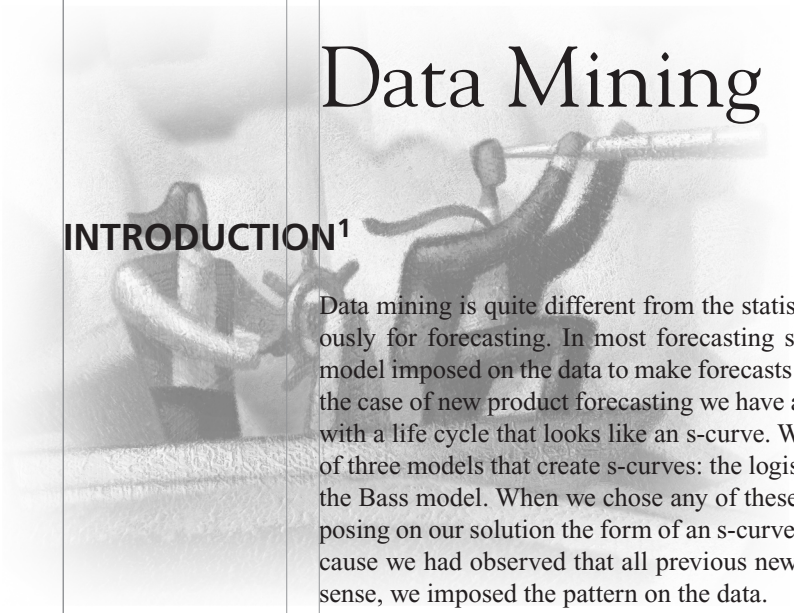
Chapter Nine

Data Mining



INTRODUCTION[1]

Data mining is quite different from the statistical techniques we have used previously for forecasting. In most forecasting situations you have encountered, the model imposed on the data to make forecasts has been chosen by the forecaster. In the case of new product forecasting we have assumed that new products "roll out" with a life cycle that looks like an s-curve. With this in mind we chose to use one of three models that create s-curves: the logistic model, the Gompertz model, and the Bass model. When we chose any of these three models we knew we were imposing on our solution the form of an s-curve, and we felt that was appropriate because we had observed that all previous new products followed this pattern. In a sense, we imposed the pattern on the data.

With data mining, the tables are turned. We don't know what pattern or family of patterns may fit a particular set of data or sometimes what it is we are trying to predict or explain. This should seem strange to a forecaster; it's not the method of attacking the data we have been pursuing throughout the text. To begin data mining we need a new mindset. We need to be open to finding relationships and patterns we never imagined existed in the data we are about to examine. To use data mining is to let the data tell us the story (rather than to impose a model on the data that we feel will replicate the actual patterns in the data). Peter Bruce points out, however, that most good data mining tasks have goals and circumscribed search parameters that help reduce the possibility of finding interesting patterns that are just artifacts of chance.

Data mining traditionally uses very large data sets, oftentimes far larger than the data sets we are used to using in business forecasting situations. The tools

[1] The authors would like to thank Professor Eamonn Keogh of the Department of Computer Science & Engineering at the University of California, Riverside, for many of the examples that appear in this chapter. We also want to thank Professors Galit Shmueli of the University of Maryland, Nitin Patel of MIT, and Peter Bruce of Statistics.com for the use of materials they created for their text *Data Mining for Business Intelligence* (John Wiley & Sons, 2007), ISBN 0-470-08485-5. The authors recommend visiting Professor Keogh's website for sample data sets and explanations of data mining techniques. We also recommend the *Data Mining for Business Intelligence* text for an in-depth discussion of all data mining techniques.

we use in data mining are also different than business forecasting tools; some of the statistical tools will be familiar but they are used in different ways than we have used them in previous chapters. The premise of data mining is that there is a great deal of information locked up in any database—it's up to us to use appropriate tools to unlock the secrets hidden within it. Business forecasting is explicit in the sense that we use specific models to estimate and forecast known patterns (e.g., seasonality, trend, cyclicality, etc.). Data mining, on the other hand, involves the extraction of implicit (perhaps unknown) intelligence or useful information from data. We need to be able to sift through large quantities of data to find patterns and regularities that we did not know existed beforehand. Some of what we find will be quite useless and uninteresting, perhaps only coincidences. But, from time to time, we will be able to find true gems in the mounds of data.

The objective of this chapter is to introduce a variety of data mining methods. Some of these are simple and meant only to introduce you to the basic concept of how data mining works. Others, however, are full-blown statistical methods commonly employed by data miners to exploit large databases. After completing this chapter you will understand what data mining techniques exist and appreciate their strengths; you will also understand how they are applied in practice. If you wish to experiment with your own data (or that provided on the CD that accompanies this text) we recommend the XLMiner© software.[2]

DATA MINING

A decade ago one of the most pressing problems for a forecaster was the lack of data collected intelligently by businesses. Forecasters were limited to few pieces of data and only limited observations on the data that existed. Today, however, we are overwhelmed with data. It is collected at grocery store checkout counters, while inventory moves through a warehouse, when users click a button on the World Wide Web, and every time a credit card is swiped. The rate of data collection is not abating; it seems to be increasing with no clear end in sight. The presence of large cheap storage devices means that it is easy to keep every piece of data produced. The pressing problem now is not the generation of the data, but the attempt to understand it.

The job of a data miner is to make sense of the available mounds of data by examining the data for patterns. The single most important reason for the recent interest in data mining is due to the large amounts of data now available for analysis.

[2] XLMiner© is an Excel add-in that works in much the same manner as ForecastX™. Both student and full versions of the software are available from Resample.com (http://www.resample.com/xlminer/). The authors also recommend *Data Mining for Business Intelligence* by Galit Shmueli, Nitin Patel, and Peter Bruce (John Wiley & Sons, 2007).

There is a need for business professionals to transform such data into useful information by "mining" it for the existence of patterns. You should not be surprised by the emphasis on patterns, this entire text has been about patterns of one sort or another. Indeed, men have looked for patterns in almost every endeavor undertaken by mankind. Early men looked for patterns in the night sky, for patterns in the movement of the stars and planets, and to predict the best times of the year to plant crops. Modern man still hunts for patterns in early election returns, in global temperature changes, and in sales data for new products. Over the last 25 years there has been a gradual evolution from data processing to what today we call data mining. In the 1960s businesses routinely collected data and processed it using database management techniques that allowed indexing, organization, and some query activity. *Online transaction processing (OLTP)* became routine and the rapid retrieval of stored data was made easier by more efficient storage devices and faster and more capable computing.

Database management advanced rapidly to include very sophisticated query systems. It became common not only in business situations but also in scientific inquiry. Databases began to grow at previously unheard-of rates and for even routine activities. It has been estimated recently that the amount of data in all the world's databases doubles in less than every two years. That flood of data would seem to call for analysis in order to make sense of the patterns locked within. Firms now routinely have what are called data warehouses and data marts. *Data warehouse* is the term used to describe a firm's main repository of historical data; it is the *memory* of the firm, its collective information on every relevant aspect of what has happened in the past. A *data mart*, on the other hand is a special version of a data warehouse. Data marts are a subset of data warehouses and routinely hold information that is specialized and has been grouped or chosen specifically to help businesses make better decision on future actions. The first organized use of such large databases has come to be called *online analytical processing (OLAP)*. OLAP is a set of analysis techniques that provides aggregation, consolidation, reporting, and summarization of data. It could be thought of as the direct precursor to what we now refer to as data mining. Much of the data that is collected by any organization becomes simply a historical artifact that is rarely referenced and even more rarely analyzed for knowledge. OLAP procedures began to change all that as data was summarized and viewed from different angles.

Data mining, on the other hand, concerns analyzing databases, data warehouses, and data marts that already exist, for the purpose of solving some problem or answering some pressing question. *Data mining* is the *extraction of useful information from large databases*. It is about the extraction of knowledge or information from large amounts of data.[3] Data mining has come to be referenced by a

[3] D. Hand, H. Mannila, P. Smyth, *Principles of Data Mining*, (Cambridge, MA: MIT Press, 2001), ISBN 0-262-08290-X.

few similar terms; in most cases they are all much the same set of techniques referred to as data mining in this text:

- Exploratory data analysis
- Business intelligence
- Data driven discovery
- Deductive learning
- Discovery science
- Knowledge discovery in databases (KDD)

Data mining is quite separate from database management. Eamonn Keogh points out that in database management queries are well defined; we even have a language to write these queries (structured query language or SQL, pronounced as "sequel"). A query in database management might take the form of "Find all the customers in South Bend," or "Find all the customers that have missed a recent payment." Data mining uses different queries; they tend to be less structured and are sometimes quite vague. For example: "Find all the customers that are likely to miss a future payment," or "Group all the customers with similar buying habits." In one sense, data mining is like business forecasting in that we are looking forward in an attempt to obtain better information about future likely events.

Companies may be data rich but are often information poor; data mining is a set of tools and techniques that can aid firms in making sense of the mountains of data they already have available. These databases may be about customer profiles and the choices those customers have made in the past. There are likely patterns of behavior exhibited, but the sheer amount of the data will mask the underlying patterns. Some patterns may be interesting but quite useless to a firm in making future decisions, but some patterns may be predictive in ways that could be very useful. For example, if you *know* which of your customers are likely to switch suppliers in the near future, you may be able to prevent them from jumping ship and going with a competitor. It's always less costly to keep existing customers than to enlist new ones. Likewise, if you were to *know* which of your customers were likely to default on their loans you might be able to take preventive measures to forestall the defaults, or you might be less likely to loan to such individuals. Finally, if you *know* the characteristics of potential customers that are likely to purchase your product, you might be able to direct your advertising and promotional efforts better than if you were to blanket the market with advertising and promotions. A well-targeted approach is usually better than a "shotgun" approach. The key lies in knowing where to aim.

What types of patterns would we find useful to uncover with data mining? The answer is quite different from the patterns we expected to find in data with business forecasting methods such as Winters' exponential smoothing. When we applied the Winters' model to time-series data we were looking for specific patterns that we knew had existed in many previously examined data sets (e.g., trend and seasonality). The patterns we might find with data mining techniques are usually unknown to us at the beginning of the process. We may find descriptive patterns in

Comments from the Field 1

MARKETING AND DATA MINING

Marketers have always tried to understand the customer.

A.C. Nielsen created a program called Spotlight as a data mining tool. The tool is for use in analyzing point-of-sale data; this data would include information about who made the purchase, what was purchased, the price paid for the item, the data and time of the purchase, and so on. The Spotlight system extracts information from point-of-sale databases, creates formatted reports that explain changes in a product's market share caused by promotional programs, shift among product segments (such as sales shifting from whole milk to soy milk), and changes in distribution and price. Spotlight can also be used to report on competing products. The Spotlight program looks for common and unique behavior patterns. Spotlight was designed to enable users to locate and account for volume and share changes for given brands. It won an award for outstanding artificial intelligence application from the American Association for Artificial Intelligence.

Spotlight became the most widely distributed data mining application in the industry for packaged-goods.

Source: This information is drawn from Byte.Com's archive of "The Data Gold Rush" that appeared in the October 1995 issue of *Byte* magazine.

our data; these tell us only the general properties of the database. We may also find predictive patterns in the data; these allow us to make forecasts or predictions in much the same manner as we have been seeing in the preceding chapters.

THE TOOLS OF DATA MINING

Shmueli, Patel, and Bruce use a taxonomy of data mining tools that is useful for seeing the big picture. There are basically four categories of data mining tools or techniques:

1. Prediction
2. Classification
3. Clustering
4. Association

Prediction tools are most like the methods we have covered in previous chapters; they attempt to predict the value of a numeric variable (almost always a continuous rather than a categorical variable). The term *classification* is used when we are predicting the particular category of an observation when the variable is a categorical variable. We might, for example, be attempting to predict the amount of a consumer expenditure (a continuous variable) in a particular circumstance or the amount that an individual might contribute yearly to a particular cause (also a continuous variable). The variable we are attempting to predict in each of these instances is a continuous variable, but the variable to be predicted might also be a categorical variable. For example, we might wish to predict

whether an individual will contribute to a particular cause or whether someone will make a certain purchase this year. Prediction then involves both categories of variables: continuous and categorical.

Classification tools are the most commonly used methods in data mining. They attempt to distinguish different classes of objects or actions. For instance, a particular credit card transaction may be either normal or fraudulent. Its correct classification in a timely manner could save a business a considerable amount of money. In another instance you may wish to know which characteristic of your advertising on a particular product is most important to consumers. Is it price? Or, could it be the description of the quality and reliability of the item? Perhaps it is the compatibility of the item with others the potential purchaser already owns. Classification tools may tell you the answer for each of many products you sell, thus allowing you to make the best use of your advertising expenditures by providing consumers with the information they find most relevant in making purchasing decisions.

Clustering analysis tools analyze objects viewed as a class. The classes of the objects are not input by the user, it is the function of the clustering technique to define and attach the class labels. This is a powerful set of tools used to group items that naturally fall together. Whether the clusters unearthed by the techniques are useful to the business manager is subjective. Some clusters may be interesting but not useful in a business setting, while others can be quite informative and able to be exploited to advantage.

Association rules discovery is sometimes called *affinity analysis*. If you have been handed coupons at a grocery store checkout counter your purchasing patterns have probably been subjected to association rules discovery. Netflix will recommend movies you might like based upon movies you have watched and rated in the past—this is an example of association rules discovery.

Au: We have removed the ascent umlaut from the character "i" as per editing in revised chapters 1 & 2. (GLOBAL)

In this chapter we will examine four techniques from the most used data mining category: classification. Specifically we will examine:

1. K-Nearest Neighbor
2. Naive Bayes
3. Classification/regression Trees
4. Logistic regression (logit analysis)

Business Forecasting and Data Mining

In business forecasting we have been seeking verification of previously held hypotheses. That is, we *knew* which patterns existed in the time-series data we tried to forecast and we applied appropriate statistical models to accurately estimate those patterns. When an electric power company looks at electric load demand, for instance, it expects that past patterns, such as trend, seasonality, and cyclicality, will replicate themselves in the future. Thus, the firm might reasonably use time-series decomposition as a model to forecast future electric usage. Data mining, on the other hand, seeks the discovery of new knowledge from the data. It does not seek to merely verify the previously set hypotheses regarding

the types of patterns in the data but attempts to discover new facts or rules from the data itself.

Mori and Kosemura[4] have outlined two ways electric demand is forecasted in Japan that exemplify the differences between data mining and standard business forecasting. The first method for forecasting load involves standard business forecasting. ARIMA models are sometimes used because of their ability to match those patterns commonly found in time-series data. However, other time-series models such as multiple regression are more frequently used because of their ability to take into account local weather conditions as well as past patterns of usage exhibited by individuals and businesses. Multiple regression is a popular and useful technique much used in actual practice in the electric power industry. Data mining tools are beginning to be used for load forecasting, however, because they are able to discover useful knowledge and rules that are hidden among large bodies of electric load data. The particular data mining model that Mori and Kosemura find useful for electric load forecasting is the regression tree; because the data features are represented in regression tree models as visualizations, if-then rules can be created and causal relationships can be acquired intuitively. This intuitive acquisition of rules and associations is a hallmark of data mining and sets it apart methodologically from the standard set of business forecasting tools.

The terminology we use in data mining will be a bit different than that used in business forecasting models; while the terms are different, their meanings are quite similar.

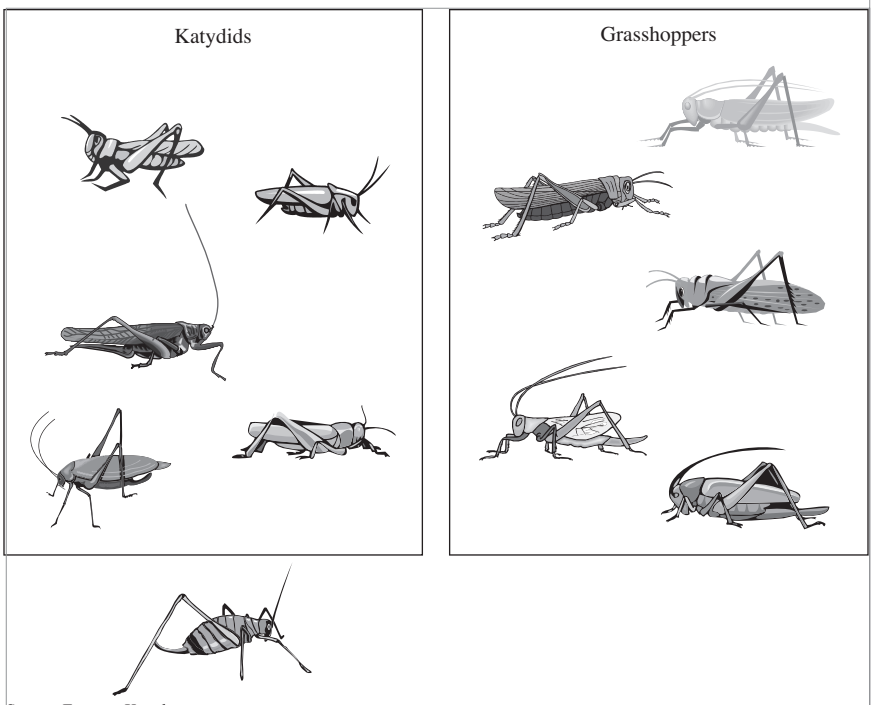
Data Mining	Statistical Terminology
Output variable = Target variable	Dependent variable
Algorithm	Forecasting model
Attribute = Feature	Explanatory variable
Record	Observation
Score	Forecast

Source: Eamonn Keogh.

A DATA MINING EXAMPLE: k-NEAREST-NEIGHBOR

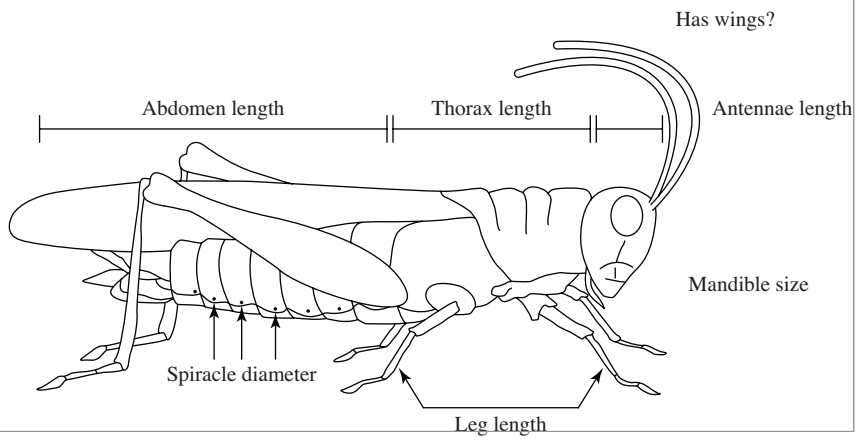
Consider the following data mining example; while it is not business related, it is easy to see the technique unfold visually. You are a researcher attempting to classify insects you have found into one of two groups (i.e., you are attempting to forecast the correct category for new insects found). The insects you find may be either katydids or grasshoppers. These insects look quite a fit alike, but there are distinct differences. They are much like ducks and geese; many similarities, but some important differences as well.

[4] "A Data Mining Method for Short-Term Load Forecasting in Power Systems," *Electrical Engineering in Japan*, Vol. 139, No. 2, 2002, pp. 12–22.



Source: Eamonn Keogh.

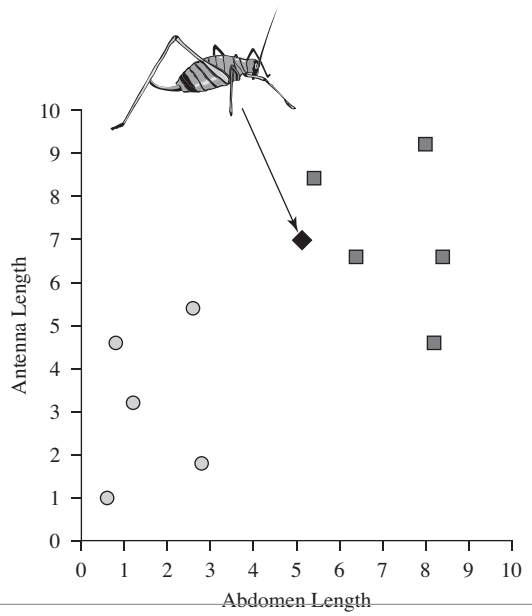
You have five examples of insects that you know are katydids and five that you know are grasshoppers. The unknown is thought to be either a katydid or grasshopper. Could we use this data set to come up with a set of rules that would allow us to classify any unknown insect as either a katydid or grasshopper? By seeing how this might be done by hand through trial and error we can begin to understand one general process that data mining techniques use.



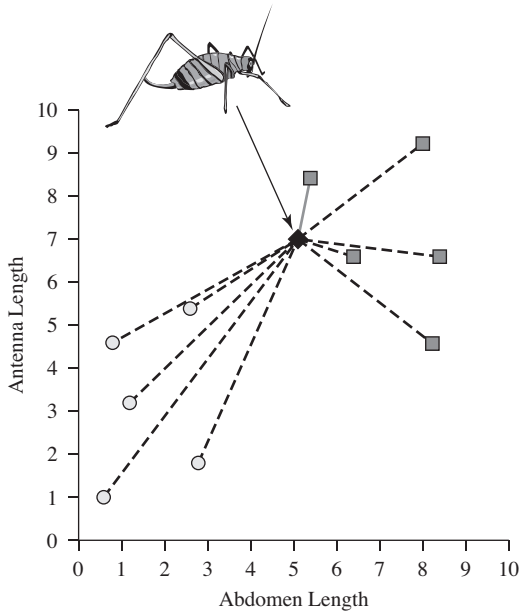
There are many characteristics we could use to aid in our classification. Some of them would include abdomen length, thorax length, leg length, and so on. The 10 insects we have in our database have the following values for the attributes titled abdomen length and antenna length.

Insect ID	Abdomen Length (mm)	Antenna Length (mm)	Insect Class
1	2.7	5.5	**Grasshopper**
2	8.0	9.1	**Katydid**
3	0.9	4.7	**Grasshopper**
4	1.1	3.1	**Grasshopper**
5	5.4	8.5	**Katydid**
6	2.9	1.9	**Grasshopper**
7	6.1	6.6	**Katydid**
8	0.5	1.0	**Grasshopper**
9	8.3	6.6	**Katydid**
10	8.1	4.7	**Katydids**
Unknown	5.1	7.0	**?**

The unknown insect is represented by the last row in the table. We have only included two attributes in our table for demonstration purposes. As we have seen in discussing business forecasting techniques, it is usually a good idea to graph the data in order to look for obvious relationships. We can do the same here by placing abdomen length on one axis and antenna length on the other, thus creating a scatterplot of the data.



The resulting plot is quite informative; the katydids (shown as squares) cluster in the upper right-hand corner of our plot while the grasshoppers (shown as circles) cluster in the lower left-hand corner of the plot. While neither characteristic by itself would do well in helping our classification, the combination of the two attributes might accurately define unknown insects. This unknown insect appears to fall closest to the katydids. But, can we come up with a mechanistic (i.e., rules-based) way of choosing the unknown as a katydid rather than as a grasshopper. One method would be to look at the geographical neighbors of the unknown insect. Which neighbors are the closest to the unknown? We could describe this process by drawing distance lines between the unknown insect and its neighbors.



If the distance to the unknown is closest to the katydids (as measured by summing the distance to katydid neighbors and comparing this to the summation of the distances to grasshopper neighbors), then the unknown is likely a katydid. In essence the k-Nearest-Neighbor model of data mining works in just this manner. In actual practice it is not necessary to calculate the distance to every neighbor; only a small subset of the neighbors are used. The "k" in k-Nearest-Neighbor refers to the number of nearest neighbors used in determining a category correctly.

When using k-Nearest-Neighbor we use a subset of the total data we have available (called a *training data* set) to attempt to identify observations in the training data set that are similar to the unknown. Scoring (or forecasting) new unknowns is assigning the unknowns to the same class as their nearest neighbors. While Euclidian distance is shown in the diagrams here, there are other metrics possible that are used to define neighbors and some are used in the various commercial data mining packages. What we are interested in is classifying future

Comments from the Field Cognos 2

Cognos is a company providing data mining software to a variety of industries. One of those industries is higher education. The University of North Texas has more than 33,000 students in 11 colleges that offer 96 different bachelor's degrees, 111 different master's degrees and 50 different doctorate degrees. The university uses data mining to identify student preferences and forecast what programs will attract current and future students. Admission, faculty hiring, and resource allocation are all affected by the outcomes of the data mining performed by Enterprise Information Systems at the university.

unknown insects, not the past performance on old data. We already know the classifications of the insects in the training data set; that's why we call it a training data set. It trains the model to correctly classify the unknowns by selecting closeness to the k-nearest-neighbors. So, the error rate on old data will not be very useful in determining if we have a good classification model. An error rate on a training set is not the best indicator of future performance. To predict how well this model might do in the real world at classifying *of* unknowns we need to use it to classify some data that the model has not previously had access to; we need to use data that was not part of the training data set. This separate data set is called the *validation data*. In one sense, this separation of data into a training data set and a validation data set is much like the difference between "in-sample" test statistics and "out-of-sample" test statistics. The real test of a business forecast was the "out-of-sample" test; the real test of a data mining model will be the test statistics on the validation data, not the statistics calculated from the training data.

In order to produce reliable measures of the effectiveness of a data mining tool researchers *partition* a data set before building a data mining model. It is standard practice to divide the data set into partitions using some random procedure. We could, for instance, assign each instance in our data set a number and then partition the data set into two parts called the training data and the validation data (sometimes researchers use a third partition called the *test set*). If there is a great deal of data (unlike the simple example of the katydids and grasshoppers), there is little trouble in using 60 percent of the data as a training set and the remaining 40 percent as a validation data set. This will insure that no effectiveness statistics are drawn from the data used to create the model. Thus an early step in any real data mining procedure is to partition the data. It is common practice to fold the validation data back into the training data and re-estimate the model if the model shows up well in the validation data.

A BUSINESS DATA MINING EXAMPLE: k-NEAREST-NEIGHBOR

What would such a model look like in a business situation? We now turn to examining a data set used by Shmueli, Patel, and Bruce.[5] This data set represents information on the customers a bank has in its data warehouse. These individuals

[5] Galit Shmueli, Nitin Patel, and Peter Bruce, *Data Mining for Business Intelligence,* (John Wiley & Sons, 2007).

TABLE 9.1
Universal Bank (Fictitious) Data
The Bank Has Data on a Customer-by-Customer Basis in these Categories

ID	Customer ID
Age	Customer's age in completed years
Experience	No. of years of professional experience
Income	Annual income of the customer ($000)
ZIP code	Home address, ZIP code
Family	Family size of the customer
CC Avg.	Average spending on credit cards per month ($000)
Education	Education level (1) Undergrad; (2) Graduate; (3) Advanced/Professional
Mortgage	Value of house mortgage if any ($000)
Personal loan	Did this customer accept the personal loan offered in the last campaign?
Securities account	Does the customer have a securities account with the bank?
CD account	Does the customer have a certificate of deposit (CD) account with the bank?
Online	Does the customer use Internet banking facilities?
Credit card	Does the customer use a credit card issued by Universal Bank?

have been customers of the bank at some time in the past; perhaps many are current customers in one dimension or another. The type of information the bank has on each of these customers is represented in Tables 9.1 and 9.2.

Universal Bank would like to know which customers are likely to accept a personal loan. What characteristics would forecast this? If the bank were to consider expending advertising efforts to contact customers who would be likely to consider a personal loan, which customers should the bank contact first? By answering this question correctly the bank will be able to optimize its advertising effort by directing its attention to the highest-yield customers.

This is a classification problem not unlike the situation of deciding in what class to place an unknown insect. The two classes in this example would be: (1) those with a high probability of accepting a personal loan (*acceptors*), and (2) those with a low probability of accepting a personal loan (*nonacceptors*). We will be unable to classify customers with certainty about whether they will accept a personal loan, but we may be able to classify the customers in our data into one of these two mutually exclusive categories.

The researcher would begin by first partitioning the Universal Bank data. Recall that partitioning the data set is the first step in any data mining technique. Since each row, or record, is a different customer we could assign a number to each row and use a random selection process to choose 60 percent of the data as a training set. All data mining software has such an option available. Once the data is selected into a training set it would look like Table 9.3. This table is produced using the XLMiner© software.

TABLE 9.2 **Universal Bank Customer Profiles**

Data includes both continuous variables such as income, as well as dummy variables like a personal loan

ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	Credit card
1	25	1	49	91107	4	1.60	1	0	0	1	0	0	0
2	45	19	34	90089	3	1.50	1	0	0	1	0	0	0
3	39	15	11	94720	1	1.00	1	0	0	0	0	0	0
4	35	9	100	94112	1	2.70	2	0	0	0	0	0	0
5	35	8	45	91330	4	1.00	2	0	0	0	0	0	1
6	37	13	29	92121	4	0.40	2	155	0	0	0	1	0
7	53	27	72	91711	2	1.50	2	0	0	0	0	1	0
8	50	24	22	93943	1	0.30	3	0	0	0	0	0	1
9	35	10	81	90089	3	0.60	2	104	0	0	0	1	0
10	34	9	180	93023	1	8.90	3	0	1	0	0	0	0
11	65	39	105	94710	4	2.40	3	0	0	0	0	0	0
12	29	5	45	90277	3	0.10	2	0	0	0	0	1	0
13	48	23	114	93106	2	3.80	3	0	0	1	0	0	0
14	59	32	40	94920	4	2.50	2	0	0	0	0	1	0
15	67	41	112	91741	1	2.00	1	0	0	1	0	0	0
16	60	30	22	95054	1	1.50	3	0	0	0	0	1	1
17	38	14	130	95010	4	4.70	3	134	1	0	0	0	0
18	42	18	81	94305	4	2.40	1	0	0	0	0	0	0
19	46	21	193	91604	2	8.10	3	0	1	0	0	0	0
20	55	28	21	94720	1	0.50	2	0	0	1	0	0	1

TABLE 9.3 **Training Data**
This data is a subset of the complete data set

Data											
Data Source	**Binneddata1D$20:$R$5019**										
Selected Variables	**ID**	**Age**	**Experience**	**Income**	**ZIP Code**	**Family**	**CCAvg**	**Education**	**Mortgage**	**Personal Loan**	**Securities Account**
Partitioning method	Randomly chosen										
Random Seed	12345										
# training rows	3000										
# validation rows	2000										

Selected Variables

Row Id.	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account
1	1	25	1	49	91107	4	1.6	1	0	0	1	0
4	4	35	9	100	94112	1	2.7	2	0	0	0	0
5	5	35	8	45	91330	4	1	2	0	0	0	0
6	6	37	13	29	92121	4	0.4	2	155	0	0	0
9	9	35	10	81	90089	3	0.6	2	104	0	0	0
10	10	34	9	180	93023	1	8.9	3	0	1	0	0
12	12	29	5	45	90277	3	0.1	2	0	0	0	0
17	17	38	14	130	95010	4	4.7	3	134	1	0	0
18	18	42	18	81	94305	4	2.4	1	0	0	0	0
19	19	46	21	193	91504	2	8.1	3	0	1	0	0
20	20	55	28	21	94720	1	0.5	2	0	0	1	0

Note that the "Row ID" in Table 9.3 skips from row 1 to row 5 and then from row 6 to row 9. This is because the random selection process has chosen customers 1, 4, 5, 6, and 9 for the training data set (displayed in Table 9.3) but has placed customers 2, 3, 7, and 8 in the validation data set (not displayed in Table 9.3). Examining the header to Table 9.3 you will note that there were a total of 5,000 customers in the original data set that have now been divided into a training data set of 3,000 customers and a validation data set of 2,000 customers.

When we instruct the software to perform a k-Nearest-Neighbor analysis of the training data the real data mining analysis takes place. Just as in the insect classification example, the software will compare each customer's personal loan experience with the selected attributes. This example is, of course, much more multidimensional since we have many attributes for each customer (as opposed to only the two attributes we used in the insect example). The program will compute the distance associated with each attribute. For attributes that are measured as continuous variables, the software will normalize the distance and then measure it (because different continuous attributes are measured in different scales). For the dummy type or categorical attributes, most programs use a weighting mechanism that is beyond the scope of this treatment.

The accuracy measures for the estimated model will tell if we have possibly found a useful classification scheme. In this instance we want to find a way to classify customers as likely to accept a personal loan. How accurately can we do that by considering the range of customer attributes in our data? Are there some attributes that could lead us to classify some customers as much more likely to accept a loan and other customers as quite unlikely? While the accuracy measures are often produced by the software for both the training data set and the validation data set, our emphasis should clearly be on those measures pertaining to the validation data. There are two standard accuracy measures we will examine: the *classification matrix* (also called the *confusion matrix*) and the *lift chart*. The classification matrix for the Universal Bank data training data is shown in Table 9.4.

When our task is classification, accuracy is often measured in terms of error rate, the percentage of records we have classified incorrectly. The error rate is often displayed for both the training data set and the validation data set in separate tables. Table 9.4 is such a table for the validation data set in the Universal Bank

TABLE 9.4
Classification Matrix (confusion matrix) for the Universal Bank Data
The number of nearest neighbors we have chosen is 3.

Validation Data Scoring—Summary Report (for k = 3)

Cut off prob. val. for success (updatable)	0.5

Classification Confusion Matrix

	Predicted Class	
Actual Class	1	0
1	118	76
0	8	1798

case. The table is correctly called either a *confusion matrix* or a *classification matrix*. In Table 9.4 there were 118 records that were correctly classified as "class 1" (i.e., probable personal loan candidates). They were correctly classified because these records represented individuals that did indeed take out a personal loan. However, eight records were classified as class 1 incorrectly; these were individuals that the model expected to take out a personal loan when, in fact, they did not historically. In addition, the table shows 1,798 records predicted to be class 0 (i.e., not probable loan candidates). These records were classified correctly since historically these individuals did not take out personal loans. Finally, 76 records were incorrectly classified as class 0 when they actually were loan acceptors. The table can then be used to compute a *misclassification rate*. This calculation simply shows the percentage of the records that the model has placed in the incorrect category. In this case we have 2,000 records in the validation data set and we have correctly classified 1,916 of them (1,798 + 118). But we have also incorrectly classified 8 records as class 1 when they were actually in class 0. We have also incorrectly classified 76 records as class 0 when they were actually in class 1. Thus we have incorrectly classified 84 records (8 + 76). The misclassification rate is the total number of misclassification divided by the total records classified (and is usually reported as a percentage). Most packages show the calculation and report it.

In Table 9.5 the misclassification rate is shown in the lower right-hand corner as 4.20 percent (calculated as 84/2,000 and expressed as a percentage). It should be noted that there are two ways in which the error occurred in our example and although some errors may be worse than others, the misclassification rate groups

TABLE 9.5
Classification Matrix (confusion matrix) and Misclassification Rate Calculation for the Universal Bank Data
The number of nearest neighbors has been chosen to be 3

Validation Data Scoring—Summary Report (for k = 3)

Cut off prob. val. for success (updatable)	0.5	(Updating the value here will NOT update value in detailed report)

Classification Confusion Matrix

	Predicted Class	
Actual Class	1	0
1	118	76
0	8	1798

Error Report

Class	#Cases	#Errors	#Error
1	194	76	39.14
0	1806	8	0.44
Overall	2000	84	4.20

these two types of errors together. While this may not be an ideal reporting mechanism, it is commonly used and displayed by data mining software. Some software also allows placing different costs on the different types of errors as a way of differentiating their impacts. While the overall error rate of 4.2 percent in the validation data is low in this example, the error of classifying an actual loan acceptor incorrectly as a non-loan acceptor (76 cases) is much greater than that of incorrectly classifying an actual nonacceptor as a loan acceptor (only 8 cases).

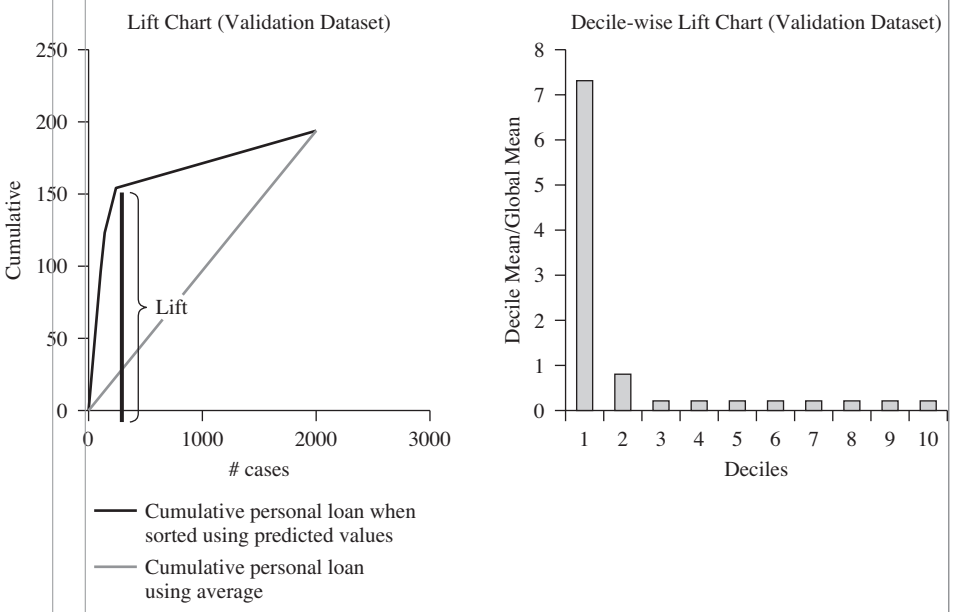
Notice that in both Tables 9.4 and 9.5 the summary report is for the k = 3 case (see the top of either table), meaning that we have used three neighbors (not three attributes) to classify all the records. The software has taken a "vote" of the three nearest neighbors in order to classify each record as either a loan acceptor or a non-loan acceptor. The software actually varied the number of nearest neighbors used from a small number to a large number and reported only the best results. Usually the researcher may specify the range over which the program searches and the program will respond by choosing the number of neighbors that optimizes the results (in this situation XLMiner© minimized the misclassification error rate).

In Table 9.6 the XLMiner© program provides an easy way to visualize how the number of nearest neighbors has been chosen. The misclassification error rate of 4.20 percent is lowest for three neighbors (it's actually the same for four neighbors but using the principle of parsimony, three nearest neighbors are chosen as the reported optimum).

TABLE 9.6
Validation Error Log for the Universal Bank Data
The best number of nearest neighbors has been chosen to be 3 because this provides the lowest misclassification rate

Validation Error Log for Different k			
Value of k	**% Error Training**	**% Error Validation**	
1	0.00	5.30	
2	1.30	5.30	
3	2.70	4.20	<— Best k
4	2.80	4.20	
5	3.43	4.70	
6	3.27	4.50	
7	3.70	4.85	
8	3.40	4.30	
9	4.47	5.15	
10	4.00	4.85	
11	4.83	5.65	
12	4.33	5.35	
13	5.00	5.60	
14	4.60	5.35	
15	5.20	5.70	
16	4.93	5.40	
17	5.33	5.75	
18	5.23	5.60	
19	5.83	6.00	
20	5.60	5.90	

FIGURE 9.1 Lift Chart and Decile-wise Lift Chart for the Universal Bank Validation Data Set



A second way of examining the accuracy and usefulness of a data mining model can be demonstrated with our Universal Bank example. All data mining programs will display a lift chart for any calculated solution; the one for the Universal Bank k-Nearest-Neighbor model is displayed in Figure 9.1.

Lift charts are the most common way (and perhaps the fastest) to compare different classification models. *Lift* is actually a ratio. Lift measures the change in concentration of a particular class when the model is used to select a group from the general population.

Consider why Universal Bank is attempting to classify the records in its database into *acceptors* and *nonacceptors*. Perhaps Universal Bank is considering a direct mailing to individuals in the database in order to solicit new personal loan applications. Based on previous experience, the percentage of individuals who respond favorably and take out a personal loan is 0.2 percent (that is not 2 percent, but two-tenths of 1 percent). Very few of the direct mailing recipients took out a personal loan. What if the bank could identify, before sending the mailing, the most likely *acceptors*? And what if the number of these likely *acceptors* was quite small relative to the size of the entire database? If the bank could successfully classify the database and identify these likely acceptors, then it would pay for the bank to restrict the direct mailing to only those individuals. Mailing and preparation costs would be saved and the bank would receive a *lift* in the percentage of recipients actually accepting a loan. What we may be able to help the bank do is to mail only to those customers with high probability of loan acceptance, as opposed to mailing to everyone in the database. Remember, most of the people represented

in the database are not likely loan acceptors. Only a relatively small number of the records in the database represent acceptors.

The lift curve is drawn from information about what the k-Nearest Neighbor model predicted in a particular case and what that individual actually did. The lift chart in Figure 9.1 is actually a cumulative gains chart. It is constructed with the records arranged on the *x*-axis left to right from the highest probability to lowest probability of accepting a loan. The *y*-axis reports the number of true positives at every point (i.e., the *y*-axis counts the number of records that represent actual loan acceptors).

Looking at the decile-wise lift chart in Figure 9.1 we can see that if we were to choose the top 10 percent of the records classified by our model (i.e., the 10 percent most likely to accept a personal loan) our selection would include approximately seven times as many correct classifications than if we were to select a random 10 percent of the database. That's a dramatic lift provided by the model when compared to a random selection.

The same information is displayed in a different manner in the lift chart on the left-hand side of Figure 9.1. This lift chart represents the cumulative records correctly classified (on the *y*-axis), with the records arranged in descending probability order on the *x*-axis. Since the curve inclines steeply upward over the first few hundred cases displayed on the *x*-axis, the model appears to provide significant lift relative to a random selection of records. Generally a better model will display higher lift than other candidate models. Lift can be used to compare the performance of models of different kinds (e.g., k-Nearest Neighbor models compared with other data mining techniques and is a good tool for comparing the performance of two or more data mining models using the same or comparable data. Notice the straight line rising at a 45-degree angle in the lift chart in Figure 9.1—this is a reference line. The line represents how well you might do by classifying as a result of random selection. If the calculated lift line is significantly above this reference line, you may expect the model to outperform a random selection. In the Universal Bank case the k-Nearest Neighbor model outperforms a random selection by a very large margin.

CLASSIFICATION TREES: A SECOND CLASSIFICATION TECHNIQUE

Our second data mining technique is variously called a classification tree, a decision tree, or a regression tree. As the name implies, it is, like k-Nearest Neighbor, a way of classifying or dividing up a large number of records into successively smaller sets in which the members become similar to one another. Regression trees are used to predict or forecast categories rather than specific quantities. Data miners commonly use a tree metaphor to explain (and to display results from) this technique. Because the term *regression* is most often used to forecast a numeric quantity, when the data mining technique is predicting numeric quantities it is called a *regression tree.* When the technique is classifying by category, it is usually called either a *classification tree* or a *decision tree.*

As a child you may have played a game called "Animal, Mineral, or Vegetable ." The origin of the game's name some believe arises from the fifteenth-century belief that all life was either animal or vegetable, while all inanimate objects were mineral. Thus, the three categories could effectively separate all matter into three neat categories. In the game, as you may recall, one player picks any object and other players must try to guess what it is by asking yes or no questions. The object is to ask the least number of questions before guessing the item correctly. In a sense, classification trees are like the game—we begin by knowing virtually nothing about the items we are sorting, but we make up rules along the way that allow us to place the records into different bins with each bin containing like objects. In "Animal, Mineral, or Vegetable," the set of questions you successfully used to determine the object's name would be the set of rules you could again use if the same object were to be chosen by another participant. In the same manner, we create a set of rules from our successful classification attempt, and these rules become the solution to the classification problem.

We now return to our insect classification problem.

In Figure 9.2 we ask first if the abdomen length is greater than 7.1. The vertical line drawn at a value of 7.1 is the graphical representation of this question (or rule). Note that when we draw this line all the known instances to the right of the line are katydids—we have a uniform classification on that side of the line. To the left of the line, however, we have a mix of katydids and grasshoppers in the known instances.

A further question (or rule) is necessary to continue the classification. This time we ask whether the antenna length is greater than six. The horizontal line drawn at a value of six in Figure 9.2 is the graphical representation of this question (or rule). An examination now of the entire set of known instances shows that

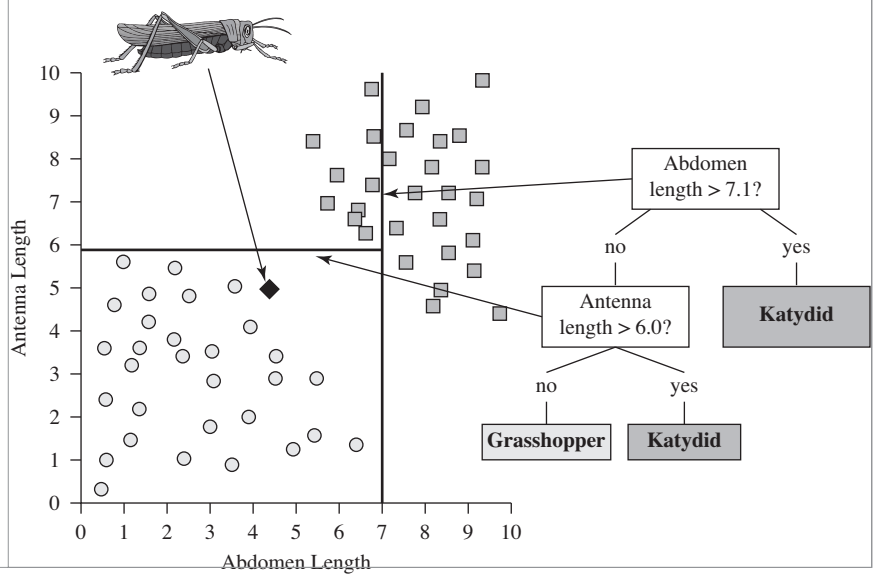


FIGURE 9.2
The Insect Classification Problem Viewed as a Classification Tree Exercise

Source: Eamonn Keogh.

there is homogeneity in each region defined by our two questions. The right-hand region contains only katydids as does the topmost region in the upper left-hand corner. The bottommost region in the lower left-hand corner, however, contains only grasshoppers. Thus we have divided the geometric attribute space into three regions, each containing only a single type of insect.

In asking the two questions to create the three regions, we have also *created* the rules necessary to perform further classifications on unknown insects. Take the unknown insect shown in the diagram with an antenna length of 5 and an abdomen length of 4.5. By asking whether the unknown has an abdomen length of greater than 7.1 (answer = no) and then asking whether the antenna length is greater than 6 (answer = no), the insect is correctly classified as a grasshopper.

In our example we have used only two attributes (abdomen length and antenna length) to complete the classification routine. In a real world situation we need not confine ourselves to only two attributes. In fact, we can use many attributes. The geometric picture might be difficult to draw but the decision tree (shown on the right-hand side of Figure 9.2) would look much the same as it does in our simple example. In data mining terminology, the two decision points in Figure 9.2 (shown as "abdomen length $>$ 7.1" and "antenna length $>$ 6") are called *decision nodes*. Nodes in XLMiner© are shown as circles with the decision value shown inside. They are called decision nodes because we classify unknowns by "dropping" them through the tree structure in much the same way a ball drops through Pachinko game. (See Figure 9.3).

The bottom of our classification tree in Figure 9.2 has three leaves. Each *leaf* is a terminal node in the classification process; it represents the situation in which all the instances that follow that *branch* result in uniformity. The three leaves in Figure 9.2 are represented by the shaded boxes at the bottom of the diagram. Data mining classification trees are *upside-down* in that the leaves are at the bottom while the root of the tree is at the top; this is the convention in data mining circles. To begin a *scoring* process all the instances are at the root (i.e., top) of the tree; these instances are partitioned by the rules we have determined with the known instances. The result is that the unknown instances move downward through the tree until reaching a leaf node at which point they are (hopefully) successfully classified.

At times the classification trees can become quite large and ungainly. It is common for data mining programs to *prune* the trees to remove branches. The unpruned tree was made using the training data set and it probably matches that data perfectly. Does that mean that this unpruned tree will do the best job in classifying new unknown instances? Probably not. A good classification tree algorithm will make the best split (at the first decision node) first followed by decision rules that are made up with successively smaller and smaller numbers of training records. These later decision rules will become more and more idiosyncratic. The result may be an unstable tree that will not do well in classifying new instances. Thus the need for pruning. Each data mining package uses a proprietary pruning algorithm that usually takes into account for any branch the added drop in the misclassification rate versus the added tree complexity. XLMiner© and other data

FIGURE 9.3
Classic Pachinko Game
A ball falls from the top through to the bottom and is guided by an array of pins. The user normally controls only the speed at which the ball enters the playing field. Like a slot machine the game is usually played in hope of winning a payoff.

mining programs use candidate tree formulations with the validation data set to find the lowest validation data set misclassification rate—that tree is selected as the final best-pruned tree. While the actual process is more complicated than we have described here, our explanation is essentially correct for all data mining software.

Classification trees are very popular in actual practice because the decision rules are easily generated and, more importantly, because the trees themselves are easy to understand and explain to others. There are disadvantages as well however. The classification trees can suffer from overfitting and if are not pruned well, these trees may not result in good classifications of new data (i.e., they will not score new data well). Attributes that are correlated will also cause this technique serious problems. It is somewhat similar to multicollinearity in a regression model. Be careful not to use features that are very closely correlated one with another.

A Business Data Mining Example: Classification Trees

We can once again use the Universal Bank data from Table 9.2 in an attempt to classify customers into likely or unlikely personal loan clients. The first step, as always, would be to partition the data into a training and validation data sets; the training data set was displayed in Table 9.3. Note that while the data miner selects the attributes that are to be used, the data mining software algorithm selects the decision rules and the order in which they are executed. Table 9.7 displays a portion of the classification tree output from XLMiner© for the Universal Bank data. We have used a number of attributes to help making up the decision rules; most of the attributes can be seen to intuitively affect whether a person is a likely personal loan candidate. Among the attributes used are:

- Customer's age
- Individual's average spending per month on credit cards
- Value of the individual's house mortgage
- Individual's annual income
- And others.

The scoring summary format is identical to the one we saw with the k-Nearest Neighbor technique. For the classification tree technique the misclassification rate is just 1.80 percent; this is even lower than the 4.20 percent achieved with the k-Nearest Neighbor model. A scant four individuals were expected to be likely to accept personal loans and yet did not do so.

TABLE 9.7
Scoring Summary Using the Best Pruned Tree on the Validation Data Set of the Universal Bank

Validation Data Scoring—Summary Report (Using Best Pruned Tree)

Cut off prob. val. for success (Updatable)	0.5	(Updating the value here will NOT update value in detailed report)

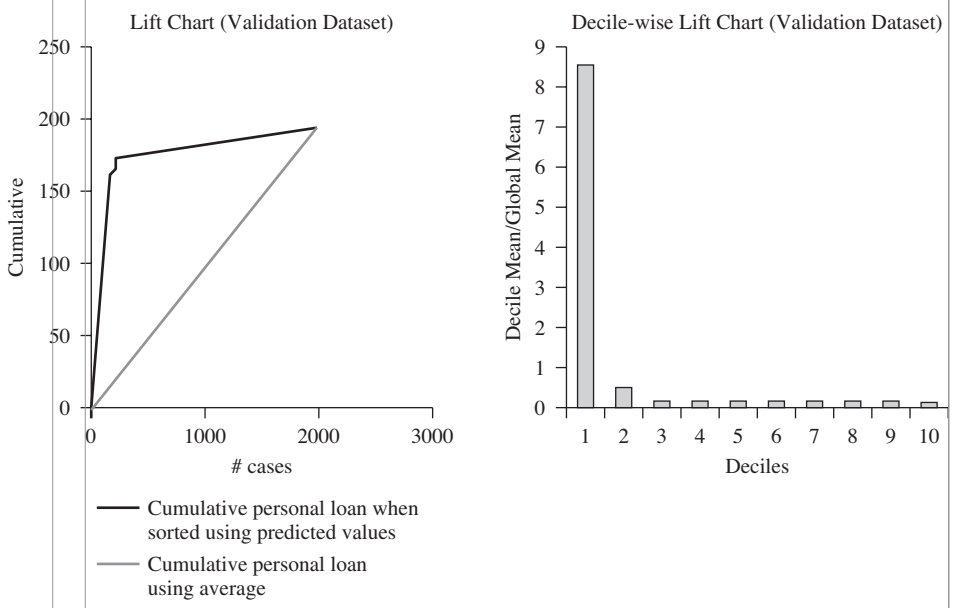
Classification Confusion Matrix

	Predicted Class	
Actual Class	1	0
1	182	32
0	4	1802

Error Report

Class	#Cases	#Errors	#Error
1	194	32	16.49
0	1806	4	0.22
Overall	2000	36	1.80

FIGURE 9.4 The Lift Chart and Decile-Wise Lift Chart Using the Best Pruned Tree on the Validation Data Set of the Universal Bank



Looking at the decile-wise lift chart on the right-hand side of Figure 9.4 we can see that if we were to choose the top 10 percent of the records classified by our classification tree model (i.e., the 10 percent most likely to accept a personal loan) our selection would include approximately 8.5 times as many correct classifications than if we were to select a random 10 percent of the database. That result is even more striking than the one we obtained with the k-Nearest Neighbor model.

The lift chart on the left-hand side of Figure 9.4 is a cumulative gains chart. Recall that it is constructed with the records arranged on the *x*-axis left to right from highest probability of accepting a loan to the lowest probability of accepting a loan. The *y*-axis reports the number of true positives at every point (i.e., the y-axis counts the number of records that represent actual loan acceptors). The fact that the cumulative personal loan line jumps sharply above the average beginning on the left side of the chart shows that our model does significantly better than choosing likely loan applicants at random. In other words, there is considerable lift associated with this model.

The actual topmost part of the classification tree that was produced by XLMiner© is displayed in Figure 9.5.

The classification tree first divides on the income variable. Is income greater than 100.5? That results in 1,507 of the instances being classified as unlikely to accept a personal loan; these individuals are shown in the leaf node in the upper left-hand corner of the diagram. XLMiner© then sorted on the basis of educational

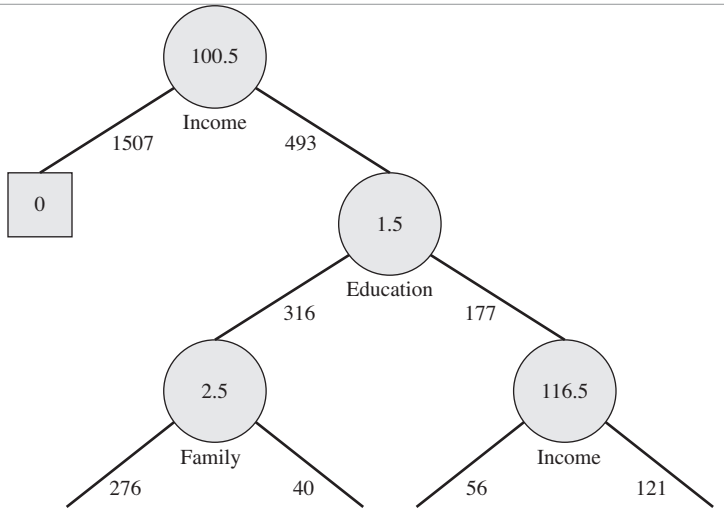


FIGURE 9.5
The Topmost Portion of the Classification Tree Using the Best Pruned Tree on the Validation Data Set of the Universal Bank

level followed by sorts based upon the family size of the customer (Family) and the annual income of the customer (Income). While examining the tree in Figure 9.5 is useful, it may be more instructive to examine the rules that are exemplified by the tree. Some of those rules are displayed in Table 9.8.

The rules displayed in Table 9.8 represent the same information shown in the diagram in Figure 9.5. Examining the first row of the table shows the split value as 100.5 for the split variable of income. This is the same as asking if the individual had a yearly income greater than 100.5? It is called a decision node because there are two branches extending downward from this node (i.e., it is not a terminal node or leaf). The second row of the table contains the information shown in the leaf on the left-hand side of the classification tree in Figure 9.5; this is called a terminal node or leaf because there are no successors. Row two shows that

TABLE 9.8 **The Topmost Portion of the Tree Rules Using the Best Pruned Tree on the Validation Data Set of the Universal Bank**

Best Pruned Tree Rules (Using Validation Data)

#Decision nodes	8

#Terminal nodes	9

Level	Node ID	Parent ID	Split Var	Split Value	Cases	Left Child	Right Child	Class	Node Type
0	0	N/A	Income	100.5	2000	1	2	0	Decision
1	1	0	N/A	N/A	1507	N/A	N/A	0	Terminal
1	2	0	Education	1.5	493	3	4	0	Decision
2	3	2	Family	2.5	316	5	6	0	Decision

1,507 cases are classified at 0, or unlikely to take out a personal loan in this terminal node. It is these rules displayed in this table that the program uses to score new data, and they provide a concise and exact way to treat new data in a speedy manner.

If actual values are predicted (as opposed to categories) for each case, then the tree is called a regression tree. For instance, we could attempt to predict the selling price of a used car by examining a number of attributes of the car. The relevant attributes might include the age of the car, the mileage the car had been driven to date, the original selling price of the car when new, and so on. The prediction would be expected to be an actual number, not simply a category. The process we have described could, however, still be used in this case. The result would be a set of rules that would determine the predicted price.

NAIVE BAYES: A THIRD CLASSIFICATION TECHNIQUE

A third and somewhat different approach to classification uses statistical classifiers. This technique will predict the probability that an instance is a member of a certain class. This technique is based on Bayes' theorem; we will describe the theorem below. In actual practice these Naive Bayes algorithms have been found to be comparable in performance to the decision trees we examined above. One hallmark of the Naive Bayes model is speed, along with high accuracy. This model is called *naive* because it assumes (perhaps naively) that each of the attributes is independent of the values of the other attributes. Of course this will never be strictly true, but in actual practice the assumption (although somewhat incorrect) allows the rapid determination of a classification scheme and does not seem to suffer appreciably in accuracy when such an assumption is made.

To explain the basic procedure we return to our insect classification example. Our diagram may be of the same data we have used before, but we will examine it in a slightly different manner.

The known instances of katydids and grasshoppers are again shown in Figure 9.6, but only a single attribute of interest is labeled on the y-axis: antenna length. On the right-hand side of Figure 9.6 we have drawn a histogram of the antenna lengths for grasshoppers and a separate histogram representing the antenna length of katydids.

Now assume we wish to use this information about a single attribute to classify an unknown insect. Our unknown insect has a measured antenna length of 3 (as shown in Figure 9.7). Look on the problem as an entirely statistical problem. Is this unknown more likely to be in the katydid distribution or the grasshopper distribution? A length of 3 would be in the far right tail of the katydid distribution (and therefore unlikely to be a part of that distribution). But a length of 3 is squarely in the center of the grasshopper distribution (and therefore it is more likely to be a member of that distribution). Of course there is the possibility that the unknown with an antenna length of 3 is actually part of the katydid distribu-

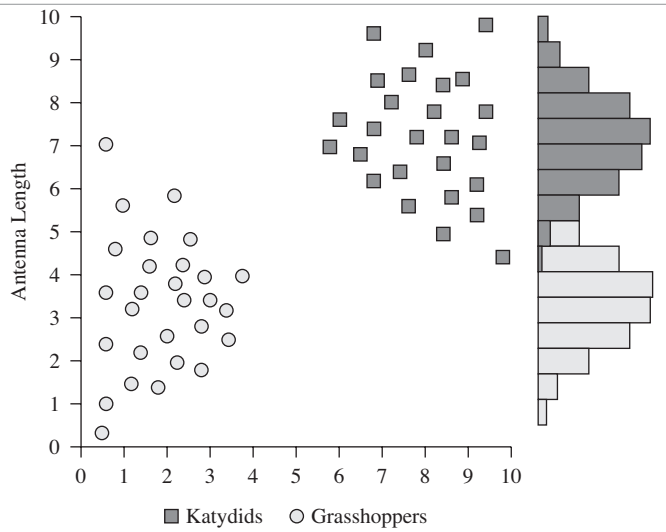


FIGURE 9.6
Insect Example This has Only a Single Attribute Displayed: Antenna length Abdomen length is still measured on the *x*-axis

Source: Eamonn Keogh.

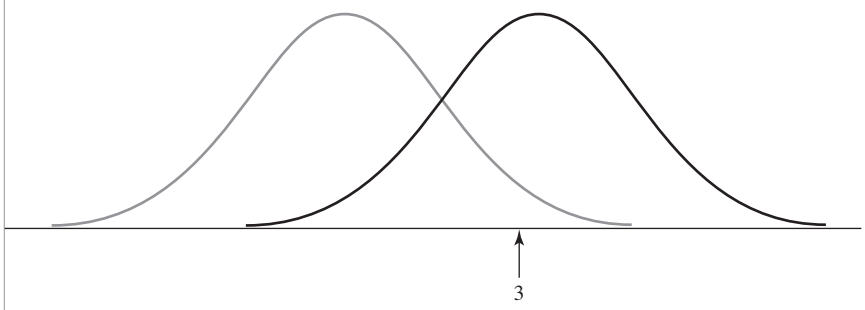


FIGURE 9.7
Histograms Representing Antenna Lengths Katydids are on the Left, and grasshoppers on the Right

Source: Eamonn Keogh.

tion (and therefore is actually a katydid) but that probability is small as evidenced by a length of 3 being in the small tail of the distribution. It is far more likely that our unknown is part of the grasshopper distribution (and is therefore truly a grasshopper). So far we have used only a single attribute. What if we again consider an additional attribute? Would that perhaps help our accuracy in making classifications?

Figure 9.8 represents two attributes (antenna length on the y-axis and abdomen length on the x-axis) for the known katydids and grasshoppers. By using the two attributes together we effectively create a quadratic boundary between the two classes of known insects. An unknown would be classified by its location above or below the boundary. One of the important features of the Naive Bayes model is that it handles irrelevant features quite well. If an irrelevant feature is included in the attributes list it has little effect on the classifications the model makes (and thus introduces little error).

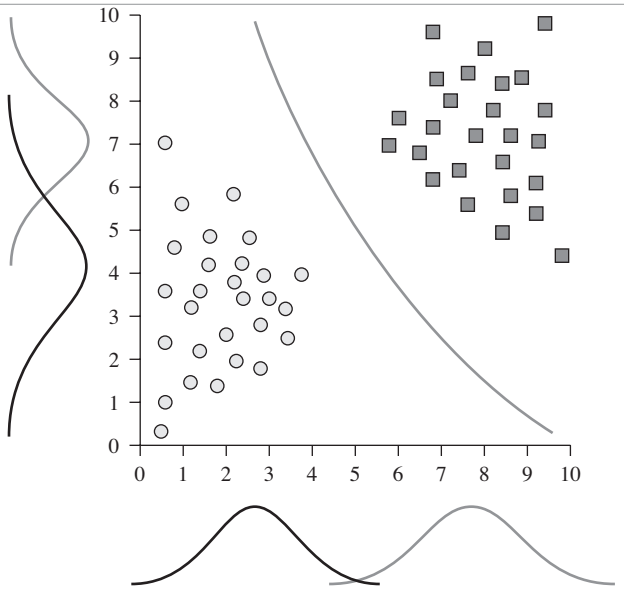


FIGURE 9.8
Two Sets of Histograms these Represent the Antenna Lengths of Katydids on the *y*-axis, and Abdomen Lengths on the *x*-axis

Source: Eamonn Keogh.

To examine this technique we will use actual data from the passenger list of the HMS *Titanic*. On Sunday evening April 12, 1912, the *Titanic* struck an iceberg. The ship sank a scant two hours and forty minutes later. We have somewhat complete information on 2,201 souls on the ship at the time of the accident. We say the information is "somewhat" complete because this data is based on a report made shortly after the event and the White Star line (the owners of the *Titanic*) kept their records in a peculiar manner[6]. For instance, boys are classified by the title "Master," but girls are not clearly distinguished from women. The data are not without some ambiguity but we can still attempt to ascertain characteristics of the survivors. We are attempting to classify individuals as survivors of the disaster or nonsurvivors (i.e., those who perished). Our data looks like the following:

Age	Sex	Class	Survived
Adult	Male	First	Alive
Adult	Male	First	Alive
Adult	Male	First	Alive
Adult	Male	First	Alive
Adult	Male	First	Alive
Adult	Male	First	Alive
Adult	Male	First	Alive
Adult	Male	First	Alive
Adult	Male	First	Alive

[6] The *Titanic* data set is used by permission of Professor Robert J. MacG. Dawson of Saint Marys University, Halifax, Nova Scotia. See "The Unusual Episode, Data Revisited." Robert J.MacG. Dawson, *Journal of Statistics Education* , v.3, n.3 (1995).

TABLE 9.9
Validation Data Scoring for the Naive Bayes Model of *Titanic* Passengers and Crew

Validation Data Scoring—Summary Report

Cut off prob. val. for success (updatable)	0.5	(Updating the value here will NOT update value in detailed report)

Classification Confusion Matrix

	Predicted Class	
Actual Class	Alive	Dead
Alive	172	123
Dead	107	478

Error Report

Class	#Cases	#Errors	%Error
Alive	295	123	41.69
Dead	585	107	18.29
Overall	880	230	26.14

The data set contains information on each of the individuals on the *Titanic*. We know whether they were adult or child, whether they were male or female, the class of their accommodations (first class passenger, second class, third class, or crew), and whether they survived that fateful night. In our list 711 are listed as alive while 1,490 listed as dead; thus only 32 percent of the people on board survived.

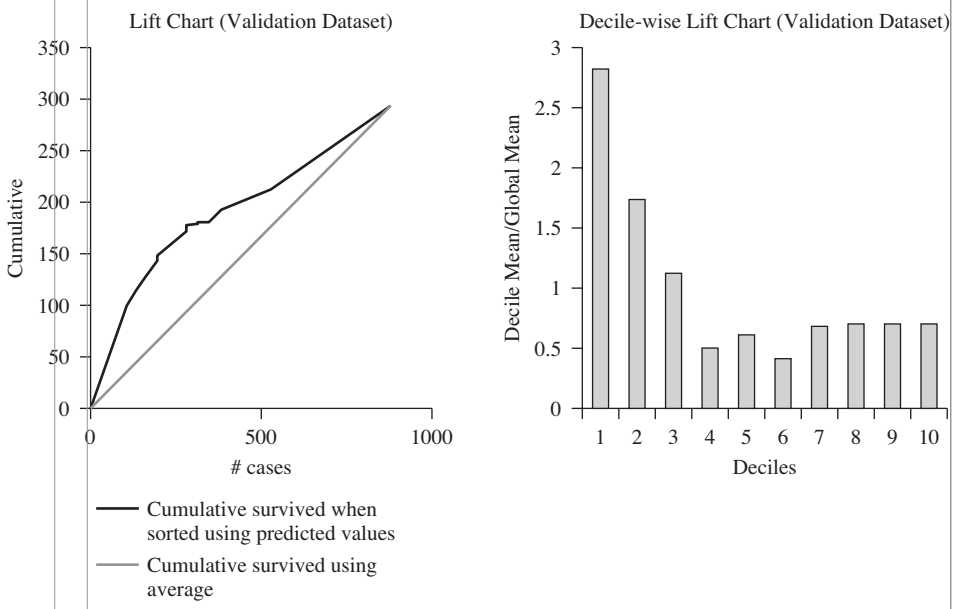
What if we wished to examine the probability that an individual with certain characteristics (say, an adult, male crew member) were to survive? Could we use the Naive Bayes method to determine the probability that this person survived? The answer is Yes; that is precisely what a Naive Bayes model will do. In this case we are classifying the adult, male crew member into one of two categories: survivor or dead.

The Naive Byes process begins like our two previous techniques; the data set is divided into a training data set and a validation data set. In Table 9.9 we present the Validation Summary Report for the model as computed in XLMiner©.

The misclassification rate computed by XLMiner© is 26.14 percent but the lift chart and the decile-wise lift chart in Figure 9.9 show that the model does improve on naively selecting a class at random for the result.

The Naive Bayes model rests on Bayes' theorem. Simply stated, *Bayes' theorem* predicts the probability of a prior event (called a posterior probability) given that a certain subsequent event has taken place. For instance, what is the probability that a credit card transaction is fraudulent given that the card has been reported lost? Note that the reported loss preceded the current attempted use of the credit card.

FIGURE 9.9 Lift Chart and Decile-Wise Lift Chart for the Naive Bayes *Titanic* Model



The posterior probability is written as P($A \mid B$). Thus , P($A \mid B$) is the probability that the credit card use is fraudulent given that we know the card has been reported lost. P(A) would be called the prior probability of A and is the probability that any credit card transaction is fraudulent regardless of whether the card is reported lost.

The Bayesian theorem is stated in the following manner:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

where:

$P(A)$ is the prior probability of A. It is *prior* in the sense that it does not take into account any information about B.

$P(A \mid B)$ is the conditional probability of A, given B. It is also called the *posterior probability* because it is derived from or depends upon the specified value of B. This is the probability we are usually seeking to determine.

$P(B \mid A)$ is the conditional probability of B given A.

$P(B)$ is the prior probability of B.

An example will perhaps make the use of Bayes' theorem clearer. Consider that we have the following data set showing eight credit card transactions. For each transaction we have information about whether the transaction was fraudulent and whether the card used was previously reported lost (see Table 9.10).

TABLE 9.10
Credit Card Transaction Data Set

Transaction No.	Fraudulent?	Reported Lost?
1	Yes	Yes
2	No	No
3	No	No
4	No	No
5	Yes	Yes
6	No	No
7	No	Yes
8	Yes	No

Applying Bayes' theorem:

$$P(Fraud \mid Card\ Reported\ Lost) = \frac{P(Lost \mid Fraud)P(Fraud)}{P(Lost)}$$

$$= \frac{\left(\frac{2}{3}\right)\left(\frac{3}{8}\right)}{\frac{3}{8}} = .667$$

and

$$P(NonFraud \mid Card\ Reported\ Lost) = \frac{P(Lost \mid NonFraud)P(NonFraud)}{P(Lost)}$$

$$= \frac{\left(\frac{1}{5}\right)\left(\frac{5}{8}\right)}{\frac{3}{8}} = .333$$

Thus, the probability of a fraudulent transaction if the card has been reported lost is 66.7 percent. The probability of a nonfraudulent transaction if the card has been reported lost is 33.3 percent.

Returning to the *Titanic* data and the Naive Bayes model calculated by XLMiner©, we may now demonstrate the calculation of the posterior probabilities of interest. These are the answers to our question concerning the probability that an adult, male crew member would survive the disaster. XLMiner© produces an additional output for the Naive Bayes model displaying the prior class probabilities and the calculated conditional probabilities. These are displayed in Table 9.11.

To answer our question concerning the survival probability of an adult, male crew member we need once again to apply Bayes' theorem. We first need to calculate the conditional probabilities required in the Bayes' theorem:

Conditional probability of "alive"if you were a crew member, male, and adult:

P(alive) = (0.295673077)(.53125)(.9375)(0.314912945) = 0.046373782

TABLE 9.11
Prior Class Probabilities and Conditional Probabilities Calculated in XLMiner© for the Titanic Data

Prior Class Probabilities

According to relative occurrences in training data

Class	Prob.	
Alive	0.314912945	← Success Class
Dead	0.685087055	

Conditional Probabilities

	Classes →			
	Alive		Dead	
Input Variables	Value	Prob	Value	Prob
Age	Adult	0.9375	Adult	0.964640884
	Child	0.0625	Child	0.035359116
Sex	Female	0.46875	Female	0.082872928
	Male	0.53125	Male	0.917127072
Class	Crew	0.295673077	Crew	0.450828729
	First	0.295673077	First	0.071823204
	Second	0.146634615	Second	0.10718232
	Third	0.262019231	Third	0.370165746

Note that we are now multiplying probabilities assuming they are independent. In like manner we calculate the "dead" probability:

Conditional probability of "dead"if you were a crew member, male, and adult:

P(alive) = (0.9640884)(0.9171270720.450828729)(0.685087055) = 0273245188

These two conditional probabilities can now be used in Bayes theorem to calculate the posterior probabilities we are seeking:

Posterior probability of "alive"if you were a crew member, male, and adult:

= (0.046373782)/(0.046373782 + 0.273245188) = 0.145090831

and

Posterior probability of "dead"if you were a crew member, male, and adult:

= (0.273245188)/(0.273245188 + 0.046373782) = 0.854909169

Naive Bayes has assumed the attributes have independent distributions. While this is not strictly true, the model seems to work well in situations where the assumption is not grossly violated. Use of larger data sets will all but eliminate the problem of including irrelevant attributes in the model. The effects of these irrelevant attributes are minimized as the data set becomes larger. We can again use the

FIGURE 9.10
The Naive Bayes Model Applied to the Universal Bank Data
Included are confusion matrix, misclassification rate, and lift charts

Validation Data Scoring—Summary Report

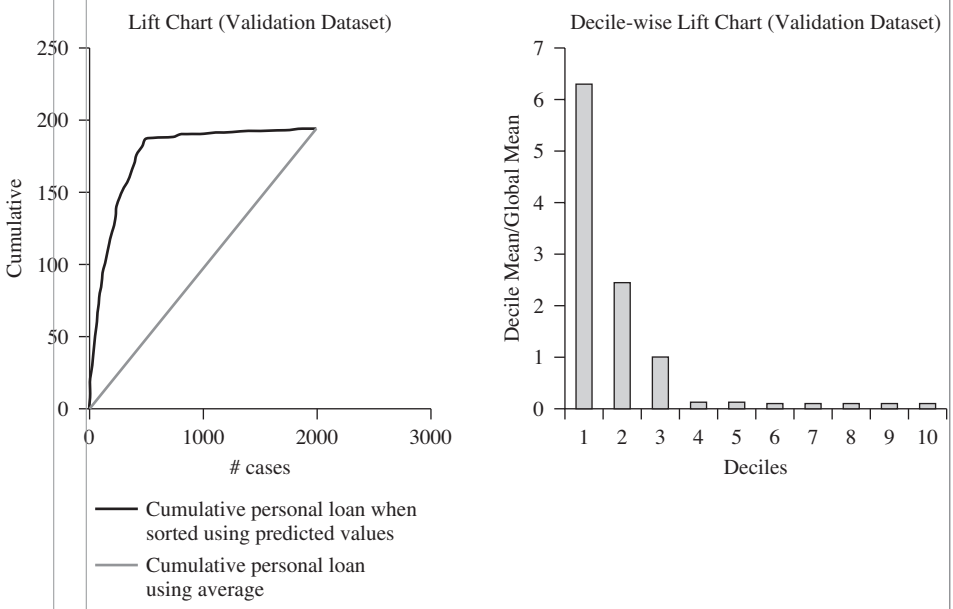
Cut off prob. val. for success (updatable)	0.5	(Updating the value here will NOT update value in detailed report)

Classification Confusion Matrix

	Predicted Class	
Actual Class	1	0
1	122	72
0	77	1729

Error Report

Class	#Cases	#Errors	%Error
1	194	72	37.11
0	1806	77	4.26
Overall	2000	149	7.45



Universal Bank data and apply the Naive Bayes model in order to predict customers that will accept a personal loan. Figure 9.10 displays the Naive Bayes results from XLMiner© for the Universal Bank data.

Once again it is clear that the model performs much better than a naive selection of individuals when we try to select possible loan acceptors. Looking at the decile-wise lift chart on the right-hand side of Figure 9.10 we can see that if we

Comments from the Field 3

TRUE LIFT

Victor S. Y. Lo of Fidelity Investments uses logistic regression models and Naive Bayes models to differentiate between customers who would be expected to respond to some sort of stimulus (e.g., a mail campaign or a telephone call) from customers who would respond without any added stimulus. He calls the result *true lift*. The methods Lo use specifically find customers who will take a desirable action regardless of the treatment. His aim was to find the characteristics of customers whose response decisions can be positively influenced by a campaign. Customer development campaigns, including upselling and cross-selling, would benefit greatly from such a tool.

Source: Victor S. Y. Lo, "The True Lift Model—A Novel Data Mining Approach to Response Modeling in Database Marketing," *SIGKDD Explorations,* December 2002, Volume 4, Issue 2, p. 74–86.

were to choose the top 10 percent of the records classified by our classification tree model (i.e., the 10 percent most likely to accept a personal loan) our selection would include approximately 6.5 times as many correct classifications than if we were to select a random 10 percent of the database. While Naive Bayes models do extremely well on training data, in real world applications these models tend not to do quite as well as other classification models in some situations. This is likely due to the disregard of the model for attribute interdependence. In many real world situations, however, Naive Bayes models do just as well as other classification models. While the Naive Bayes model is relatively simple, it makes sense to try the simplest models first and to use them if they provide sufficient results. Clearly data sets that contain highly interdependent attributes will fare poorly with Naive Bayes.

REGRESSION: A FOURTH CLASSIFICATION TECHNIQUE

Our final classification data mining technique is logistic regression or logit analysis (both names refer to the same method). This technique is a natural complement to linear least squares regression. It has much in common with ordinary linear regression models we examined in Chapters 4 and 5. Ordinary linear regression provides a universal framework for much of economic analysis; its simplified manner of looking at data has proven useful to researchers and forecasters for decades. Logistic regression serves the same purpose for categorical data. The single most important distinction between logistic regression and ordinary regression is that the dependent variable in logistic regression is categorical (and not continuous). The explanatory variables, or attributes, may be either continuous or categorical (as they were in linear least squares models). Just like the ordinary linear regression model, logistic regression is able to use all sorts of extensions and sophisticated variants. Logistic regression has found its way into the toolkits of not only forecasters and economists; also, for example, into those of toxicologists and epidemiologists.

TABLE 9.12
Data on 20 Students and Their Test Performance and Hours of Study.

Student No.	Hours of Study	Pass/Fail
1	2.5	0
2	22.6	1
3	17.8	0
4	5.4	0
5	14	0
6	13.3	1
7	26	1
8	33.1	1
9	13.6	0
10	45.3	1
11	1.9	0
12	31.4	1
13	27	1
14	10.1	0
15	2.7	0
16	16.3	1
17	14.5	1
18	4.5	0
19	22.6	1
20	17.1	1

The Universal Bank situation we have been examining provides a case in point. The dependent variable, the item we are attempting to forecast, is dichotomous—either a person accepts a loan or rejects the loan. There is no continuous variable here; it is more like an on/off switch. But why are we unable to use linear least squares models on this data?

Consider Table 9.12—it contains information about 20 students, the hours they spent studying for a qualifying exam, and their results. If they passed the exam the table shows a 1, if they failed the exam, the table shows a 0.

If we graph this data as a scatterplot we see there are two possible outcomes: pass (shown as 1) and fail (shown as 0).

FIGURE 9.11
Scatterplot of Student Performance

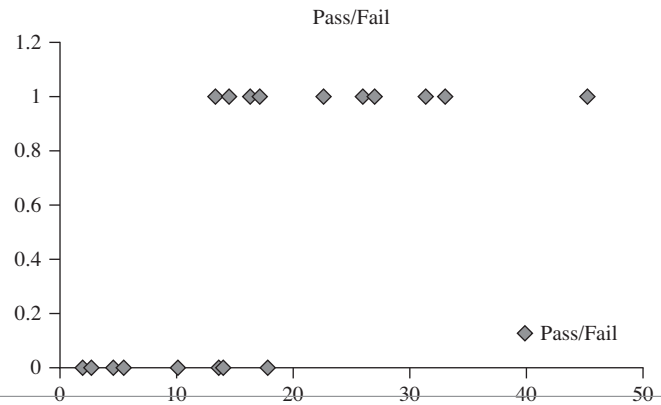


TABLE 9.13 Linear Least Squares Regression

Audit Trail--Coefficient Table (Multiple Regression Selected)

Series Description	Included in Model	Coefficient	Standard Error	T-test	P-value	Elasticity	Overall F-test
Hours of Study	Yes	0.03	0.01	4.47	0.00	1.00	
Pass/Fail	Dependent	0.0020532033	0.15	0.01	0.99		20.02

It appears from the scatterplot that students who spent more time studying for the exam had a better chance of passing. We might seek to quantify this perception by running a least squares regression using "hours of study" as the independent variable and "pass/fail" as the dependent variable. Running this regression results in the output in Table 9.13.

Since the "hours of study" coefficient is positive (+0.03), it appears to indicate that more study leads to a higher probability of passing. But, is the relationship correctly quantified? Suppose an individual studies for 100 hours. How well will this individual do on the exam? Substituting into the regression equation we have:

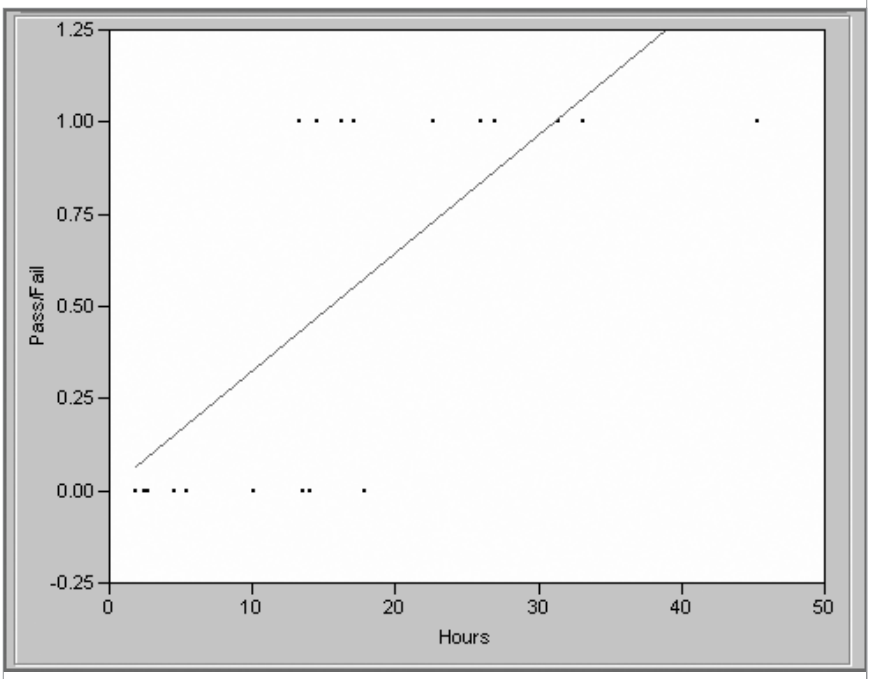
$$\text{Pass/fail} = 0.002053 + (0.032072) \times (100)$$
$$3.209253 = 0.002053 + (0.032072) \times (100)$$

What does this mean? Is the predicted grade 3.209 percent? This doesn't seem to make sense. Examining the regression line estimated and superimposing it on the data scatter may make the problem clear (see Figure 9.12).

The difficulty becomes clearer when examining the diagram. There are only two states of nature for the dependent variable (pass and fail). However, the regression line plotted in Figure 9.12 is a straight line sloping upward to the right and predicting values all along its path. When predicting the outcome from 100 hours of study the regression chooses a number (i.e., 3.209253) that is much greater than the maximum value of 1 exhibited in the data set. Does this mean the individual has passed the test 3.209 times? Or does this mean that the expected score is 3.209 percent? Or does this have any meaningful explanation at all? This result means that we have used an inappropriate tool in attempting to find the answer to our question. In earlier chapters we assumed the dependent variable was continuous, this one is not. Linear least squares regression does not restrict the predictions of the dependent variable to a range of zero to one as we would like in this case.

We would really like to use this same data but predict the probability that an individual would pass the test given a certain number of hours of study. We will modify the linear least squares model by modifying what we use as the dependent variable. Ordinarily we simply use Y as the dependent variable; in logistic regression we will use a function of Y as the dependent variable instead. This function of the dependent variable will be limited to values between zero and one. The function we use is called a *logit* and that is the reason the technique is called logistic regression.

FIGURE 9.12
Linear Least Squares Regression Plot

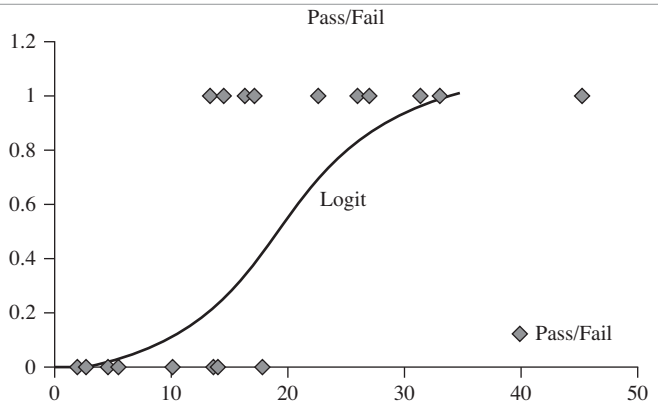


The logit is Log ($e^{\alpha+\beta 1\times 1+\beta 2\times 2+\cdots+\beta p x^p}$). You will recognize this as being similar to our explanation of the logistic curve in Chapter 3. In fact the concepts are one and the same. We are going to use some knowledge of how growth works in the real world just as we did in that chapter. Recall that the diffusion models' assumed growth proceeded along an s-curve. When we used these models to predict new product sales we did so in the knowledge that real world new products almost always follow such a path. We now make the same assumption that real-world probabilities will behave in a similar manner. This assumption has withstood the test of time as logistic regression has proven very predictive and accurate in actual practice.

The logistic regression model will estimate a value for pass/fail as a probability with zero as the minimum and one as the maximum. If we were to look at the entire range of values that a logistic regression would estimate for the student data it would appear like that in Figure 9.13. Recall that "hours of study" are represented on the x-axis while "pass/fail" is represented on the y-axis. If, for instance, an individual had studied for a scant 10 hours the model would predict a probability of passing somewhere near 10 percent (since the y-axis represents the values from zero to one it can be read directly as the probability of occurrence of the dependent event). However, if the individual in question were to have studied for 30 hours the probability of passing is predicted to be near 90 percent.

Let's examine the Universal Bank data with a logistic regression model and note the difference in the output from the other classification models we have

FIGURE 9.13
The Logit Estimated for the Student Data



used. As always, we begin by using a partition of the original data set to act as a training data set. In much the same way we used ForecastX to set up an ordinary linear regression, XLMiner© can be used to set up a logistic regression. The dependent variable is specified. The dependent variable in any logistic regression is categorical; in the Universal Bank case it is the personal loan variable that takes on a value of either 1 (if a personal loan is accepted) or 0 (if no personal loan is accepted). The independent variables are also specified just as in ForecastX. Note that unlike the student study example presented above, the Universal Bank logistic regression will include more than one independent variable. In this sense, it is a multiple logistic regression. Table 9.14 shows the output from the XLMiner© program for this data.

The output resembles ordinary linear regression output from ForecastX. There is a constant term, the values of the coefficients are reported, and standard errors

TABLE 9.14
Logistic Regression for the Universal Bank Data

The Regression Model				
Input Variables	**Coefficient**	**Std. Error**	**p-value**	**Odds**
Constant term	−12.8019095	2.16737223	0	*
Age	−0.04461157	0.08051153	0.57950926	0.95636886
Experience	0.05816582	0.07969882	0.46549997	1.05989075
Income	0.05698515	0.00351325	0	1.05864012
Family	0.62984651	0.09647165	0	1.87732232
CCAvg	0.11668219	0.05372836	0.02987786	1.12376225
Education	1.80500054	0.15606253	0	6.07997465
Mortgage	0.00141519	0.0007293	0.05232161	1.00141621
Securities Account	−0.8171795	0.37658975	0.03001092	0.44167566
CD Account	3.56768751	0.41729182	0	35.43455887
Online	−0.70467436	0.21116103	0.00084645	0.49426949
CreditCard	−1.10061717	0.26931652	0.00004375	0.33266568

for those coefficients allow p-values to be reported. The interpretation of these variables is much the same as it was with ordinary linear regression.

For example, the Family variable represents the family size of the customer. Since the logistic regression coefficient for this variable is positive (i.e., +0.62984651) we would expect that the probability of accepting a personal loan increases with family size. Further, since the p-value for this variable is very small (reported as zero in the printout but actually a very small number) we believe that the coefficient is significant at the 99 percent level. That is, we believe that there is very little chance that the real relationship between family size and the acceptance of a personal loan is zero. This ability to examine individual attributes is similar to the manner in which we examined the individual coefficients of an ordinary linear regression.

Table 9.15 displays the validation data set confusion matrix and misclassification rate. This information allows us to judge in part the overall fit of the logistic regression model. The confusion matrix and misclassification rate give an overall sense of fit. In this case the 5.05 percent misclassification rate would indicate how well we believe the model would classify new data into the correct categories.

The validation data set lift charts are also used to judge the overall fit of the model. Examining the right-hand side of Figure 9.14 shows that selecting the top 10 percent of the training data observations (i.e., those persons judged most likely to accept a personal loan), resulted in a better that 7 times result when compared to selecting individual cases at random. The same information is again displayed in the left-hand side of Figure 9.14 in the lift chart. The steep rise in the lift compared to the reference line for a naive selection of cases (i.e., the straight line in the figure) shows significant explanatory power in the model.

TABLE 9.15
Logistic Regression Confusion Matrix and Misclassification Rate

Validation Data Scoring—Summary Report

Cut off prob. val. for success (updatable)	0.5	(Updating the value here will NOT update value in detailed report)

Classification Confusion Matrix

	Predicted Class	
Actual Class	1	0
1	125	69
0	32	1774

Error Report

Class	#Cases	#Errors	%Error
1	194	69	35.57
0	1806	32	1.77
Overall	2000	101	5.05

Comments from the Field: Fool's Gold 4

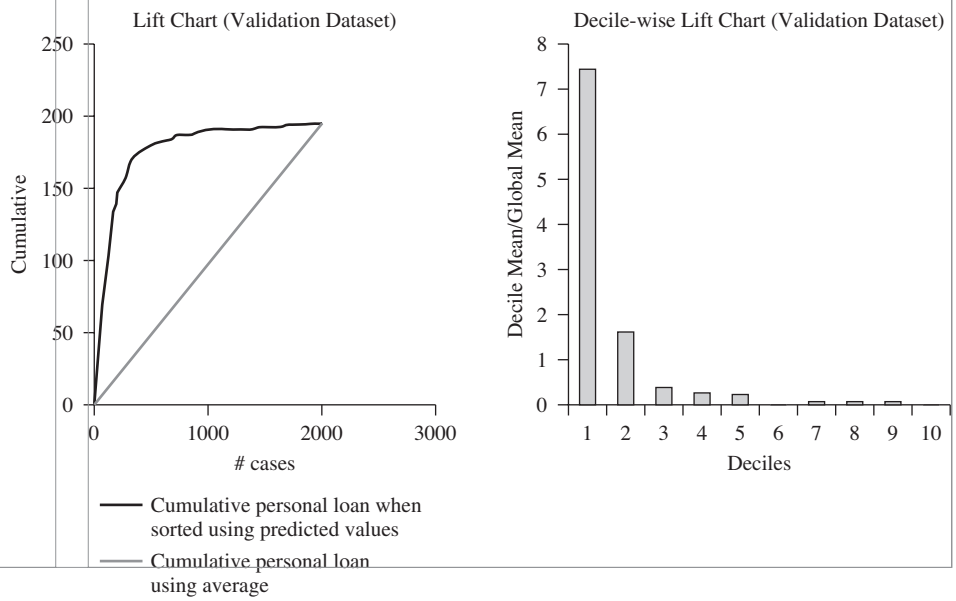
BusinessWeek related the story of Michael Drosnin, author of a bestselling book, *The Bible Code*. In the book Drosnin claimed to have found references in the Bible to President Bill Clinton, dinosaurs and the Land of Magog. Peter Coy, the author of the article, pointed out many of the useful tasks to which data mining had been put (e.g., weeding out credit card fraud, finding sales prospects, and discovering new drugs).

But Coy also pointed out that data mining was the method used by Michael Drosnin to make his "discoveries" in the Bible. It seems Drosnin wrote out the Hebrew Bible as a grid of letters and used data mining to look for words on the diagonal, up, down, and across. Not surprisingly he found some recognizable words and references. Cryptic messages appeared according to Coy such as the close juxtaposition of the word dinosaur with the word asteroid. According to Andrew Lo of the Massachusetts Institute of Technology, ''Given enough time, enough attempts, and enough imagination, almost any pattern can be teased out of any data set."

The moral to the story is that a formula that fits the data may not have any predictive power at all! There is always the chance that what a data mining technique "observes" or "discovers" may just be a coincidence in the current data set and not something that is reproducible in the future with other data. The chance of this being true in data mining is more prominent than in the standard business forecasting routines we presented earlier in the text. Most of those techniques relied on proven economic theory as a basis for specifying a particular type of model. With data mining it is the data itself that specifies the model and the analyst should be wary of making the same mistake as Michael Drosnin. Coy gives the example of David J. Leinweber, managing director of First Quadrant Corporation who sifted through a United Nations CD-ROM and discovered that historically, the single best predictor of the Standard & Poor's 500-stock index was butter production in Bangladesh. Leinweber called this an example of "stupid data miners' tricks."

Source: Peter Coy, "Commentary: He Who Mines Data May Strike Fool's Gold." *BusinessWeek* , June 16, 1997, p. 40.

FIGURE 9.14 **Validation Data Lift Charts for the Logistic Regression Model**



Summary

In this chapter we have covered four classification techniques that are commonly used by real data miners. Classification, however, is only a single aspect of data mining. In general there is no one best classification technique; the individual data in a particular situation will determine the best technique to use. The diagnostic statistics will lead the researcher to choose an appropriate model; there may be no optimal model.

Data mining also uses other tools such as clustering analysis and neural network analysis. These tools are not covered here but there are excellent resources for those interested in pursuing the study of data mining. The growth in the use of commercial data mining tools rivals the growth in business forecasting software sales; SAS Enterprise Miner and SPSS Clementine have become important additions to the forecaster's toolkit in recent years.

Suggested Readings

Berry, Michael J. A.; and Gordon S. Linhoff. *Data Mining Techniques for Marketing, Sales, and Customer Relationship Management*. Indianapolis: Wiley Publishing, Inc., 2004.

Cramer, J.S. *Logit Models from Economics and Other Fields*. Cambridge: Cambridge University Press, 2003.

Han, Jiawei; and Micheline Kamber. *Data Mining Concepts and Techniques*. San Diego, California: Academic Press, 2001.

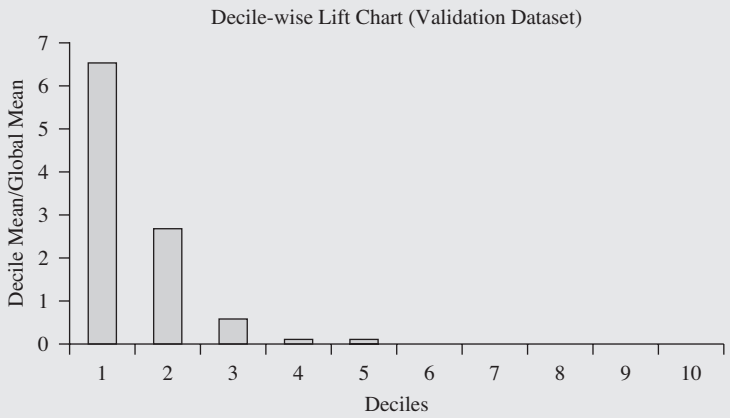
Shmueli, Galit; Nitin R. Patel; and Peter C. Bruce. *Data Mining for Business Intelligence*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2007

Witten, Ian H.; and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Amsterdam: Elsevier, 2005.

Exercises

1. A data mining routine has been applied to a transaction dataset and has classified 88 records as fraudulent (30 correctly so) and 952 as nonfraudulent (920 correctly so).

   The decile-wise lift chart for a transaction data model:



   Interpret the meaning of the bars in this chart.

2. Which of the following situations represents the confusion matrix for the transactions data mentioned in question 1 above? Explain your reasoning.

A

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	58	920
0	30	32

B

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	32	30
0	58	920

C

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	30	32
0	58	920

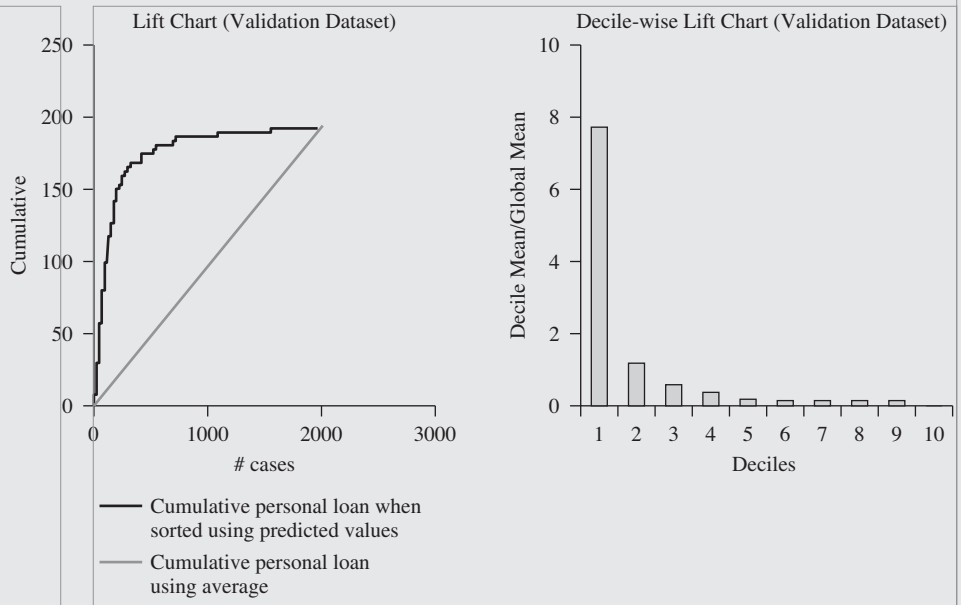
D

Classification Confusion Matrix		
	Predicted Class	
Actual Class	1	0
1	920	58
0	30	32

3. Calculate the classification error rate for the following confusion matrix? Comment on the pattern of misclassifications. How much better did this data mining technique do as compared to a naive model?

	Predict Class 1	Predict Class 0
Actual 1	8	2
Actual 0	20	970

4. Explain what is meant by Bayes' theorem as used in the Naive Bayes model.
5. Explain the difference between a training data set and a validation data set. Why are these data sets used routinely with data mining techniques in the XLMiner© program and not used in the ForecastX™ program? Is there, in fact, a similar technique presented in a previous chapter that is much the same as partitioning a data set?
6. For a data mining classification technique the validation data set lift charts are shown below. What confidence in the model would you express given this evidence?



7. In data mining the candidate model should be applied to a data set that was not used in the estimation process in order to find out the accuracy on unseen data; that unseen data set. What is the unseen data set called? How is the unseen data set selected?
8. Explain what the "k" in the k-Nearest Neighbor model references.