't understand the relationship

between the customer's age and the success of the promotion, and thus may be unable to predict how the product will do in a more general setting. The goal of a data mining project is to increase business understanding and knowledge by building a model to answer a specific set of questions raised at the beginning of the project that can generalize to future situations.

Both regression and classification problems are referred to as **supervised problems**  because for the data on which the model is based, the response variable's outcome is known. For example, for the PVA data we encounter, we *know* for at least one group of donors the values of the responses *TARGET-B* (did you give?) and *TARGET-D* (how much did you give?). We use these values to construct the model. All the models you've seen so far in the text have been for supervised problems.

By contrast, there are problems for which there is no variable that measures what we are trying to predict. For example, a company may want to cluster their customers into groups with similar buying behaviors and tastes. That's a fine goal, but, unfortunately, customers are not labeled this way. In this case, there is no response variable—we just want to group together similar customers. Whether the grouping we find is “correct” is impossible to determine. Whether it is useful has to be determined by the person who will actually use the results. Problems like these are known as **unsupervised problems**  because there is no actual response variable.

Cluster analysis , grouping similar objects together, is the most common type of unsupervised problem. Many different algorithms are available to find clusters, but they all rely on a measure of how similar (or, equivalently, how distant) the objects are to each other. For example, when clustering bank customers, we might base the distance between them on the sum of the total (absolute) differences that they spend in different categories. If customer A spent \$100 on travel and \$300 on restaurants last month and customer B spent \$200 on travel and \$200 on restaurants, a possible distance between them might be $|100-200| + |300-200| = 200$. Customers who spend most similarly would be the first to be put into the same spending group (see sidebar). (Further discussion of clustering and other unsupervised methods is beyond the scope of this text.)

Data Mining—Some History

In the 1980s, credit card companies realized that with the incredible amount of data they routinely had to collect, there might be some extra value in the data, other than simply recording all their transactions. They knew that customers had different “profiles.” For example, some customers used their cards primarily for travel, while others used them for everyday purchases. This market segmentation was enabled by analyzing the transactional data.

By putting customers into different groups, credit card companies could more effectively choose the offers each group would receive.

Data mining is similar to traditional statistical analysis in that it involves exploratory data analysis and modeling. However, several aspects of data mining distinguish it from more traditional statistical analysis. Although there is no consensus on exactly what data mining is and how it differs from statistics, some of the most important differences include:

- **The size of the databases.** Although no particular size is required for an analysis to be considered data mining, an analysis involving only a few hundred cases or only a handful of variables would not usually be considered data mining.
- **The exploratory nature of data mining.** Unlike a statistical analysis that might test a hypothesis or produce a confidence interval, the outcome of a data mining effort is typically a model used for prediction. Usually, the data miner is not interested in the values of the parameters of a specific model or in testing hypotheses. A classic example is predicting ZIP codes from hand-written envelopes. The manager of the New York Post Office is unlikely to care what variables were included in the algorithm that reads the correct ZIP code automatically with a high level of accuracy.
- **The data are “happstance.”** In contrast to data arising from a designed experiment or survey, the typical data on which data mining is performed have not been collected in a systematic way. This is an important difference, often overlooked in business. Warnings not to confuse association with causation are especially pertinent for data mining. In addition, the sheer number of variables involved makes any search for relationships among variables prone to Type I errors. Humans love to see signals and relationships among variables even when none exists. Resist the temptation to believe too strongly in these models. Although they can be useful to generate hypotheses to be tested, they are, by nature, exploratory. Be especially careful to check the “pedigree” of the data. The phrase “Garbage in, garbage out” was never more true than when applied to data mining.
- **The results of a data mining effort are “actionable.”** For business applications, there should be a consensus of what the problem of interest is and how the model will help solve the problem. There should be an action plan in place for a variety of possible outcomes of the model. Exploring large databases out of curiosity or just to see what they contain is not likely to be productive.
- **The modeling choices are automatic.** Typically the data miner will try several different types of models to see what each has to say, but doesn’t want to spend a lot of time choosing which variables to include in the model or being careful in the kinds of choices he or she might make in a more traditional statistical analysis. Unlike the analyst who wants to understand the terms in a stepwise regression, the data miner is more concerned with the predictive ability of the model. As long as the resulting model can help make decisions about who should receive the next offer, who is best suited for an online coupon, or who is most likely to switch cable providers next month, the data miner is likely to be satisfied.

Data Mining Myths

Data mining software usually contains a variety of exploratory and model-building tools and a graphical user interface designed to guide the user through the data mining process. Some people buy data mining software in the hope that the tools will find information in their databases and write reports that spew knowledge with little or no effort or input from the user. Software vendors often capitalize on this hope by exaggerating the capabilities and the automatic nature of data mining to increase sales of their software. Data mining can often assist the analyst to find meaningful patterns and help predict future customer

behavior, but the more the analyst knows about his or her business, the more likely he or she will be successful using data mining. Data mining is not a magic wand that can overcome poor data quality or collection. A product may have tools for detecting outliers and may be able to impute (assign a value for) missing data values, but all the issues you learned about good statistical analysis are still relevant for data mining.

Here are some of the more common myths about what data mining can do.

- **Myth 1:** Find answers to unasked questions. Even though data mining can build a model to help answer a specific question, it does not have the ability to answer questions that haven't been asked. In fact, formulating a precise question is a key first step in any data mining project.
- **Myth 2:** Automatically monitor a database for interesting patterns. Data mining techniques build predictive models and can answer queries, but do not find interesting patterns on their own any more than regression models do.
- **Myth 3:** Eliminate the need to understand the business. In fact, the more an analyst understands his or her business, the more effective the data mining effort will be.
- **Myth 4:** Eliminate the need to collect good data. Good data are as important for a data mining model as for any other statistical model that you've encountered. While some data mining software contains tools to help with missing data and data transformations, there is no substitute for quality data.
- **Myth 5:** Eliminate the need for good data analysis skills. The better the data analysis skills of the miner—the skills you've learned in every chapter of this text—the better the analysis will be when using data mining tools. Data mining tools are more powerful and flexible than a statistical tool like regression but are similar in the way they work and in how they're implemented.
- **Myth 6:** Machine learning algorithms think like humans. Some of the hype about data mining and machine learning revolves around the belief that these algorithms, or models, think and learn. In fact, they "learn" about the relationship between the predictors (x) and response (y) no more, and no less, than linear regression learned about it. They simply fit a model to the data. Some do it in an automatic way, which may lead one to think they are actually learning, but this is simply a human interpretation of a powerful algorithm.

Just Checking

A bank vice president brings in her analytic team to talk about several things she wants to know about a recent offer the bank mailed out to open an IRA (individual retirement account). For each of the following, identify if it's an appropriate data mining problem. If so, is it a supervised or an unsupervised problem? If supervised, is it a regression or a classification problem?

1. Why did some people not respond to the offer?
2. What percentage of those under 65 who have been bank customers for more than 5 years responded to the offer?
3. Using the bank products that customers currently use and other demographic information, group customers into different segments for the purposes of sending different retirement offers.
4. On the basis of past data, what is the chance that a given customer will accept a new offer that the bank sends out?

5. How many transactions is a given customer likely to make next month based on their past behavior and other demographics?

Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

21.2 The Data Mining Process

Data mining projects require a number of different skills. For that reason, a data mining project should be a team effort. No one person is likely to have the business knowledge, computer and database management skills, software expertise, and statistical training needed for all the steps in the process. Because data mining projects tend to be complex, it's useful to map out the steps for a successful project. A group of data mining experts have shared their combined expertise in a project called the Cross Industry Standard Process for Data Mining (CRISP-DM).⁴ A CRISP-DM schematic of the data mining cycle appears in **Figure 21.1**.

⁴In 2015, IBM announced an updated version of CRISP-DM called ASUM-DM, which stands for Analytics Solutions Unified Method for Data Mining/Predictive Analytics, but the basic principles remain the same.

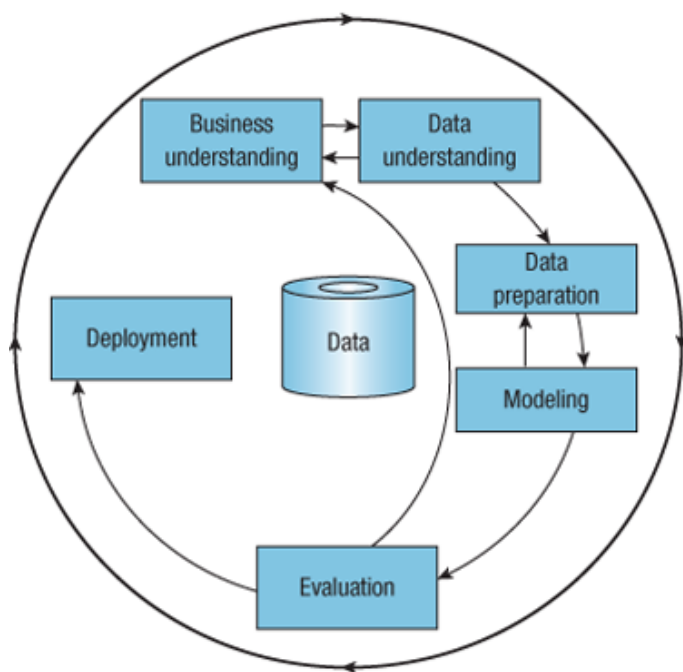


Figure 21.1

A picture of the CRISP-DM data mining process.

Let's use PVA as an example and follow the data mining cycle from beginning to end. The process starts with the *business understanding* phase, in which the problem to be addressed is carefully articulated. For any data exploration, it is best to have a specific problem before starting, but for data mining, with potentially very large databases, this is absolutely crucial. For PVA, the goal of "understanding how to best manage donors" might sound like a good idea, but is not precise enough for a data mining project. A better, more specific question might be to build a model to discover which donors are most likely to give enough in the next solicitation to justify sending them a free gift. It's important to have all the members of the data mining team involved at this stage, and the team should be representative of all parts of the business that may be affected by the resulting decisions. If key constituents aren't represented, the model is in danger of not

answering the right question. It can't be overemphasized how important it is to have consensus on a precise, correctly formulated problem to be addressed before continuing with the data mining effort.

The next phase of *data understanding* is central to the entire data mining project. If you want to know which donors are likely to participate in the next mailing, you have to have data that can actually answer that question. This is where many data mining projects fail. A traditional statistics approach might be to survey donors before the next mailing, but, typically, a data mining project uses only data the company has already collected. Much of that data is **transactional data** ⓘ—data that simply record a transaction between the customer and the company. This could be a purchase, a donation, or any other transaction, even a phone call to an 800 number or a web query. Other data are **demographic** ⓘ—information about the customer that might include the customer's income, number of children, education, and other customer-specific information. This information may have been supplied by the customer or may be purchased from other companies. Still other data are **historical** ⓘ based on past history between the customer and the company. These data sources are typically stored on different computers or servers, which means that they need to be merged before the analyst can access them. Usually, someone well versed in database architecture and data storage is needed to extract the relevant information in different databases and to merge them into a form that is useful for analysis. This requires teamwork and communication skills. And, unfortunately, all of this means that the answer to a specific question may not be found in these data, even when using a very powerful algorithm designed to search for it.

Customer-specific data can also be purchased from a variety of commercial organizations. For example, a credit card company may want to send out an offer of free flight insurance to customers who travel frequently. To help know which customers those are, the company may buy information about customers' magazine subscriptions to identify customers who subscribe to travel or leisure magazines. The sharing and selling of individual information is controversial and raises privacy concerns, especially when the purchased data involve health records and personal information. In fact, concerns over the sharing of health data in the United States led to a set of strict guidelines known as HIPAA (Health Insurance Portability and Accountability Act) regulations. Restrictions on the collection and sharing of information about customers vary widely from country to country.

The PVA customer database⁵ is a typical mix of variables. It contains 481 demographic and historical variables on each of nearly 100,000 donors. There are 479 potential predictor variables and two responses: *TARGET_B*, a 0/1 variable that indicates whether the donor contributed to the most recent campaign, and *TARGET_D*, that gives the dollar amount of the contribution. The demographic variables are both donor specific (e.g., age, income, number of children) and ZIP code specific (e.g., percent with college education, average income, percent minority). Both types of variables can be collected by the company itself or purchased from other organizations, both private and governmental (the U.S. Census Bureau is a source of information at various levels). About 10% of the PVA variables describe past giving behavior collected by the PVA itself. More than half of the variables are regional data based on ZIP codes, most likely purchased from the Census Bureau. The rest of the variables are donor specific, either gathered by the PVA or purchased from other organizations. The historical data pertain to the past behavior of the donor and include whether and how much the individual donated to various campaigns in the past. **Table 21.1** ⓘ shows the first 18 records for a subset of the 481 variables found in the PVA data. We can guess what some of the variables mean from their names, but others are more mysterious.

5The PVA made some of their data available for the 1998 Knowledge Discovery and Data Mining (KDD) contest. The object of the contest was to build a model to predict which donors should receive the next solicitation based on demographic and past giving information. The results were presented at the KDD conference (www.kdnuggets.com). The variables discussed in this chapter are some of the ones that PVA made available. We have used various subsets of the PVA dataset in previous chapters, sometimes changing the variable names for clarity. The entire dataset can be found at the University of California at Irvine Data Repository: kdd.ics.uci.edu/databases/kddcup98/kddcup98.html.

Table 21.1

Part of the customer records from the PVA dataset. Shown here are 15 of the 481 variables and 18 of the nearly 100,000 customer records used in the 1998 KDD (Knowledge Discovery and Data Mining) data mining competition.⁶ The actual PVA customer database contains several million customer records.

ODATEDW	OSOURCE	TCODE	STATE	ZIP	DOB	RFA_2A	AGE	OWN	INC	SEX	WEALTH	AVGGIFT	TARGET_B	TARGET_D
9401	L16	2	GA	30738	6501	F	33	U	5	F	2	11.66667	0	0
9001	L01	1	MI	49028	2201	F	76	H	1	M	2	8.777778	0	0
8601	DNA	1	TN	37079	0	E		U	1	M		8.619048	1	10
8601	AMB	1	WI	53719	3902	G	59			M		16.27273	0	0
8601	EPL	2	TX	79925	1705	E	81	H	2	F	6	10.15789	0	0
8701	LIS	1	IN	46771	0	F				M		8.871333	0	0
9201	GRI	1	IL	60016	1807	F	79	H	4	M	6	13.8	0	0
9401	HOS	0	KS	67218	5001	G	48	U	7	F	7	18.33333	0	0
8901	DUR	0	MI	48304	1402	F	84	H	7	M	9	12.90909	0	0
8601	AMB	0	FL	34746	1412	F	83	H	2	F	3	9.090909	0	0
9501	CWR	2	LA	70582	0	D		U	5	F	5.8		0	0
9501	ARG	0	MI	48312	4401	E	54			F	8		0	0
8601	ASC	0	TX	75644	2401	G	74	U	7	F		13.20833	0	0
9501	DNA	28	CA	90059	2001	E	78	H	1	F		10	0	0
9201	SYN	0	FL	33167	1906	F	79	U	2	M	3	10.09091	0	0
9401	MBC	2	MO	63084	3201	F	66	H	5	F		10	0	0
9401	HHH	28	WI	54235	0	F		H	7	F		20	0	0
9101	L02	28	AL	36108	4006	F	58	H	5	F		10.66667	0	0

6The KDD Cup is the leading data mining competition in the world and is organized by the SIGKDD interest group of the ACM (Association for Computing Machinery).

Becoming familiar with a complex dataset can take many months. Information about the variables, including their definitions, how they are collected, the date of collection, and so forth, is collectively called **metadata** ⓘ. The variables in the PVA dataset shown in **Table 21.1** ⓘ are typical of the types of data found in the customer records of many companies. We shouldn't be surprised to find out that *AGE* is the donor's age measured in years and that *ZIP* is the donor's postal ZIP code. But without the metadata, it would be hard to know that *TCODE* is the code used before the title on the address label (0=blank, 1=Mr., 2=Mrs., 28=Ms., and so on) or that *RFA_2A* is a summary of past giving.

In response to this question on Quora (quora.com), James Madison (data scientist) answered:

1. Clean data.
2. Clean data.
3. –Clean data.
13. Do some math.
14. Try to get everyone to understand my findings.
25. –Try to get everyone to understand my findings.
26. Repeat.

The choice of the variables to explore depends on the business knowledge of the team. As is clear from the PVA data, it is probably inefficient to spend a lot of time exploring each variable. This is one of the great challenges of a data mining project. At the beginning of the project, it's best to include as many variables as possible that might be useful, but keep in mind that having *too* many variables can make the model selection phase more difficult.

Once the variables have been selected and the response variable(s) agreed upon, it's time to begin the *data preparation* phase for modeling. As mentioned earlier, this can be a time-consuming part of the process and is also likely to be a team effort (see sidebar). Investigating missing values, correcting wrong and inconsistent entries, reconciling data definitions, and merging data sources are all challenging issues. Some of these can be handled automatically, while others require painstaking, detailed analysis. The team will have to decide how much effort is reasonable to make the dataset as complete and reliable as possible, given the time and resource constraints of the project.

Part of the data preparation phase involves both transforming predictor variables (as we did in [Chapter 17](#)) and creating new variables from combinations of them. This is often referred to as **feature creation**. A new feature (variable) can be created *a priori* from the analyst's domain knowledge or sometimes more automatically by the data mining algorithms themselves. For example, a banker trying to model the chance of customer bankruptcy might see that customer *debt* and *income* have been measured and, from his or her experience, decide to create the *debt/income* ratio. For the Paralyzed Veteran's data, the numerous past giving variables were combined into three new variables that summarized when the donor last contributed (called *recency*), how often the donor gives (*frequency*), and how much the donor gives (*monetary*). This type of RFM (recency, frequency, monetary) analysis is widely used in marketing science.

Once the data have been prepared, the analysts begin the *modeling* phase by exploring and developing models. If the number of variables is very large, a preliminary, exploratory model (such as a decision tree, described in a later section) might be a good way to narrow the candidate predictor variables down to a reasonable number. However, if the number of predictors is reasonably small, modelers can use traditional graphical displays (histograms, bar charts) to explore each variable and then investigate the relationship between each predictor and the response with displays such as scatterplots, boxplots, or segmented bar charts. The more knowledge of the data and the variables that go into the model, the higher the chance of success for the project.

Once the collection of variables has been determined, the analysts typically fit several models for the response variable. At this point the *evaluation* phase can begin. Looking at the structure of each model, including which predictors are important for each model, should give the analysts information about the relationship between the predictors and the response that each model is built on. In the *evaluation* phase, the candidate models are tested by predicting the response values on the test set as we discussed in [Chapter 19](#) , Section 4. Various criteria are used to judge the models, depending on the type of problem. For example, for a regression problem with a quantitative response, the sum of squares of the residuals (or a version of R^2 —discussed in the next section) on the test set might be compared across models. For a problem predicting a two-level categorical variable (with levels NO and YES), the two types of errors that arise (predicting YES when the true response is NO and vice versa) can be balanced in different ways. Placing different costs on the two types of errors might be warranted, and the total cost of misclassification should reflect that (see sidebar).

Why Not Just Report the Total Error Rate When Judging Models?

Fraud detection is a huge problem in the credit card industry and costs billions of dollars a year. Fortunately, it is fairly rare, affecting less than 1% of all transactions (wallethub.com/edu/credit-debit-card-fraud-statistics/25725/). So, simply reporting all transactions as fraud-free will result in an overall error rate of less than 1%. An accuracy rate of greater than 99% may sound good, but it's much better to look at the two types of errors (that we learned about in [Chapter 13](#) ) , separately, even weighing the different costs of each mistake.

At this point, the business question that motivated the project should be revisited. Does the model help to answer the question? If not, it might be necessary to go back to one of the previous steps to investigate why that happened. This kind of cycling through the various steps of the process is not uncommon. The reason that so many phases of the CRISP-DM diagram ([Figure 21.1](#) ) have arrows both to and from them is that this process is not a straightforward movement through the phases—it is an iterative and interconnected process.

If a particular model (or average of several models) seems to give insight into the business problem and predicts well enough to be useful, then it's time for the *deployment* phase. Usually, that means using the model to predict an outcome on a larger dataset than the original, or on more recent data than were used to build the model. Knowledge gained at any phase may trigger a reexamination of an earlier phase. Even the “final” phase, deployment, is not really final. In many business situations, the environment changes rapidly, so models can become stale quickly. Although a data mining project is complex and involves the efforts of many people, mapping out the different phases can help ensure that the project is as successful and reproducible as possible.

Successful Data Mining

The size of a typical data warehouse makes any analysis challenging. The ability to store data is growing faster than the ability to use it effectively. Commercial data warehouses often contain terabytes (TB)—more

than 1,000,000,000,000 (1 trillion) bytes—of data (1 TB is equivalent to about 260,000 digitized songs), and warehouses containing petabytes (PB—1 PB=1000 TB) are now common. According to *Gizmodo* magazine⁷, about 300 million photos and 500 TB of data are uploaded to Facebook every day. All the U.S. Census data from 1790 to 2000 would take about 600 TB. Data miners hope to uncover some important strategic information lying hidden within these massive collections of data.

⁷gizmodo.com/5937143/what-facebook-deals-with-everyday-27-billion-likes-300-million-photos-uploaded-and-500-terabytes-of-data

As we have seen, to have a successful data mining outcome, the first step is to have a well-defined business problem. With 500 variables, there are over 100,000 possible two-way relationships between pairs of variables. The number of pairs that will be related just by chance is likely to be large. And it's human nature to find many of these relationships interesting and even to posit plausible reasons why two variables might be associated. Some of these variables may appear to provide a useful predictive model, when in fact they do not. A well-defined business objective can help avoid going down a lot of blind paths.

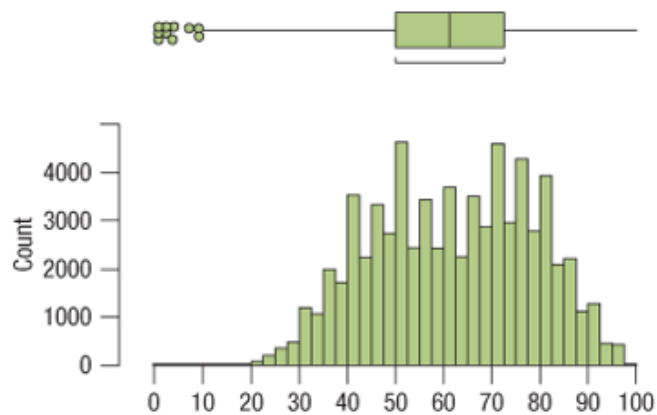
As in painting a house, much of the effort of a data mining project lies in preparation and cleaning. And just as a painter would much rather get right to the more enjoyable part of painting, unless enough time is spent prepping the wood, all that effort will ultimately be wasted. It is estimated that for any data mining project, 65% to 90% of the total project time is spent in such data preparation—investigating missing values, correcting wrong and inconsistent entries, reconciling data definitions, and possibly creating new variables from the original ones. The data may need to be extracted from several databases and combined. Errors must be corrected or eliminated, and outliers must be identified.

So, a successful data mining effort often requires a substantial amount of time devoted to basic preparation of the data before any modeling is performed. It will have a clear objective, agreed to by a team of people who will share the work and the responsibility of the data mining. It will have an action plan once the results of the data mining are known, whether those results are what the team expected or desired. Finally, the data mining should be accompanied by as much knowledge about both the data and the business question as possible. Blind searches for patterns in large databases are rarely fruitful and will waste valuable analytic resources.

In Practice 21.1 Data preparation

MANAGER I've started looking at our (Paralyzed Veterans of America) database to try to understand who our current donors are. Just looking at the variable *AGE*, can we use it to predict who is most likely to give to our next campaign? What corrections or adjustments would be appropriate?

ANALYST Here are a histogram and boxplot for the variable *AGE* for all 94,649 records of the PVA dataset.



	Quantiles	
100.0%	maximum	98.000
99.5%		95.000
97.5%		90.000
90.0%		83.000
75.0%	quartile	75.000
50.0%	median	62.000
25.0%	quartile	48.000
10.0%		39.000
2.5%		31.000
0.5%		26.000
0.0%	minimum	1.000
Mean		61.598019
Std Dev		16.666865
Std Err Mean		0.0624749
upper 95% Mean		61.720469
lower 95% Mean		61.475568
N		71170

There is a group of cases with ages below 20 that should immediately draw our attention. Are people that young likely to be donors? A closer examination reveals that there are even a handful of cases whose ages are below 15, of whom 17 are younger than 5 years old and 9 are 1 year old. There are also 23,479 missing values for the variable *AGE*. From what I know about our database, ages below 20 (and certainly those below 10) are likely to be errors. I could go back and investigate those individually, but I think given how few there are, simply deleting the values below 21 is probably the wisest course of action for now.⁸

⁸Further analysis revealed that age was calculated from date of birth by a computer algorithm, but only the last two digits of the date of birth year were used. So, some people (perhaps no longer living) who were in the database had ages 100 years too young.

Just Checking

The bank vice president from the previous Just Checking wants to know some things about the data and the data mining process. Respond briefly to her assumptions and questions to a data mining consultant.

6. To me, it seems best and most efficient to build the team from one department, like my information technology group, rather than make a cross-disciplinary team for the project. What do you think?
7. I assume that once we identify the data sources we'll use, we can get right into modeling the issue, correct?
8. These data mining techniques are so powerful that I'm virtually guaranteed the predictions will be accurate, right?
9. Rather than look at one small problem, I think it might be a good idea to just explore the database and see what's in there. What do you think?
10. Once we test the model and it seems to work well, we'll be able to use it for the next few years at least.

valerie.albury@student.ashford.edu

Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

21.3 Data Mining Algorithms: A Sample

Data mining models are often referred to as **algorithms**, a term that describes a sequence of programming steps with a specific purpose. You may hear a method (even one like linear regression) referred to as a model, an algorithm, a tool, or generically as a method. The terms seem to be used interchangeably. Technically, the algorithm is the set of rules that the computer uses to compute the value of the model. The algorithm then uses the data to find the optimal values of the parameters, to fit the data most closely. As with linear regression, by most closely we typically mean minimizing the sum of squares of the differences between the data and the predicted values.

This section touches on only a few of the models used in data mining. Some of the most common models used for prediction are decision trees, neural networks and random forests (discussed below), and the regression methods discussed in **Chapters 18** and **19**. It's a lively research field and new algorithms are appearing all the time.

Model Selection: Training the Model

Data mining algorithms are powerful. That's great, but with that power comes the ability to fit every pattern in the data, including noise. As mentioned in **Chapter 19**, to avoid overfitting, the data are typically split randomly into training and test data. The training data are used to fit (or train) the model. Then the model predicts the response of the test data. The test set has not been used to choose the parameters or the variables in the model. We use this cross-validation to see how well the model predicts on data that it has never seen.

What Is a Data Mining Algorithm?

"A data mining algorithm is a well-defined" procedure that takes data as input and produces output in the form of models or patterns."

Principles of Data Mining, Hand, Mannila, and Smyth

**Well-defined* means it can be encoded into a set of commands into a computer.

A more thorough way to evaluate models is to split the dataset randomly into training and test repeatedly, cross-validating more than once. Typically, 90% of the data are used to fit and 10% used to evaluate in each iteration. When repeated test sets are used, they are often called **hold outs**. The model fitting is repeated on all the training sets, computing a performance measure on each hold out. The average performance on all the hold out sets will be the final measure of how well the model fits.

What do we mean by predicting well? For a continuous predictor, we might use the sum of squares of the deviations on the test set, or the R^2 on the test set defined as:

$$R^2 = 1 - \text{SSE} / \text{SST}.$$

This is the same definition we used for multiple regression, but for the new data, R^2 can actually go negative if the predictions for the test are farther away from the actual response than they are from their mean.

For categorical data there are a variety of performance measures all based on the number of misclassified cases in the test set. The simplest is to count up all the misclassifications. However, it's often more serious to misclassify one of the levels of the response than another, so weighting the misclassifications is also popular.

For example, it is far worse to miss finding cancer than it is to tell someone that he or she may have cancer, and then to discover that it is not there. Similarly, it may well be much worse to permit credit card fraud than to inconvenience people with a false alarm (when the company says that a transaction may be fraud, but, in fact, it was not).

Fortunately, many algorithms do some sort of cross-validation automatically and report back the best model. We'll see how this works in practice in the following examples.

Tree Models

Trees are simple models used for both classification and regression problems. To illustrate how a tree works, imagine that we want to predict whether someone will default on a mortgage. To build the tree, we use past data from a group of customers on whom we have measured some predictor variables and for whom we know whether they defaulted or not. (So, this is a supervised classification problem.) For this simplified example, let's assume we'll predict *Default* only from the following variables:

- *Age* (years)
- *Household Income* (\$)
- *On the Job* (years)
- *Debt* (\$)
- *Homeowner* (Yes/No)

Perhaps the easiest way to understand what a tree is doing is to look at its output. **Figure 21.2**  shows a tree fit to our hypothetical default example. At each level of the tree, data are split into two parts based on the values of a predictor variable. The first split is on *Household Income* at the value \$40,000. Those whose incomes are > \$40,000 (indicated by "Yes") are on the right. Following down the tree, the tree next splits that group on household *Debt*. For people who have both *Income* > \$40,000 and *Debt* > \$10,000, the tree estimates a 0.05 probability of *Default*. For those with *Income* > \$40,000 but *Debt* not > \$10,000, the probability is only 0.01. Going back up to the top, for the left branch of the first split, the tree splits on years *On the Job* (and not *Debt*). For the customers with *Income* not > \$40,000 and more than 5 years *On the Job*, the *Default* probability is estimated to be 0.06, and for those with *Income* not > \$40,000 and 5 years or less *On the Job*, the probability is highest—at 0.11.

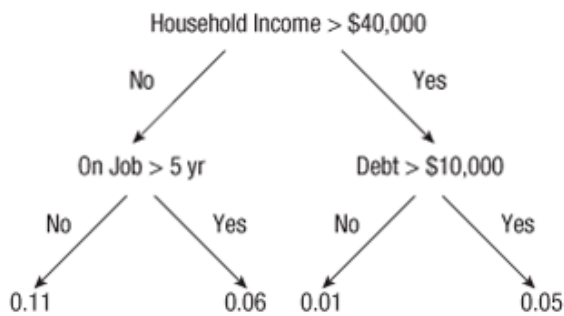


Figure 21.2

Part of a tree model on a hypothetical examination of mortgage default. The tree selects *Household Income* as the most important variable to split on and selects \$40,000 as the split point. For those customers whose *Income* is more than \$40,000, *Debt* is the next most important variable, while for those whose income is less than \$40,000, how long they've been *On* their current *Job* is more important. For each of the four groups, the estimated probability of *Default* is shown at the bottom at the terminal nodes.

One of the benefits of the tree is how simple it is to interpret. But how does the tree choose the variables and the values to split on? The answer might be surprising. The tree essentially does it by brute force, trying every possible predictor variable and split point to see which combination produces the largest *difference* in default rates.⁹

⁹This is a simplification. Tree algorithms have shortcuts so they don't have to try every possible combination. In fact, they are computationally very efficient. Readers who want to know the details can read more advanced books on data mining or decision trees.

The tree starts by considering all the data, and tries splitting the data into two groups based on each predictor variable, looking at the impact that makes on the response variable. It searches for that combination of variable and split that produces the largest difference in the response.¹⁰ For a nonordered categorical predictor, the model considers every possible way of splitting the categories into two groups. For an ordered categorical variable, the modeler may choose to respect the order and restrict the possible splits (see sidebar). For a quantitative variable, the possible splits are at every possible value the variable can take (other than the min and the max).

¹⁰There are several different criteria for measuring the difference in the response on the two sides of the split.

After the algorithm finds the first split, it continues searching again on the two resulting groups, finding the next best variable and split point (possibly using the same variable as the previous split again). It continues in this way until one of several criteria is met (for example, number of customers is too small, not a large enough difference in rates is found) and then stops at what are called **terminal nodes** ⓘ, where the model produces a prediction. The predictions at the terminal nodes are simply the average (if the response is quantitative) or the proportions of each category (for a classification problem) of the cases at that node.

Splitting a Categorical Variable

If our categorical predictor is *Region* (East, North, South, West), there are 7 possible splits:

East—North, South, West
North—East, South, West
South—East, North, West
West—East, North, South
East, North—South, West
East, South—North, West
East, West—North, South

If the categorical variable is ordered and the modeler wants to keep that order, then the splits are restricted. For example, if the variable is *Age* (*Child*, *Young Adult*, *Adult*, *Senior*), then the only possible splits are:

Child—*Young Adult*, *Adult*, *Senior*
Child, *Young Adult*—*Adult*, *Senior*
Child, *Young Adult*, *Adult*—*Senior*

Tree models are very easy to implement and, in principle, are easy to interpret. They have the advantage of showing their logic clearly and are easy to explain to people who don't have a deep background in statistics. Even when they aren't used as a final model in a data mining project, they can be very useful for selecting a smaller subset of variables on which to do further analysis.

Unlike many other data mining algorithms, a tree can handle a great number of potential predictor variables, both categorical and quantitative. This makes it a great model for starting a data mining project. It also handles outliers and high influence values in the predictors well because it simply splits the predictor into two groups. Once the tree model is fit, the analyst can then choose whether to use this model as a final model for predicting the response, or simply to use the variables suggested by the tree as inputs to other models that are more computationally intensive and less able to handle very large numbers of predictors.

In Practice 21.2 Gaining insights from a tree model

MANAGER Can you help me understand who responded to our most recent solicitation?

ANALYST I think that a tree model would be appropriate here. The tree model starts at the root node with all 94,649 donors (listed under Count in the top box of the figure).

The first split of the tree is on the variable *RFA_4*, a variable that summarizes past giving. This is one of the RFM variables that marketing created to summarize past giving. The fact that the tree chose it first is a great credit to their efforts! The algorithm identifies this as the best single predictor of *TARGET_B*, splitting the levels of that variable into two groups, with 29,050 customers split to the left and the remaining 65,599 customers to the right. The variable *RFA_4* comprises codes that contain information on how recently the last gift solicitation was received, how frequently the donor has given, and how large the last gift was. (Examples of these codes are A3C, S4B, and so on, seen in parentheses after the variable name in the output display.) The percentages next to the levels 0 and 1 indicate the proportion of those who donated or not (*TARGET_B* 1 or 0, respectively) in each node. Notice that

7.34% of those in the left node contributed, which was almost twice the 4.06% in the right node who contributed.

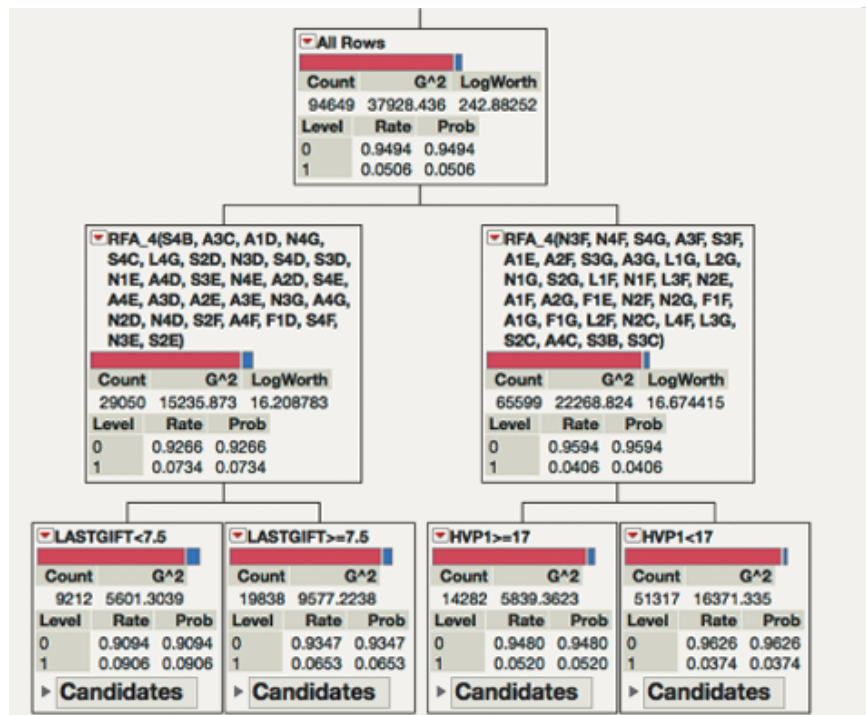


Figure 21.3

Part of a tree model run on the PVA data using JMP® software. (Source: Created with JMP® 13 Pro software. Copyright 2016, SAS Institute, Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute, Inc, Cary, NC.)

The tree continues to split each subpopulation. Notice that the variable chosen to split the left branch of the tree (*LASTGIFT*) is not the same as the variable chosen to split the right branch (*HVP1*). The terminal nodes are found in the bottom row. The terminal node at the bottom left of the figure shows that 9.06% of the 9212 potential donors in this subgroup donated, a substantial improvement over the average of all potential donors (5.06%) shown in the root node at the top. Even at this stage of the analysis, if directed our solicitations to names on our list that satisfied the conditions found in the path through the tree to this node, we might reduce our costs and increase our yield substantially.

It's interesting to see that of the 479 potential predictor variables, 2 of the top 3 variables selected by the decision tree algorithm involve past giving history rather than demographics. *RFA_4* is a summary of past giving and *LASTGIFT* measures the amount of the last gift. *HVP1* is a demographic variable, measuring the percentage of residents in the donor's ZIP code whose homes are worth at least \$200,000. By letting the tree grow further, I've increased the list of potential predictors to the top 20 (see [Figure 21.4](#)). Notice that most of the important variables involve past giving of the donors and a few key demographic variables about the donor's neighborhood. I suggest we look at the data only for those 20 variables and spend some time visualizing these variables and assessing data quality and dealing with missing data issues before proceeding with more model building.

Column Contributions				
Term	Number of Splits	G ²		Portion
RFA_4	1	423.738142		0.4194
CARDGIFT	2	80.107907		0.0793
LASTGIFT	2	78.4402851		0.0776
LASTDATE	3	74.1487712		0.0734
HVP1	1	58.1273009		0.0575
CARDPROM	1	32.3495992		0.0320
EC2	1	29.4524333		0.0291
MC3	1	25.6105524		0.0253
HUPA3	1	22.7727912		0.0225
VC2	1	22.5842906		0.0224
ADATE_22	1	18.5629611		0.0184
HHN3	1	17.7936628		0.0176
HU5	1	17.4192077		0.0172
MINRDATE	1	17.0423069		0.0169
HHAS3	1	16.7292576		0.0166
HHN1	1	16.6986151		0.0165
ETHC1	1	15.8670199		0.0157
INCOME	1	15.549749		0.0154
AGEC6	1	14.9778014		0.0148
AGE	1	9.93632478		0.0098

Figure 21.4

The 20 most important predictor variables from a large tree model. Notice that the top 4 account for nearly two thirds of the total variable importance and all involve past donor giving.

*How Big (or Deep) Should the Tree Be?

If we continue to split the data enough times, eventually each data point will be in its own cell and the tree will fit the data “perfectly,” a classic example of overfitting. Splitting only once is likely to underfit the data. So, what’s the right compromise? Although there are some theoretical criteria that attempt (like adjusted R²) to choose the right size, cross-validation is usually a good way, in practice, to answer the question. Here’s an example using 30% of the data as a hold out on the PVA data:

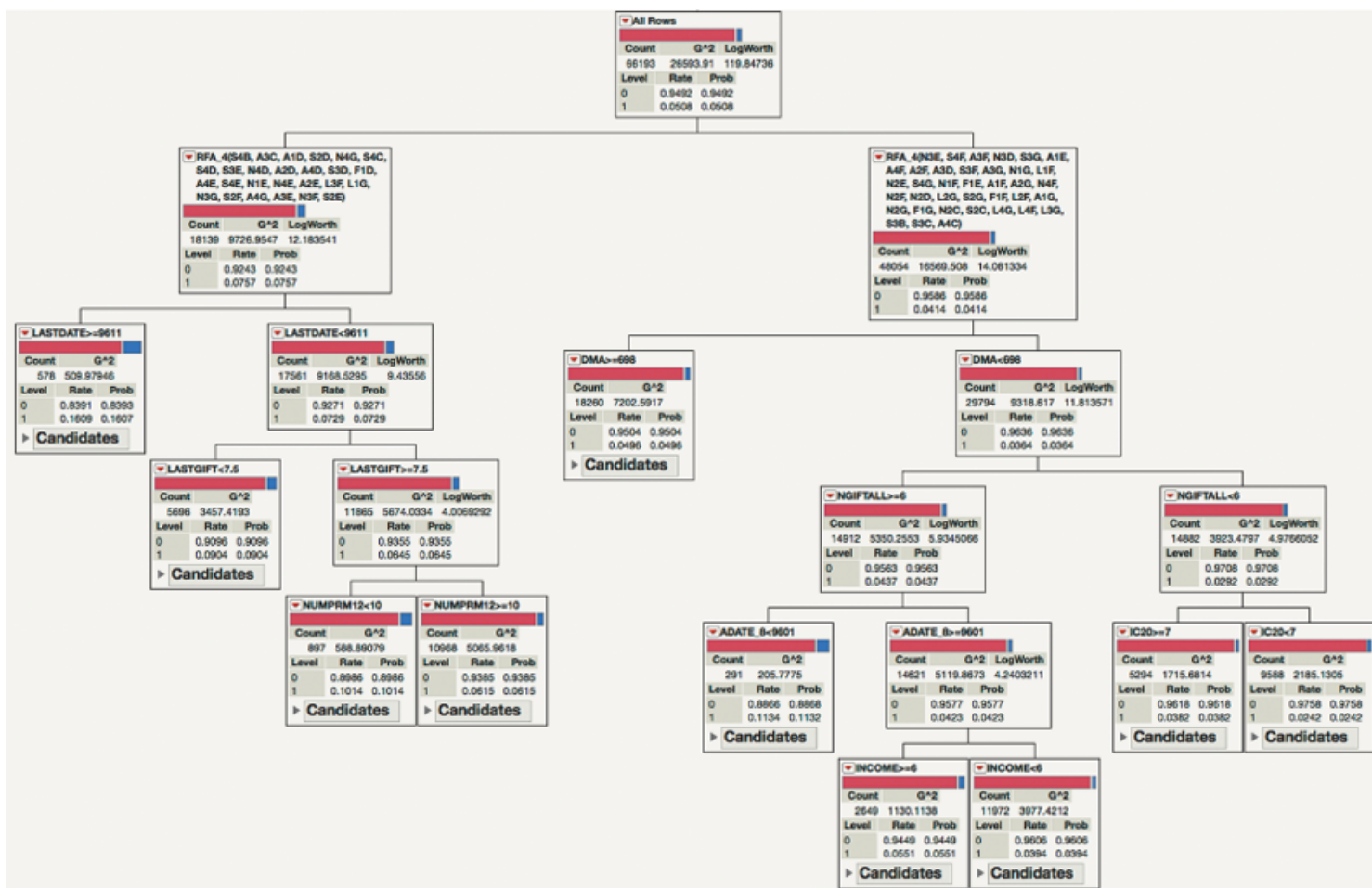


Figure 21.5

With a random 30% of the data as hold out, the tree algorithm found that this size tree, using 70% of the data as training to fit the model, had the best performance on the remaining 30% test set. The criterion used was based on the number of misclassifications—predicting that someone would respond to the solicitation when they didn't and vice versa. (Source: Created with JMP® 13 Pro software. Copyright 2016, SAS Institute, Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute, Inc, Cary, NC.)

Exploring the tree by splitting by hand (as the analyst did in the [In Practice 21.2](#)) can be a useful way to explore the data, but some sort of cross-validation should always be used to avoid overfitting.

Just Checking

The vice president is interested in using a tree model to predict the likelihood that a customer will sign up for the IRA offer recently sent out by the bank. She has a few questions.

11. When we run the tree model, will it automatically find the best combination of variables to predict the sign-up probability?
12. Last year we used a logistic regression to predict a similar sign-up probability. Will the tree perform better?
13. We have a few customers with very high incomes that proved to be a problem for logistic regression because of their high impact. We had to take them out of the model and treat them separately. How will the tree deal with them?

Neural Networks

Another popular data mining tool is the multilayer perceptron or (artificial) **neural network** ⓘ. This algorithm sounds more impressive than it really is. Although it was inspired by models that tried to mimic the function of the brain, it's really just an automatic, flexible, non-linear regression model. The neural network, like multiple regression, models the response variable with a function of the predictor variables but, unlike multiple regression, the functions are not linear. They are more closely related to the smoothing functions you saw back in **Chapter 20** ⓘ. And unlike stepwise regression, the neural network retains all the predictors, even if some of them have little effect on the response.

Because the “network” is made up by adding together many small non-linear functions, the equation that predicts the response is uninterpretable and results in what's called a **black box** ⓘ model. That's a major disadvantage for many applications. In multiple regression, the fitted model shows how each predictor influences the response. In a tree, the splits show the role of each predictor as you move down the tree. Sometimes, however, a black box model is acceptable. For example, if the neural network can correctly distinguish the digits of a hand-written ZIP code on a random letter with high accuracy, the US Postal Service might not really care how it's doing it. But, if a neural network determined that you are not qualified for a student loan, you'd probably want to know why—something that would be hard to tell from a black box model!

The basic idea of a neural network is to first create new functions (called **nodes** ⓘ or **features** ⓘ) from the original predictors. These nodes are just linear combinations (weighted averages) of the original predictors. The weights used are similar to coefficients in a linear regression. The collection of these new functions is called the **hidden layer** ⓘ. The reason they are called hidden is that the response, y , is not used directly to estimate the weights. The fitting is done by a computationally intensive method, the most common of which is called **back propagation** ⓘ. The hidden functions are then transformed by an S-shaped curve (like the one you saw in logistic regression) and are given the colorful name **sigmoidal activation functions** ⓘ (which really just means S-shaped curve!). Finally, the response is fit by a regression on these transformed functions. If the response is quantitative, the regression is linear. If it's categorical, then another sigmoid is applied to model the probability of belonging to each class. The fact that the nodes are transformed using a non-linear function gives the model its flexibility—and the ability to approximate a wide variety of phenomena.

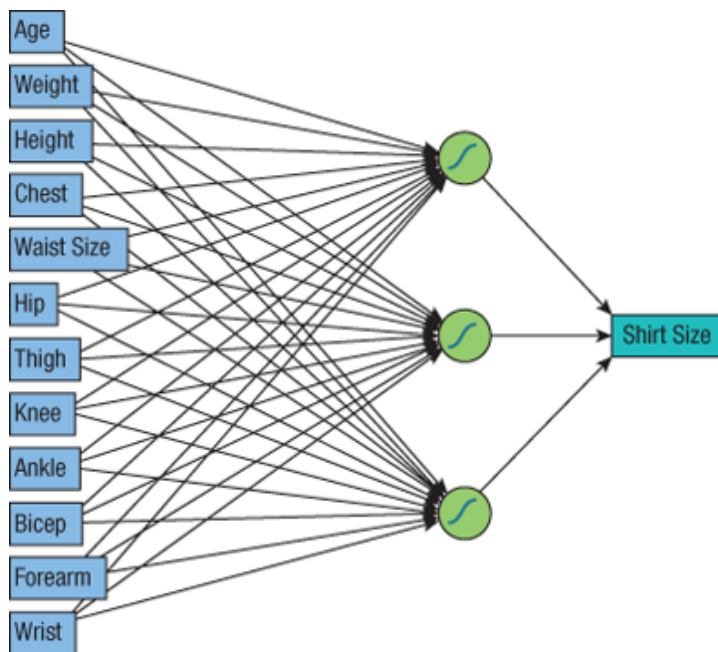


Figure 21.6

A neural network to predict *Shirt Size* from various other body measurements. The network has 3 nodes in the hidden layer. The S symbol indicates that the network is using sigmoidal activation functions.

A diagram of such a network with three nodes in the hidden layer is shown in [Figure 21.6](#). In this example, a shirt manufacturer wants to predict a man's shirt size (to the nearest half inch) from his other measurements (data in **Shirt sizes full**). Various methods are employed to ensure that the model doesn't overfit the data, and the final model is chosen by cross-validation.

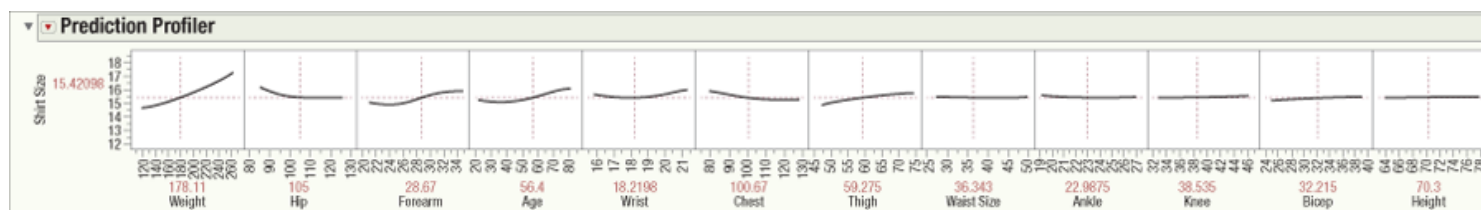


Figure 21.7

The profiler in JMP® shows the influence of each predictor variable on *Shirt Size*. The neural network model is nearly linear in *Weight*, and somewhat curved in the next few variables. The last few have relatively little effect on the response. (Source: Created with JMP® 13 Pro software. Copyright 2016, SAS Institute, Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute, Inc, Cary, NC.)

Although the model is, in general, uninterpretable, there are some implementations of neural networks that provide insight into the effect of the predictors by profiling the effect of each predictor using sensitivity curves and measuring the importance of the predictors. [Figure 21.7](#) shows the profile of predictor influences on *Shirt Size* (measured in inches), with a listing of variable importance as produced by JMP® ([Table 21.2](#)).

This model was fit using 75% of the data for training and 10% for testing (out of 250 cases). Another 15% was set aside as a "tuning set." We specified 3 hidden nodes and used the tuning set to select the best model. [Table 21.2](#) shows the performance on each set. On the training data, the model reports an R2 of 0.69, and the model actually fit the test set best, but this is probably an anomaly.

Table 21.2

The Summary Report shows numerical values for the relative importance of each predictor. *Weight* is by far the most important predictor of *Shirt Size*, accounting for more than half of the predictive power of the model. The model reports an R2 of 0.69 on the training set and higher on the test set. (This increase is probably just an anomaly, but it at least indicates that the model is not overfit.)

Summary Report			
Column	Main Effect	Total Effect	
Weight	0.432	0.557	
Hip	0.069	0.187	
Forearm	0.049	0.157	
Age	0.019	0.151	
Wrist	0.033	0.132	
Chest	0.011	0.086	
Thigh	0.029	0.069	
Waist Size	0.012	0.051	
Ankle	0.013	0.043	
Knee	0.008	0.03	
Bicep	0.008	0.022	
Height	0.003	0.01	

Crossvalidation			
Source	RSquare	RASE	Freq
Training Set	0.6911	0.48978	186
Validation Set	0.7641	0.50461	37
Test Set	0.8702	0.30148	25

When the R2 on the test set is much lower than the R2 on the training set, this usually indicates overfitting. When the performance is approximately the same, it doesn't necessarily indicate a powerful model, but at least it's evidence that the model is not overfitting. When the model actually performs best on the test set (data that it hasn't seen) it's probably just random noise. Models in general perform best on the data on which they are fit (or trained).

The language used to describe neural network models and the method of estimating their parameters is colorful and designed to make you think of the way the human brain works. Don't be fooled. The model is really just a flexible function fitter and the network doesn't "learn" any more (or less) than any other statistical model. If you applied these terms to multiple regression, the intercept and slopes would be called weights. The act of fitting the regression to a set of training data would be called training the network, or "learning." The resulting predictions would be called knowledge. A glossary of terms is given in [Table 21.3](#).

Table 21.3

A glossary of some of the most common terms used in neural networks.

Glossary	
Neural network	A non-linear regression model using transformed linear combinations of the predictors to predict the response
Weight	A coefficient in the neural network model
Back propagation	A computational method for estimating all the weights in the model

Hidden node	A linear combination of the original predictor variables
Sigmoidal activation function	An S-shaped curve used to transform the hidden “nodes”
Training and learning	Colorful names for the process of estimating the coefficients of the model from a set of training data
Knowledge	The output from the model—really just the predictions
Deep learning	A neural network with more than one hidden layer

In the 1980s some mathematical theorems were proved that showed that neural networks can approximate nearly any function (with enough data). They also showed that one hidden layer (with enough nodes) is all that is needed. In spite of this, some researchers have put a lot of effort into studying multi-layer networks. One of these research groups won an image recognition contest in 2012 with an algorithm that was twice as accurate as the runner-up. These multi-layer networks now go by the name **deep learning** ⓘ, continuing the trend in making fanciful names that sound like they are doing more than they are.

Just Checking

Our vice president is back with a couple of questions.

14. Because the neural network thinks like a human, it will be the best model that we can run, correct?
15. A customer who had a very high probability of taking our offer, as predicted by the neural network, didn't. Can the neural network tell us why?

Personal use
2020-11-23
valerie.albury@student.ashford.edu

Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

21.4 Models Built from Combining Other Models

Many other algorithms besides trees and neural networks are used by data miners, and new ones are being developed all the time. Typically, a data miner will build several different models, test them in various ways, and then simply select one to use based on a combination of the best performance and other criteria like interpretability. Or, the data miner might decide to average several models. For a classification problem, the “average” prediction for a case might actually be the “mode”—the class that’s predicted most often by all the models under consideration. For a regression problem, the prediction might be the average or a weighted average of the predictions of all the models. Averaging models provides protection against choosing the “wrong” model, but usually at a cost of not being able to interpret the resulting predictions. Finding the best ways to combine many different models is an active and exciting area of current research in data mining. We’ll discuss a few of these ideas in this section.

In recent years, research has focused on combining many versions of similar models. Imagine you’re a CEO or president who needs to make a decision. You assemble your advisors, listen to each, and then combine their opinions in some way to make a final choice. The most common methods of combining models are called **bagging** and **boosting**. Each of these methods creates “committees” of models. In bagging (short for **bootstrap aggregating**), you fit a model (say, a tree) to the training data. Then you take a bootstrap sample of the data (see [Chapter 11](#)) and fit another tree. Repeat this a large number of times (say, 1000). Now use the average prediction (or most frequent class) as the prediction for each case in the test set. Boosting is more technical, but one way to think about it for classification is to fit a small model (for example, a tree with just 2 or 3 splits) to the training data. Now, give more weight to the cases that were misclassified, and create another tree. Repeat this over and over, each time giving more weight to the misclassified cases. The later trees will fit the harder cases better (and the easier cases perhaps less well). Now average all the trees by their overall performance. In principle, you can use bagging and boosting for nearly any model, but commonly trees are used.

Random Forests

Perhaps the most common version of bagged trees is the **random forest**. The random forest creates trees on many bootstrapped samples of the training data and then averages them, but with one important wrinkle. Every time each tree looks for a split, it does so on only a small random sample of the predictors (often 10%). This seems counterintuitive. For example, suppose that only 10 or so of the nearly 500 predictors in the PVA data are important. If we restrict each split to only 50 randomly chosen predictors each time, there’s a good chance that none of those 10 will be in the sample and the tree will make a uninformative split. But it turns out that that doesn’t really hurt performance. In fact, it provides a richer set of trees that seem to find more signal in the data than simply bagging lots of trees.

Continuing our *Shirt Size* example, here is the output of a random forest of 100 trees fit to the training data ([Table 21.4](#)) and the profiler, showing the effect of the predictors ([Figure 21.8](#)).

Table 21.4

The Column Contributions show the relative importance of each predictor. The order is quite different than the order produced by the neural network, although both models claim to fit the test set fairly well.

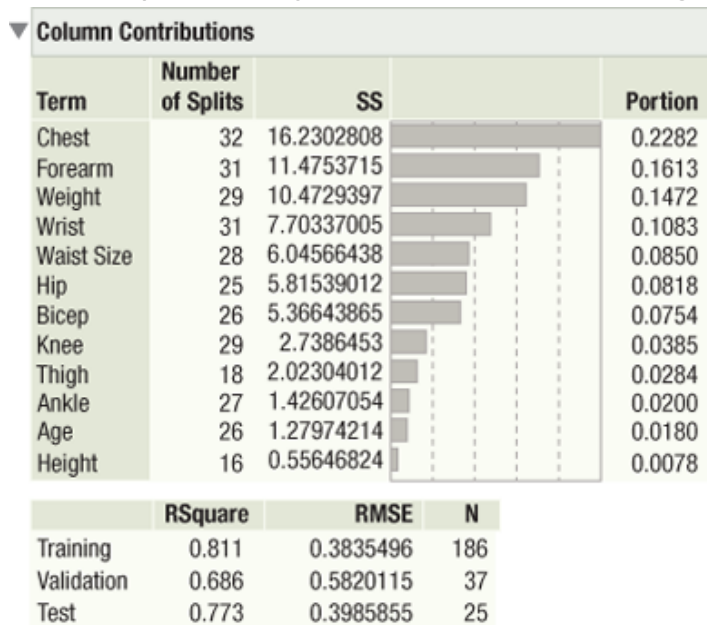


Figure 21.8

The profiler for the random forest shows a typical pattern for tree-based models, with sharp step-like increases for *Shirt Size* at various levels of the predictors. (Source: Created with JMP® 13 Pro software. Copyright 2016, SAS Institute, Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute, Inc, Cary, NC.)

The random forest fit an R^2 of 0.811 on the training set and then slightly lower (0.773) on the test set (Table 21.4). Although it is a somewhat automatic method, performance can often be improved by optimizing some of the parameters (number of trees fit, percentage of predictors chosen at each split, size of each tree, and so on). Notice also that the random forest used the predictors quite differently than the neural network did, placing more importance on *Chest* and *Forearm* size and less on *Waist Size*. It's important to remember that these models are sensitive to the training data that were selected as well as some random aspects of their fitting. Large differences between data mining models can provide insights into the phenomenon being studied, or they can be an indication that the models are fitting noise as well as signal. It is always wise to be cautious in putting too much faith in a model that is automatically fit, especially on a “found” set of data.

Boosted Trees

Trees can also be boosted, and the resulting ensemble (set) of trees is usually a more powerful predictor than a single tree. Of course, like any ensemble, it's much less interpretable, especially compared to the simple structure of a tree. The random forest creates an ensemble by fitting trees on different bootstrap samples of the dataset. The boosted tree does not resample the data, but fits a *sequence* of trees on the original sample. However, after the first tree, it gives more importance (weight) to the cases that were predicted badly. So the next tree focuses more on the hard-to-predict cases. At each stage, the misclassified cases are upweighted so that later trees focus on repeatedly misclassified cases. These later trees may not do as well overall on classifying the data, but they may perform better on the hard-to-predict cases. The final model averages the predictions from all the trees, with more weight given to the trees that do best overall. Boosting can be applied to a variety of models, and the resulting ensemble often outperforms a single model.



Figure 21.9

The profiler for the boosted tree shows the same typical pattern for tree-based models with sharp step-like increases for *Shirt Size* at various levels of the predictors. (Source: Created with JMP® 13 Pro software. Copyright 2016, SAS Institute, Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute, Inc, Cary, NC.)

The boosted tree on the *Shirt Size* data gave the summary in Table 21.5 and the profiler in Figure 21.9. The model produces the typical step functions of a tree, as one can see in the profiler. In contrast to the random forest, however, the importance it gives to the variables is similar to that given by the neural network. Its performance on the test set ($R^2=0.792$) falls between the two previous models.

Table 21.5

The Column Contributions show that the boosted tree agrees with the neural network that *Weight* is the most important predictor. Its performance is quite consistent with an R^2 on the test set of 0.792, close to the training set R^2 of 0.834.

Column Contributions				
Term	Number of Splits	SS		Portion
Weight	23	315.348009		0.4973
Wrist	19	79.3915553		0.1252
Waist Size	15	77.6304535		0.1224
Forearm	26	61.5933939		0.0971
Chest	16	47.27591		0.0746
Bicep	4	13.0290865		0.0205
Ankle	12	11.016775		0.0174
Thigh	9	7.89480506		0.0124
Age	10	6.37984947		0.0101
Hip	7	5.90129527		0.0093
Height	4	4.94421839		0.0078
Knee	5	3.73171268		0.0059

Overall Statistics			
	RSquare	RMSE	N
Training	0.834	0.3589126	186
Validation	0.717	0.5529333	37
Test	0.792	0.3814433	25

Just Checking

And . . .

16. Which of the 1000 trees that the Random Forest uses is the best? Can we look at it?
17. When we classified people into “Likely to buy” and “Not likely to buy,” using a Boosted Tree we got a 20% error rate on the training set, but a 50% error on the test set. What’s the likely cause of that?

21.5 Comparing Models

Usually there is no way of knowing which of many models will predict best on future data. Choosing a model (or models) requires judgment. If several models perform about equally well, one might be preferred on the basis of scientific or other domain knowledge. Performance measures for quantitative responses include R^2 on the test set (defined earlier) and the **RASE or RMSE (square root of averaged squared prediction error or root mean squared error)** on the test set, which is the same as the residual standard deviation that we saw in multiple regression (except that it's evaluated on the test set).

Measuring Test Set Performance for Quantitative Responses

$$R^2 = 1 - \text{SSE} / \text{SST}$$

$$\text{RASE} = \text{RMSE} = \sqrt{\text{SSE} / n}$$

For categorical responses, there is always a trade-off between Type I and Type II errors. A graphical display called an ROC curve¹¹ can often help compare models over a range of choices. This display shows the trade-off between Power and Type I error over a range of possible error rates.

11Originally called receiver operating characteristic (ROC) curves, they plot the true positive rate (TPR) against the false positive rate (FPR) for all values between 0 and 1. The TPR is equivalent to the Power (the probability of correctly classifying a positive) and the FRP is the probability of a Type I error (probability of misclassifying a negative).

These measures (in addition to business knowledge) can help guide researchers in choosing the “best” model. Sometimes, however, there is no reason to prefer one over another and the analyst might be just as happy to average the predictions from all of them. This average can be a simple arithmetic average of each prediction, or a weighted average, or the predictions from various models can even be used as inputs to another neural network, which will then optimize a function of them based on other training data. In the next Guided Example, we will display all the models we have just discussed for *Shirt Size* and compare them to a simple multiple regression.

Guided Example Model Comparisons

Let's return to the problem of estimating a man's shirt size (inches around the neck to the nearest half inch) from other body measurements.



Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

Define the problem—state the objective of the study.

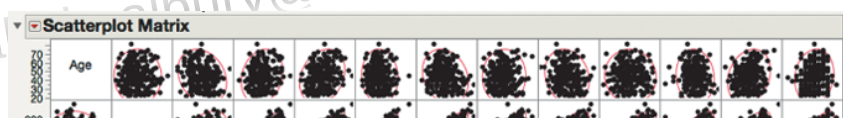
Identify the variables.

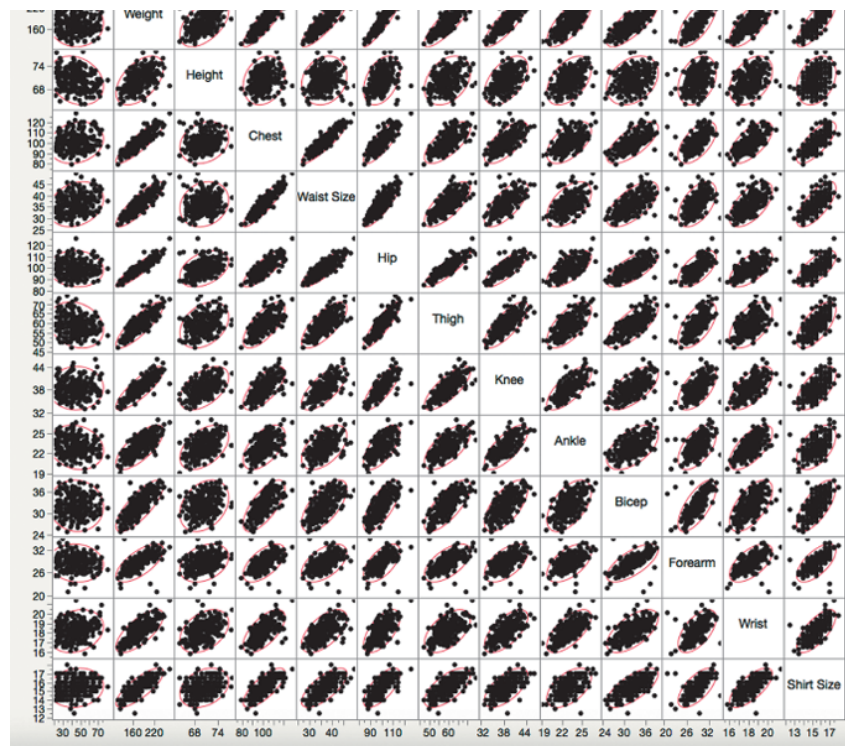
If reasonable, plot the variables and look for relationships among the variables, outliers and other interesting features.

If there are too many variables to display, running an exploratory model such as a tree can help select the most important variables, which can then be displayed.

We want to build and compare several models to predict *Shirt Size* from other body measurements. The available variables on men from a scientific study include:

Age	(yr)
Weight	(lb)
Height	(in.)
Chest	(cm)
Waist	(in.)
Hip	(cm)
Thigh	(cm)
Knee	(cm)
Ankle	(cm)
Bicep	(cm)
Forearm	(cm)
Wrist	(cm)
Shirt Size	(nearest half in.)





There are many strong, nearly linear relationships among the variables. In particular, most of the measurements (except Age) seem to have a positive linear relationship with *Shirt Size*, which makes sense. Men who are larger, heavier, and taller tend to have larger shirt sizes. There are no obvious outliers.

Mechanics Fit a stepwise multiple regression model, starting with all the predictors and their interactions.

Fit a tree, neural network, random forest, and boosted tree.

Here is the computer output for the stepwise multiple regression, starting with all 13 predictor variables and their interactions, and proceeding forward until no more candidates were nominated for exclusion.

Dependent variable is: Shirt Size

R-squared=69.1%, R-squared (adjusted)=68.1%,

s=0.499 with 186-7=179 degrees of freedom

Term	Estimate	Std Error	t-Ratio	Prob > qtq
Intercept	13.34017106	2.02415021	6.59	<0.0001
Weight	0.032632385	0.005077533	6.43	<0.0001
Height	-0.04587073	0.017441901	-2.63	0.0093
Hip	-0.044188241	0.016614473	-2.66	0.0085
Ankle	-0.109697048	0.041553449	-2.64	0.009
Forearm	0.047588487	0.024155895	1.97	0.0504
Wrist	0.278018404	0.063206759	4.4	<0.0001

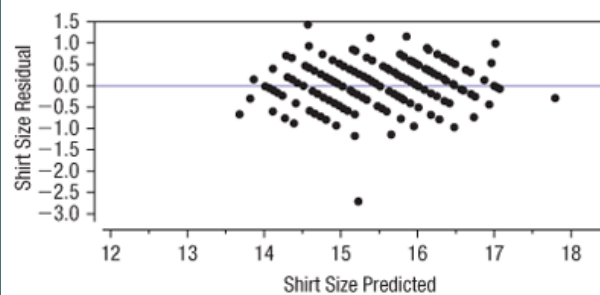
The estimated equation is:

$$\text{Shirt Size} = 13.34 + 0.033 \text{ Weight} - 0.046 \text{ Height} - 0.044 \text{ Hip} - 0.110 \text{ Ankle} + 0.048 \text{ Forearm} + 0.278 \text{ Wrist}$$

Effect Summary			
Source	LogWorth		PValue
Weight	8.942		0.00000
Wrist	4.729		0.00002
Hip	2.069		0.00853
Ankle	2.045		0.00903
Height	2.032		0.00928
Forearm	1.298		0.05037

Crossvalidation			
Source	RSquare	RASE	Freq
Training Set	0.6911	0.48978	186
Validation Set	0.7641	0.50461	37
Test Set	0.8702	0.30148	25

Variable importance shows that *Weight* and *Wrist* size are the most important predictors. The model gets an R2 of 0.691 on the training set and a surprisingly high 0.870 on the test set.



The residual plot shows one case with a low residual, but without large influence on the model. Other than the striped pattern because the shirt sizes are discrete (every half inch), the plot is patternless.

All Rows					
Count	186	LogWorth	47.467853	Difference	1.22994
Mean	15.408602				
Std Dev	0.8836235				

Weight<177.5			
Count	101	LogWorth	8.29755874
Mean	14.846535	Difference	0.66894
Std Dev	0.6393066		

Weight>=177.5			
Count	85	LogWorth	16.076471
Mean	16.076471	Difference	0.7487
Std Dev	0.6338933		

Chest<93.4			
Count	39	LogWorth	2.6406748
Mean	14.435897	Difference	0.70833
Std Dev	0.6706192		

Chest>=93.4			
Count	62	LogWorth	15.104839
Mean	15.104839	Difference	0.43571
Std Dev	0.462859		

Wrist<19.2			
Count	59	LogWorth	5.1189791
Mean	15.847458	Difference	0.60507
Std Dev	0.5512856		

Wrist>=19.2			
Count	26	LogWorth	16.596154
Mean	16.596154	Difference	0.64286
Std Dev	0.4902903		

Forearm<26.4 Count 15 Mean 14 Std Dev 0.46291 Candidates	Forearm>=26.4 Count 24 Mean 14.708333 Std Dev 0.6412736 Candidates	Forearm<28.7 Count 42 Mean 14.964286 Std Dev 0.4471162 Candidates	Forearm>=28.7 Count 20 Mean 15.4 Std Dev 0.3479262 Candidates	Weight<189.75 Count 23 Mean 15.478261 Std Dev 0.4122626 Candidates	Weight>=189.75 Count 36 Mean 16.083333 Std Dev 0.5 Candidates	Thigh<63.1 Count 12 Mean 16.25 Std Dev 0.398862 Candidates	Thigh>=63.1 Count 14 Mean 16.892857 Std Dev 0.3496466 Candidates
---	---	--	--	---	--	---	---

Column Contributions

Term	Number of Splits	SS	Portion
Weight	2	74.758113	0.7075
Chest	1	10.7129952	0.1014
Wrist	1	10.3259466	0.0977
Forearm	2	7.20353007	0.0682
Thigh	1	2.67032967	0.0253
Age	0	0	0.0000
Height	0	0	0.0000
Waist Size	0	0	0.0000
Hip	0	0	0.0000
Knee	0	0	0.0000
Ankle	0	0	0.0000
Bicep	0	0	0.0000

	RSquare	RMSE	Number of Splits	AICc
Training	0.732	0.4566308	186	7 255.264
Validation	0.656	0.6089036	37	
Test	0.699	0.4590413	25	

The tree fits less well on the test set than the multiple regression, with an R^2 of 0.699 on the test set. The most important variable is *Weight*, followed by *Chest* size. *Wrist*, second in the multiple regression, is third here.

Results for the neural network, random forest, and boosted tree are found in the previous section. Here is a summary of all 5 models and their performance on the test set:

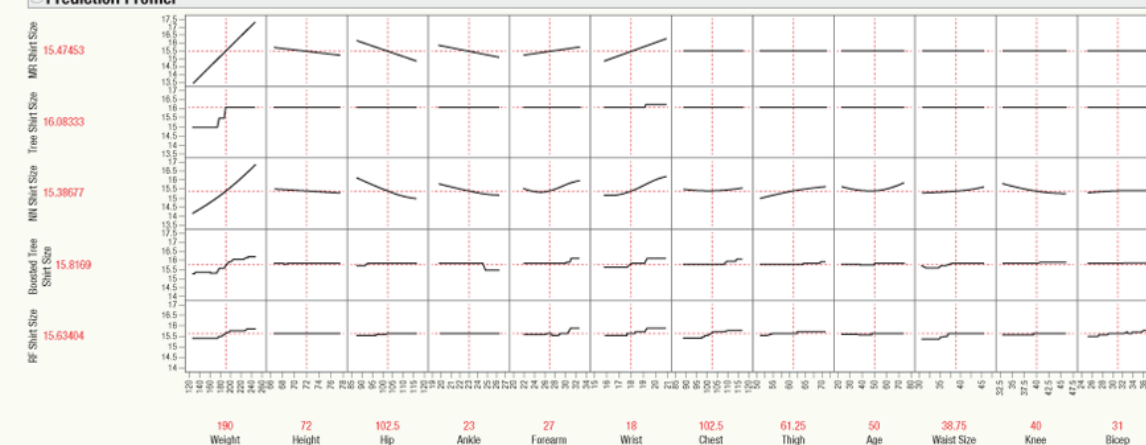
Model Comparison Test=2

Measures of Fit for Shirt Size

Predictor	Creator	2.4.6.8	RSquare	RASE	AAE	Freq
MR Shirt Size	Fit Least Squares		0.8702	0.3015	0.2369	25
Tree Shirt Size	Partition		0.6990	0.4590	0.3492	25
NN Shirt Size	Neural		0.8224	0.3526	0.2740	25
Boosted Tree Shirt Size	Boosted Tree		0.7921	0.3814	0.2811	25
RF Shirt Size	Bootstrap Forest		0.7933	0.3804	0.2973	25

Profiler Test=2

Prediction Profiler



Source: Created with JMP® 13 Pro software. Copyright 2016, SAS Institute, Cary, NC, USA. All Rights Reserved. Reproduced with permission of SAS Institute, Inc, Cary, NC.

Summary and
present your

Memo

conclusions.

Summarize your results and state any limitations of your model in the context of your original objectives.

Re: Models for shirt size prediction

Five models for *shirt size* were fit to a training set and predicted on a small test set of 25 randomly chosen cases. The models included multiple regression, a decision tree, a neural network, a random forest, and a boosted tree.

The models were similar in finding that *Weight* and *Wrist* size are important predictors. *Chest*, *Hip*, and *Forearm* size also appeared important in some models to a lesser degree. *Weight* and *Waist* size are positively related to *Shirt Size*, but because of collinearity with those variables in the model, *Hip* size is negatively related. Most of the predictors are roughly linear with *Shirt Size*, which helps to explain why a multiple linear regression was competitive with more flexible models.

All of the models gave R^2 values on the test set between 0.70 and 0.87, with the multiple regression highest and the tree lowest. The RASE averaged about 0.38 inch, which means that, roughly, we can predict a typical men's shirt size to within about 0.75 inch (2×0.38), which is about a size and a half. This may not be accurate enough for remote fitting. We recommend that the man try on the shirt for best fit.

Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

21.6 Summary

There are many similarities between the modeling process of data mining and the basic approach to modeling that you've been learning throughout this text. What makes data mining different really has to do with the large number of algorithms and types of models available to the data miner and with the size and complexity of the datasets. But most of what makes a data mining project successful would make any statistical analysis successful. The same principles of understanding and exploring variables and their relationships are key to both processes. A famous statistician, Jerry Friedman, was once asked if there was a difference between statistics and data mining. Before answering, he asked if the questioner wanted the long answer or the short answer. Because the response was "the short answer," Jerry said simply, "No." We never got to hear the long answer, but we suspect it may have contained some of the differences discussed in this chapter.

Having a good set of statistical and data analysis tools is a great beginning to becoming a successful data miner. Being willing to learn new techniques, whether they come from statistics, computer science, machine learning, or other disciplines, is essential. Learning to work with other people whose skills complement yours not only will make the task more pleasant but also is key to a successful data mining project. The need to understand the information contained in large databases will increase in the years ahead, so it will continue to be important to know about this rapidly growing field.

What Can Go Wrong?

- **Be sure that the question to be answered is specific.** Make sure that the business question to be addressed is specific enough so that a model can help to answer it. A goal as vague as "improving the business" is not likely to lead to a successful data mining project.
- **Be sure that the data have the potential to answer the question.** Check the variables to see whether a model can reasonably be built to predict the response. For example, if you want to know which type of customers are going to a particular web page, make sure that data are being collected that link the web page to the customer visiting the site.
- **Beware of overfitting to the data.** Because data mining tools are powerful and because the datasets used to train them are usually large, it is easy to think you're fitting the data well. Make sure you validate the model on a test set—a dataset not used to fit the model.
- **Make sure that the data are ready to use in the data mining model.** Typically, data warehouses contain data from several different sources. It's important to confirm that variables with the same name actually measure the same thing in two different databases. Missing values, incorrect entries, and different time scales are all challenges to be overcome before the data can be used in the model-building phase.
- **Don't try it alone.** Data mining projects require a variety of skills and a lot of work. Assembling the right team of people to carry out the effort is crucial.

Ethics in Action

With U.S. consumers becoming more environmentally conscious, there has been an explosion of eco-friendly products on the market. One notable entry has been the gas-electric hybrid car.

A large nonprofit environmental group would like to target customers likely to purchase hybrid cars in the future with a message expressing the urgency to do so sooner rather than later. They understand that direct mailings are very effective in this regard, but staying true to their environmental concerns, they want to avoid doing a nontargeted mass mailing.

The executive team met to discuss the possibility of using data mining to help identify their target audience. The initial discussion revolved around data sources. Although they have several databases on demographics and transactional information for consumers who have purchased green products and donated to organizations that promote sustainability, someone suggested that they get data on contributions to political parties. After all, there was a Green Party, and Democrats tend to be more concerned about environmental issues than Republicans. Another member of the team was genuinely surprised that this was even possible. She wondered how ethical it is to use information about individuals that they may assume is being kept confidential.

- **Identify the ethical dilemma in this scenario.**
- **What are the undesirable consequences?**
- **Propose an ethical solution that considers the welfare of all stakeholders.**

Chapter 21 From Learning to Earning

Learning Objectives

Understand the uses and value of data mining in business.

Recognize data mining approaches.

- OLAP approaches answer specific questions about data in a database that may involve many variables.
- Predictive model approaches build complex models that attempt to predict a response.

Be aware that data preparation is essential to sound data mining.

- Data must be cleaned of errors and impossible or implausible values.
- Missing values may be a serious issue.
- Data preparation may take the majority of time and effort devoted to a data mining project.

Be able to articulate the goal of a data mining project and understand that data mining efforts differ according to the kind of dependent variable.

- When the goal is to predict a quantitative response, the problem is called a regression problem.
- When the goal is to classify individuals according to a categorical variable, the problem is called a classification problem.

Be aware of the variety of models and algorithms that are used to build predictive models.

- Tree models build interpretable models that select predictor variables and split points to best predict the response in either a regression or classification problem.
- Neural networks build black box models that use all the predictor variables for both regression and classification problems.
- Many other data mining algorithms such as random forests, boosting and ensemble methods that combine models are used as well, and new competitors are continually being developed.

Understand that successful data mining is a team effort that requires a variety of skills.

- Defining the problem and objective of interest is crucial for success.
- The CRISP data mining process provides a good outline for a data mining project.

Terms

Algorithm

A set of instructions used for calculation and data processing. The algorithm specifies how the model is built from the data.

Back propagation

A computational method for estimating the weights in the neural network model.

Bagging

Abbreviation for B(ootstrap) agg(egat)ing a model. For any model, take a bootstrap sample and refit the model. Repeat this many times and use the average of all the models as the prediction.

Big Data

The collection and analysis of datasets so large and complex that traditional methods typically brought to bear on the problem would be overwhelmed.

Black box

A model whose equation is either hidden or so complicated that it is impossible to interpret. Examples include neural networks and random forests.

Boosting

A process of taking a simple model, fitting it to a set of data, and then using the results of that model to build a new model, putting more emphasis on the values that were predicted poorly.

Business analytics (or analytics or predictive analytics)

The process of using statistical analysis and modeling to make predictions and drive business decisions.

Classification problem

A prediction problem that involves a categorical response variable. (See also regression problem.)

Cluster analysis

A task of grouping similar objects (rows) based on their values of a set of variables.

Data mining

A process that uses a variety of data analysis tools to discover patterns and relationships in data that are useful for making predictions.

Data preparation

The process of cleaning data and checking its accuracy prior to modeling. Data preparation includes investigating missing values, correcting wrong and inconsistent entries, and reconciling data definitions.

Data warehouse

A digital repository for several large databases.

Decision tree (data mining version)

A model that predicts either a categorical or quantitative response in which the branches represent variable splits and the terminal nodes provide the predicted value.

Deep learning

A term often used to describe a neural network with more than one hidden layer.

Demographic variable

A variable containing information about a customer's personal characteristics or the characteristics of the region in which the customer lives. Commonly used demographics include age, income, race, and education.

Feature (and feature creation)

A variable in a machine learning model. Feature creation implies that a new variable has been created.

Hidden node (hidden layer)

A linear combination of the original predictor variables used to predict the response (sometimes called a latent feature or variable). The ensemble of hidden nodes is called the hidden layer.

Hold out (or test) set

The dataset used in a supervised classification or regression problem not used to build the predictive model. This dataset is withheld (held out) from the model building stage of the process, and the model's predictions on the test set are used to evaluate the model performance.

Knowledge

The insights gained from the model and its predictions.

Learning

The process of estimating the parameters of the model.

Machine learning

The name given to a collection of models that are heavily computational and often black box models. Neural networks and random forests are common machine learning algorithms or models.

Metadata

Information about the data including when and where the data were collected.

Neural network

A model based on analogies with human brain processing that uses combinations of predictor variables and non-linear regression to predict either a categorical or quantitative response variable.

Node

The variables in a neural network model, including the variables in the "hidden" layer are referred to as nodes. The variables in the first hidden layer are linear combinations of the original variables.

Online analytical processing (OLAP)

An approach for providing answers to queries that typically involve many variables simultaneously.

Predictive model

A model that provides predictions for the response variable.

Random forest

A model built by bagging a set of trees, with the added restriction that only a small random subset of predictors are used at each split of each tree.

Regression problem

A prediction problem that has a quantitative response variable. (See classification problem.)

Sigmoidal activation function

An S-shaped curve used to transform the hidden “nodes.” The logistic function from logistic regression is one of the most common activation functions.

Supervised problem

A classification or regression problem in which the analyst is given a set of data for which the response is known to use to build a model.

Terminal node

The final leaves of a decision tree where the predictions for the response variable are found.

Test set

The dataset used in a supervised classification or regression problem *not used* to build the predictive model. The test set is withheld from the model-building stage of the process, and the model's predictions on the test set are used to evaluate the model performance.

Training

Fitting a model, usually from a training set.

Training set

The dataset used in a supervised classification or regression problem to build the predictive model.

Transactional data

Data describing an event involving a transaction, usually an exchange of money for goods or services.

Unsupervised problem

A problem, unlike classification or regression problems, with no response variable. The goal of an unsupervised problem is typically to put similar cases together into homogeneous groups or clusters.

Weight

A coefficient in the neural network model (a parameter of the model).

Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

Personal use only, do not reproduce.
2020-11-23
valerie.albury@student.ashford.edu

Chapter 21 Exercises

Section 21.1

1. Your manager is confused by Big Data. Explain what makes “big data” different than “data.”
2. Your manager learned all about databases, and frequently makes queries such as: “What fraction of our customers who bought a product in the last six months are female and live within 5 miles of the store?” He says that he is data mining. Is he right? Explain.

Section 21.2

3. Your manager has read a couple of articles about data mining and artificial intelligence. He wants to use data mining to find the best customers. What would you suggest to structure this problem?
4. Is any one portion of the CRISP-DM more important than the others? Why?

Section 21.3

5. What are the advantages and disadvantages of using tree vs. neural network models?
6. Your manager wants to use the total accurate classification rate (percent of all cases properly classified) as the metric to evaluate the division’s models. Is this a good idea? Why or why not?

Section 21.4

7. What are the pros and cons of combining multiple models to produce a prediction? Should we always combine models?
8. Your manager wants to just find and use “the best” model, but you have found that a combined model (boosting) is better. Explain why boosting might help, and why it may be better than trying to find “the best.”

Section 21.5

9. Your manager asked you to fit many models to predict which customers will buy this holiday season. The firm is going to pick only one. What criteria would you recommend that the firm use to evaluate the various models and to compare them?