Chapter 7

Evaluation of Dimensions

Chapter Outline

What You Will Learn in This Chapter

- How do some attributes or variables influence other attributes?
- When do you need to evaluate dimensions?
- What type of data can be used and how should it be structured?
- How does linear regression work and why would you use it?
- How can attributes for discrete Y-values be estimated?
- How do you begin an analysis when you know little about the data?
- Is there a way to organize the attributes to see how they explain the decision?
- How can the modeling process be automated to handle nonlinear relationships?
- How do the model results compare to each other?

Union Pacific

Union Pacific is the nation's largest railroad, operating 900 trains a day. Although many people are no longer familiar with railroad technology, a key element in safety and performance is derailments, which can cost millions in damages and lost time. In the early 2000s, Union Pacific installed acoustic and visual sensors on the undersides of its rail cars. The sensors have focused on imminent problems with the track and wheels—notably the status of wheel bearings; and have reduced bearing-related derailments by 75 percent. The acoustic sensors identify an average of three cars a day and send the warning information to engineers who move the cars offline for repair at the earliest opportunity. Around 2010, the company installed visual sensors (cameras) as well—to help identify flattening of wheels and other problems. More recently, the company developed predictive software that analyzes the data from acoustic and visual sensors for patterns. Lynden Tennison, Union Pacific CIO, noted that the system can analyze 40 million patterns a day and alert engineers of a potential problem within five minutes. [Hickins 2012].

Predicting events is better than waiting for bad things to happen.

Michael Hickins, "Union Pacific Using Predictive Software to Reduce Train Derailments," *The Wall Street Journal*, March 30, 2012. http://blogs.wsj.com/cio/2012/03/30/union-pacific-using-predictive-software-to-reduce-train-derailments/

Introduction

How do some attributes or variables influence other attributes? Data mining ultimately comes down to this question. In fact, the question is the heart of any discipline from business to medicine to science and philosophy. Ideally, people can study data, build models, and develop theories that explain how changes in some attributes will cause changes in other variables. But, **causality** is a tricky subject. It is difficult to use statistics to prove causal changes. Most of the time, statistics can evaluate **correlation**—two things that move together. When the sun comes up in the morning and the temperature increases, does that mean the sun caused the temperature increase? Probably, and we have a physics model to explain it. When the sun comes up in the morning and traffic in the city increases, does that mean the sun causes traffic jams? Not exactly but there is a chain of events that can explain the relationship. If the sun comes up in the morning and the number of deaths per hour decreases at a local hospital, did the sun cause it? This last correlation presents a much harder problem to answer. The point is that if data items are correlated, the relationship raises questions that point analysts in a direction to look for explanations. Items that are not correlated are less interesting—so statistical data mining that identifies relationships and shows where they do not exist can be a powerful tool for exploration.

This chapter examines several data mining tools that can be used to evaluate how attributes affect other dimensions. Once relationships are identified, the results can be used for classification and prediction. Classification in the sense that values for the attributes can divide an outcome variable into groups. For instance, certain characteristics of age, income, and education might divide customers into heavy purchasers and weak purchasers. For instance, perhaps older, higher income, people with more education tend to be heavier purchasers of items. Prediction is used when managers and analysts want to generalize results and plug in values of attributes to see what might happen. For example, if the firm increases marketing to attract higher-income customers, how much might sales increase? Specifically, will sales and profits increase enough to compensate for the added marketing costs?

The primary tools for this type of study are: (1) linear regression, (2) logistic regression, (3) naïve Bayes, (4) decision trees, and (5) neural networks. With all of these tools, the analyst identifies an outcome or predictable Y-attribute that managers want to observe or forecast. The goal is to examine a set of X-attributes that might conceivably explain movements in the predictable variable. The differences in the tools are primarily in the techniques used to estimate correlations and create predictions. There is one main exception. Linear regression requires that the dependent variable consist of continuous data, not discrete or categorical observations. The other techniques are geared towards categorical outcomes, although in some cases they can handle continuous data as well.

This chapter looks at each of the five tools and identifies the goal of the tool with a brief look at the methodology. The methods are described in mathematical terms, but the coverage is light and designed to highlight the challenges and applicability of the tool. Deeper mathematical and programmatic discussions of the tools are found in many other books and Web sites. The goal of this chapter is help you understand the tool so you can use it effectively. Each section also describes the data needed. The last four tools are all demonstrated using the same data set from the Rolling Thunder Bicycle Company. The methods are demonstrated us-

ing the versions in Microsoft BI. In the case of linear and logistic regression, the methods are also demonstrated using more traditional tools. These two methods have a long history, and the Microsoft BI results are slightly different from the traditional approaches. It is good to know both methods. The basic results are presented from the application of the methods and they are examined for insight into attribute evaluation and prediction. At the end, the four major discrete-evaluation tools are compared to see what information can be learned for the sample problem.

Business Situation

When do you need to evaluate dimensions? The essence of data mining is to focus on an outcome or fact variable. Common business examples include sales, profit, whether a loan will be repaid. As a manager, you want to know how other variables or dimensions affect the outcome. How sensitive are customers to price? If you change the color of a product or its packaging, how many more units can you expect to sell? Or, what happens if you change a production process—such as switching to organic ingredients—will sales increase? Or, perhaps the cost function is complex and no one completely understands what happens to costs as production increases.

Another classic example from the banking industry involves the question of which customers will be able to repay loans. What characteristics or dimensions are the most important: salary, job tenure, savings balances? Are there tradeoffs among the dimensions: If someone has a low salary, how large does the savings account need to be to compensate?

These types of questions involve **classification** and **prediction**. Classification in the sense that you want to know which customers fit into defined categories (successful loans, big spenders, and so on). Prediction is used when you encounter new data you want to be able to predict the eventual outcome. Essentially, you are estimating a **model** of the relationships between the dimensions and the outcome variable. In many cases, linear relationships are useful because they are easy to estimate and easy to understand. These are often estimated using linear regression. More complex systems require nonlinear relationships. These can be estimated with neural networks, but the results can be more difficult to interpret.

Model

How do you know which model to use? How do you know which attributes to choose? The analyses in this chapter are guided by the analyst. In the basic case, you must select the dependent variable and the list of independent variables to be examined. You also select a tool to estimate the model, and the choice of the tool often imposes a structure on the model. For example, using linear regression restricts the model to simple linear relationships. It is possible to minimize the restrictions—particularly by choosing a neural network tool to estimate the relationships—but your role in determining the model is important.

Figure 7.1 shows two ways to present a simple model: (1) graphically showing the explanatory attributes affecting the dependent variable (Sales), and (2) as a linear equation, where the goal is to estimate the coefficients (b-terms). The equation approach is compact and it is sometimes easier to understand. The equation is typically estimated with a **linear regression** tool.

Figure 7.2 shows a portion of a different type of model. A **decision tree** is sometimes used to create a model to analyze the data. A decision tree handles discrete (categorical) data as well as continuous data. Even the dependent variable

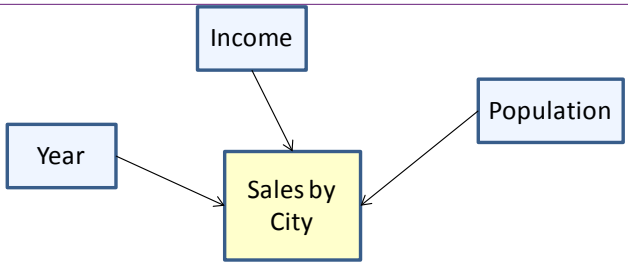


Figure 7.1

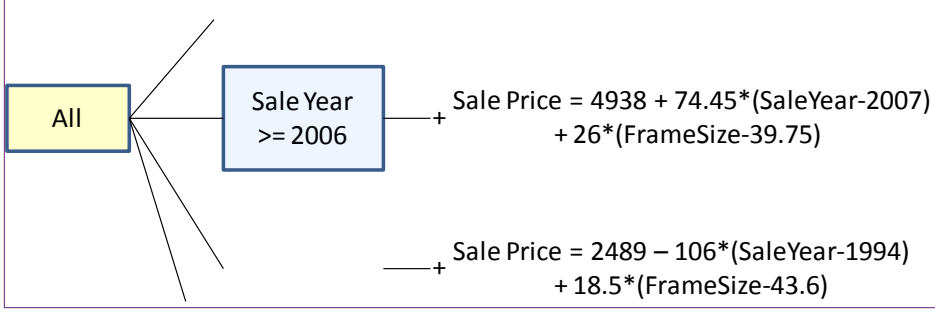
Basic linear model. Sales by City the dependent variable are likely affected by the per capita Income and the Population of the city. Sales also probably increase over time, so Year is included as an explanatory variable. The model can handle only numeric data.

can be discrete data. Only a tiny portion of the model is shown in the figure. Decision trees often grow quite large. Each node represents a split point where values for an independent variable have a different effect on the outcome. In this small example, the Sale Price is affected by the Sale Year and the Frame Size. This portion of the tree shows that prices are affected differently for years before 1996 and after 2006. Other year segments are also different but not shown. Marked by the plus sign (+) the tree can be expanded to the right to see the effect of other attributes—notably Frame Size. Decision trees can be complex, but they provide a visual way to help you explore the impact of various attributes on the chosen dependent variable. They are also good for finding interactions among the attributes. In the example, the effect of Frame Size varies depending on the year.

One of the goals of data mining is to reduce the need of the analyst to specify models. This approach is both good and bad. It makes it easier to explore many possible effects without knowing too much about the underlying data. But, without specifying a model in advance, you impair the ability to make statistical tests. The results you obtain might not be statistically useful. Explorations can be useful when you are beginning your study of the data.

Figure 7.2

Decision tree model. The model handles numeric and categorical (discrete) data. It splits the attributes into sections that have an effect on the dependent variable. Each node represents a split point in the data.



If you do need to specify a model ahead of time—and in most cases you at least need to come up with potentially important attributes—you need to look to experts who have studied the problem. In many cases, business disciplines including economics, accounting, finance, and marketing have developed models that can be used to examine problems. For instance, economics identifies several factors that affect the demand for products. So sales and price should be influenced by attributes such as customer income, prices of other products, and expectations about the future. Economics and accounting have developed models to estimate costs, including fixed costs and per-unit or marginal costs. Marketing models identify many attributes that affect consumer choices. Finance has developed complex models that evaluate financial products and risk. In other areas, you need to turn to different scientists, including biological, growth, and DNA models. If you can find a model that someone has already developed and validated, it will be much easier to identify the important attributes and test their importance with your data.

The main point to remember is that to evaluate and understand how various attributes affect an outcome variable, you need to know enough about the subject to choose the appropriate attributes. Once you know the types of data, you then need to choose the appropriate modeling or mining technique that matches the data. These concepts are covered in the next sections.

Data

What type of data can be used and how should it be structured?

The data needed for evaluating dimensions is similar to that used in the cube browser: A fact or outcome measure and a set of dimensions that you want to examine. It is helpful to think in terms of the traditional regression model, where the data consists of observations or rows of data. One observation is essentially a row of data consisting of values for each of the dimensions and the corresponding attribute you wish to predict. The prediction attribute is sometimes called the **dependent** or endogenous variable; the other factors are **independent** or exogenous variables.

Figure 7.3

Standard data layout. A column represents a single dimension attribute. Each row is one observation, such as a purchase or a customer. Typically, you will need a key column attribute that identifies each row.

CityID	Pop	Income	Year	Sales
4650	48950	34004	1994	2700
14438	2049	23020	1996	2430
4943	13064	31282	2001	2680
17664	2662	45101	2004	2420
342	16844	23153	2007	3910
4586	8491	39605	1997	1790
2625	7173	32016	1997	2630

Attributes and Observations

For all of the tools in this chapter, the data is most often viewed as a table—where the columns represent the attributes and each row consists of one observation. Figure 7.3 shows sample data for a problem with independent attributes on each city (population, per-capita income, and the sale year). Total sales value to customers in that city for a given year is in the Sales column. Some tools allow the use of multiple dependent variables. For example, you might add sales volume (count the number of items) to this example. However, most standard data mining tools treat multiple predicted columns as separate problems. Statistically, you gain little by combining them into one problem. It might be slightly more convenient to list multiple dependent variables, but you need to be careful in the estimation process. You must also be careful when you explain the results and remember that even if the two variables are logically related, the process treats them separately. In general, it is safer to use a single dependent variable and run separate analyses. However, most advanced econometric problems have multiple dependent variables—and multiple equations. These simultaneous equation models require special estimation methods that are not handled by typical data mining tools.

Microsoft BI requires that every dataset must include a key column that uniquely identifies each row. This column does not need to be used within the analysis, but it ensures the rows are identifiable. An important catch is that only one column can be used—not composite keys consisting of multiple columns. If you do not have a single-column key, you can create a new column in the dataset that combines the values from the columns you do have. For example, if you want analyze sales by customer by year you start with two key columns: CustomerID and Year. In the data view, you need to create a new column:

```
Cast(CustomerID as nvarchar) + '-' + Cast(Year as nvarchar)
```

This expression creates a new column that contains both the CustomerID and the Year, separated by a hyphen. For example: 192-2009. The *Cast* function is needed because the data must be converted to text (nvarchar). Without the cast, SQL Server would simply add the two numbers numerically.

The structure of the data helps you focus on the goals of this type of analysis. You choose the dependent variable that you want to understand or predict. Then you select various attributes that you think might affect that variable. You collect observations of cases and they become the rows of the table. The data mining tools examine the rows and use them to estimate weights or values that have an impact on the dependent attribute.

Continuous and Discrete Data

As you will see in the following sections, one of the key issues affecting your choice of tools is the type of data available. In particular, it is important to know if each attribute contains numeric continuous data, discrete numbers, or even categorical data. For example, regression tools require numeric data, but decision trees can handle categorical values. Figure 7.4 summarizes the data types that can be handled by the tools covered in this chapter. Notice that the regression tools are the most constrained. Decision trees are flexible, but it can be harder to interpret the results. Neural networks are also flexible, but some features are only available when you use continuous data. Also, the specific requirements can vary depending on the specific version of the tool. Some vendors are less flexible than others. The requirements listed here apply to the Microsoft BI tools.

Tool/Method	Dependent	Independent
Linear Regression	Numeric: continuous	Numeric: continuous Numeric: discrete
Logistic Regression	Numeric: discrete	Numeric: continuous Numeric: discrete
Decision Tree Naïve Bayes	Numeric: continuous Numeric: discrete Categorical	Numeric: continuous Numeric: discrete Categorical
Neural Network	Numeric: continuous* Numeric: discrete Categorical	Numeric: continuous* Numeric: discrete Categorical

Figure 7.4

Data requirements for tools. Some tools require numeric data. The dependent variable is often the most critical—particularly with regression tools. Decision trees are generally the most flexible. *Neural networks typically support most data types but some options require continuous numeric data.

Because each tool has slightly different results and interpretations, you need to consider the results you want to see as you select the data. If you find that a specific tool should be used for its interpretation, you might need to recode the data. For instance, you could recode categorical data into numbers (e.g., Female=1, Male=2) if you want to use regression, which requires numeric data. Some tools can automatically recode categorical data into discrete numbers. Conversely, many tools convert continuous data into discrete groups, sometimes called **discretized data**. However, the tools in this chapter generally work better by leaving continuous data in its original form.

Missing Data

Missing data or null values can cause problems with some methods. In particular, tools that rely on continuous data do not support missing values. Also, all the attributes must have the same number of rows. Regression (linear and logistic) are the most restrictive of the tools in this chapter. If possible, the data should not contain missing values. If so, the most common approach is to remove those rows from the analysis. An alternative is to substitute values for the missing entries, such as computing the mean value of an attribute and using it to replace missing values. But, replacing values can alter the results, so be cautious.

Tools that support discrete values can tolerate null values. The missing value simply becomes another discrete case. For example, a gender attribute might contain *Female*, *Male*, and null values. Hence, the tool will evaluate three cases instead of two. So, if your data has a large number of null values, you should consider estimating decision trees or neural networks instead of regression equations.

Linear Regression

How does linear regression work and why would you use it?

Linear regression is one of the oldest data analysis tools. Its theory, properties, complications, and applications have been studied for many years. It is a powerful statistical tool often used in research. When applied statistically, it is a fundamen-

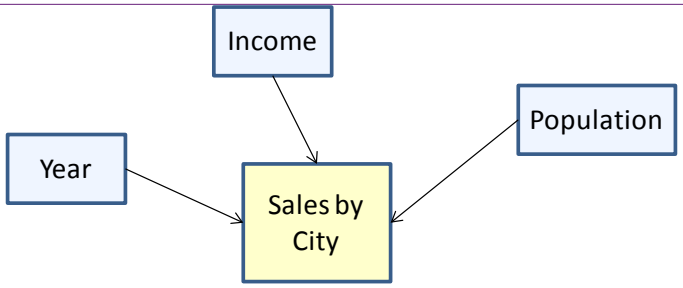


Figure 7.5

Linear regression results. The tool estimates the values of the coefficients which represent the strength of the association between each attribute and the dependent variable. The equation can be used to predict new values by plugging data into the independent attributes.

tal tool of science that is used to prove or disprove hypotheses and to test models. However, when used for exploratory purposes, such as most data mining projects, the statistical validity often disappears. Although the tool is the same, the selection of the data and the way it is handled often violate the theoretical foundations of the regression statistics. In particular, for statistically valid results, the data must meet certain conditions—which are typically obtained through random samples. Also, you would have to start with an underlying model that can be tested. Repeatedly trying combinations of attributes and multiple models on the same set of data changes the statistical value of the results.

Because this book focuses on data mining and exploration, the details of the underlying statistics are not covered here. The reference section at the end of the chapter lists a couple of classic econometrics textbooks that provide details of how to use regression techniques appropriately in a scientific setting. But, you are probably not searching for scientific truths if you are using data mining. Instead, you are trying to see how various attributes might influence an outcome variable. Later, you can determine if you need to build a complete model, collect more data, and conduct formal tests.

If you know ahead of time that you will want to conduct a more scientific study, you should conduct your data mining on only part of the available data. Hold some of the data for use later—a random selection of rows might be useful.

Goals

Regression is commonly used to determine how various attributes influence a selected dependent attribute variable. Linear regression estimates the coefficients of a linear model where each attribute has an independent effect on the dependent variable. These coefficients reveal the impact of each attribute. They are also easy to use when predicting the outcome of new data. Figure 7.5 shows the basic result using three independent attributes. The objective of linear regression is to find the coefficients that best fit a linear relationship. Linear relationships are useful because they are easy to understand. Also, they are relatively robust in the sense that small changes in the data usually have only a small impact on the coefficients.

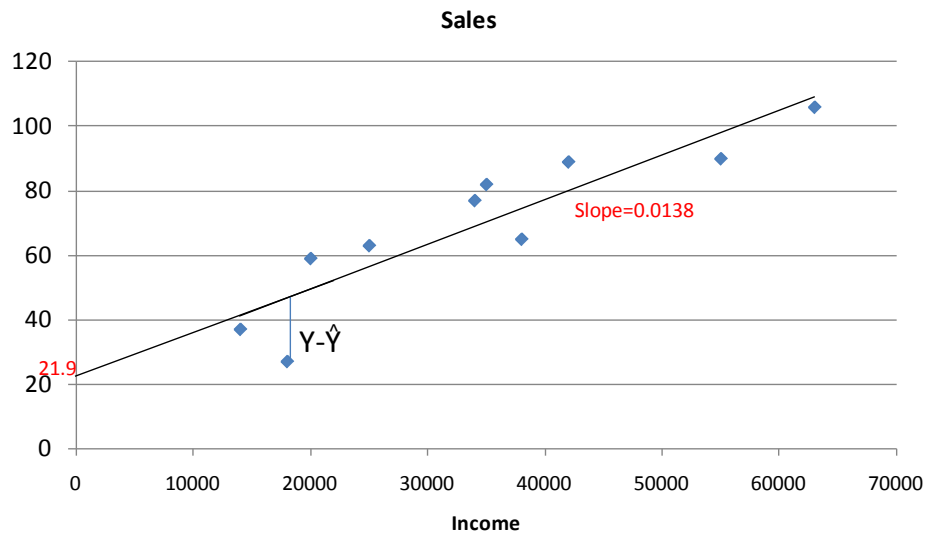


Figure 7.6

Linear regression: fitting a line. A line can be defined by its intercept and slope. These two coefficients are found by minimizing the sum of the squared error.

Figure 7.6 shows the regression process in more detail—using only one independent attribute. The objective of linear regression is to find the line that best fits the observed points. Best fit is usually defined as finding the line that minimizes the sum of the squared error—or the distance between the desired line and a given point. Squared error is important because the deviation could be positive or negative and squaring forces it to be a positive value. Historically, squared values were chosen for two reasons: (1) The result is mathematically easier to analyze, and (2) Outliers or values that are far away from the line have a stronger impact. Today, with fast computers, some systems have the ability to minimize the mean absolute deviation—which uses the absolute value instead of the squared value of the difference. You might use absolute values if you need to decrease the impact of outliers—such as when you believe the outliers are strange or unlikely values. Mathematically, there are some relatively fast methods to find the linear coefficients. Some tools use this direct approach, others use a broader search method. Either way, the tools always report the values of the coefficients.

But, what do the coefficients mean? The best way to understand the value of the linear regression coefficients is to apply a little bit of math. From the two-dimensional case, it is clear that the coefficient on the X-attribute is the slope of the line. This interpretation is true for multiple coefficients as well. Recall that the slope of a line is:

dy/dx = change in y / change in x

Or, in words, the slope coefficient answers the question: If the x-value increases by one unit, what happens to the y-value? Consider a simple sales example where

the income coefficient equals 0.0138. It seems like a small number, but income is evaluated in thousands. So, if the firm can attract customers who make 1000 more than the existing customers, how much will sales increase? The answer is:

slope*1000 = 0.0138*1000 = 13.8

That value might not seem like much, but if the average sales value is 69.5, it would represent a 30 percent increase in sales. The point is that you can use the coefficients to predict what will happen if the underlying attribute values change. The coefficients also reveal which attributes are more important than others, so you know where to focus your efforts if you want to change an organization.

Regression coefficients are found by minimizing the sum of squared errors. In matrix terms, the outcome Y variable consists of a matrix of n rows and 1 column. The X observations form a matrix of n rows with k columns of X-attributes. Usually, the first column values are 1 to handle the intercept constant. The errors are written as the actual y value minus the estimated value. The estimated value is XB, where B is the 1 x k matrix of linear coefficients. Then, the goal is to find the B values that minimize:

$$e'e = (y-XB)'(y-XB) = y'y - BX'y + B'X'XB$$

Differentiating the squared error term with respect to the B values to find the minimum point leads to the optimal value for B:

$$B = (X'X)^{-1}X'y$$

Now, as a data analyst you do not need to memorize the formula for finding the coefficients. But, it does make a couple of things clear. First, finding the linear coefficients entails computing the covariance matrix X'X and then finding its inverse. Several computer algorithms exist to find the inverse efficiently. But, some problems can arise—such as the covariance matrix not having full rank—meaning it is not invertible (similar to the issue of not dividing by zero with constants). This situation most commonly arises when the X-attributes chosen are dependent on each other (linear combinations).

Data

Regression is a powerful tool, but it is somewhat restrictive in terms of the types of data that can be analyzed. One of the most critical constraints is that the dependent variable must contain continuous numeric data. This data should not be constrained or truncated (e.g., greater than zero, less than 100). Econometricians have devised methods to handle some typical issues with data, but the common data mining or BI tools do not implement most of these changes. High-end (expensive) statistical packages implement these tools, but you probably need an experienced analyst to configure the data and interpret the results.

Data for independent attributes must be numeric, but it can be discrete. Categorical data can be recoded into numbers. Depending on the source of your data, you can use queries (such as a CASE statement) to convert discrete text values to numbers. Or, within a data mining tool, you can create a new calculated variable to handle the conversion. SQL Server uses the same CASE statement in queries and as an expression within a data view.

Figure 7.7 shows the use of the CASE statement in a SQL Server SELECT statement. You simply specify the comparison column (Gender) and then list each of its possible category values in a WHEN clause. The corresponding THEN state-

```
SELECT CustomerID, Gender,
CASE Gender
  WHEN 'F' THEN 1
  WHEN 'M' THEN 2
  ELSE 0
END As NewGender
FROM Customer
```

CustomerID	Gender	NewGender
1	F	1
2	M	2
3	M	2
4	M	2
5	M	2
6	M	2
7	M	2
8	F	1
9	M	2

Figure 7.7

Using SQL Server CASE to recode data. The CASE statement can be used in a SELECT query to convert text values in the Gender column into numeric values that could be used in a regression model. You need to list each category in a WHEN clause, using the THEN statement to define a unique number to the category.

ment converts the text value to the specified number. Note the use of the ELSE statement to handle missing or invalid data. This new column NewGender could then be used in a regression analysis.

Figure 7.8 shows the same CASE statement being used to define a new Named Calculation within a Data Source View in the data mining tool. Notice that the CASE statement is identical to the one used in the SQL query. It is often helpful to test these conversions in a query first so you can run the query and correct any problems that arise as you are developing the formula. The basic steps to create a named calculation in the SQL Server BI tool are:

1. Open the data source view.
2. In the Design screen, right-click the Customer table.
3. Choose New Named Calculation.
4. Enter the values shown and click OK.

After you create the expression in the data mining tool, you can right-click the table and browse it to see the converted values to ensure the expression is working correctly. You should always verify your work as you go. If you make a mistake in the CASE statement, it can be very difficult to find later. Worse, you might never see the error, run the regression analysis and reach the wrong conclusions because the data was bad.

Missing data is usually not allowed in regression analyses. Some tools tolerate columns with missing data, but they generally resolve the problem by discarding the entire observation. Because discarding the row is usually the best option, this approach is acceptable. Hopefully, the tool will warn you about missing data so you can go back and verify that you have included the correct data. In some cases, you might want to discard an attribute if it has too many missing values. If only one attribute is causing most of the problems, you are generally better off without that column—because it will cause the system to discard useful data from

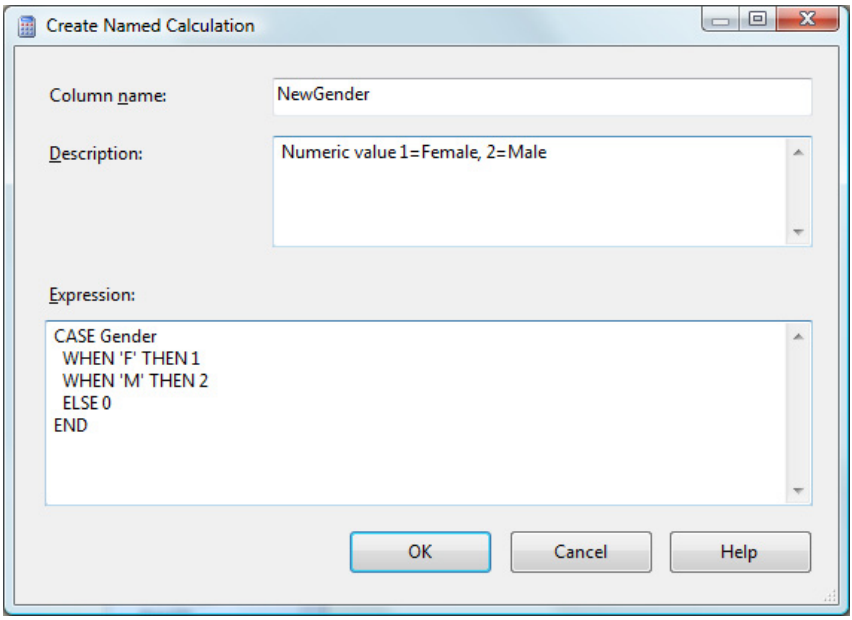


Figure 7.8

CASE statement in data mining tool. The CASE statement can be used as an expression to recode text values in the Gender column into numeric values that could be used in a regression model. You need to list each category in a WHEN clause, using the THEN statement to define a unique number to the category.

the other attributes. For example, you might have an Age column that you want to use to predict sales. But, customers are typically reluctant to reveal their age, which leads to have many missing data points. If the Age attribute is included in a regression model, all data associated with that customer will be discarded. The results will be based only on the data from customers who did report their ages—which is probably not a representative group, and could be quite small. When you encounter missing data, you will have to decide whether to discard the rows or the specific column—based on the number or percentage of missing values; and on how important the column might be to your exploration.

Tools

Many tools exist to perform linear regression. Microsoft Excel even has a built-in regression tool under Data/Data Analysis. (Although, you probably have to install the Analysis Toolpak Add-in to find it.) Regression is also a common feature of most statistical packages. These packages provide detailed control over the analysis, can handle complex data problems, and provide detailed results and evaluation statistics. However, as external packages, you would need to find a way to integrate or convert your data to the statistical software.

Microsoft BI does support linear regression—but it uses a unique computational approach. In particular, it relies on decision-tree methodologies to evaluate each attribute specified in the model. Consequently, although Microsoft BI eventually does produce estimates of the equation, the results might not exactly match those generated by other tools. Additionally, statistical packages provide more detailed

```
SELECT City.CityID, Population2000, Income2004,
        YEAR(Bicycle.OrderDate) As SaleYear, SUM(SalePrice) As TotalSales
FROM City
Inner Join Customer
        ON City.CityID=Customer.CityID
INNER JOIN Bicycle
        ON Bicycle.CustomerID = Customer.CustomerID
WHERE Population2000 Is Not Null
        And Income2004 Is Not Null
GROUP BY City.CityID, Population2000, Income2004, YEAR(Bicycle.OrderDate);
```

Figure 7.9

SQL query to get sales by city. Note the use of the Bicycle, Customer, and City tables. The YEAR function returns just the year portion of the date. SUM computes the total sales. The query returns the total sales to each city along with the income and population of the city, plus the year of the sale.

information about the estimates. Still, it is usually more convenient to use the tools provided within the data mining system. To understand some of the differences, results from both SQL Server BI and traditional regression are covered in this section.

Traditional Least-Squares Regression

Consider the traditional tools first. Organizing the data properly is the first step to using these tools. Typically, data has to be in a flat file of simple columns and rows. Each column represents a single attribute. Each row contains one observation. The Rolling Thunder Bicycle Company presents a useful example. The managers are thinking about expanding marketing in specific cities. Because the company sells high-end bicycles, the managers think that wealthier consumers are probably a better target. But, they also suspect it is important to target larger cities. In browsing the database for the company, you might have noticed the City table—which lists thousands of U.S. cities along with their population (from the 2010 Census) and per-capita income (from 2009 Census data). The Bicycle table makes it easy to find the sales to each customer, and each customer is linked to the City table. It is probably a good idea to include the year of the sale to control for any trends over time. To obtain the data, you create a basic query that retrieves the total sales by city, along with the population and income columns. You can obtain the sale year by using the YEAR function.

Figure 7.9 shows the SQL query. The YEAR function returns just the year portion of the date and SUM computes the total. Notice the GROUP BY clause needs the CityID, plus the population, income, and year attributes. This query produces the total sales by city and year and it drags along the population and income values for each city. Each row represents sales for one city and one year. To run the regression analysis, you need to first run the query and copy or export the data. For small and mid-sized problems, it is relatively easy to copy the data from the query and paste it into an Excel spreadsheet.

The Data/Data Analysis/Regression tool in Excel is a quick way to estimate linear regression coefficients. It does not support missing data and the independent (X) attributes must be in contiguous columns (a

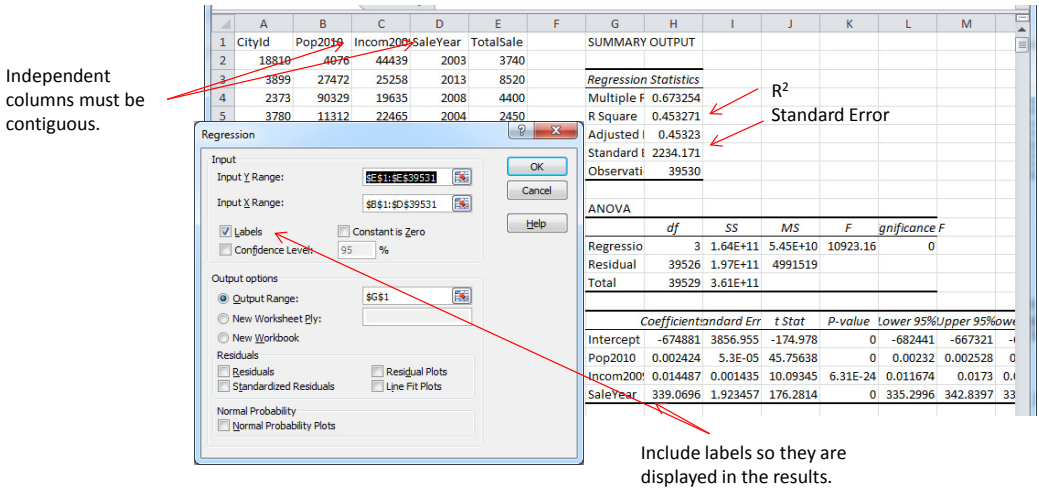


Figure 7.10

Linear regression in Excel. Copy-and-paste the data from SQL Server into a new worksheet. Choose Data/Data Analysis/Regression from the menu. Select TotalSales as the Y (prediction) variable. Select the population, income, and Year columns as the X range. Check the labels option and run it.

single group with no column gaps). Figure 7.10 shows the basic setup. Simply pick the Y (predictable) column, then the X (independent) columns. Set the Labels checkbox to ensure that the column names are displayed in the output. Choose where you want the results to be displayed and click the OK button to run the regression analysis. The results in this case were displayed on the same worksheet. The coefficient values are displayed in the table. The results include additional useful information and these values are described in the Results section.

Microsoft Data Mining Linear Regression

Instead of the well-known least-squares approach, Microsoft data mining uses a specially-configured version of its decision tree tool to estimate linear coefficients. This approach automatically attempts to break the data into a decision-tree framework as well. However, this section focuses on the linear regression approach; the decision tree issues are covered in a later section. A critical aspect to Microsoft's approach is that the data set must contain a single key column that identifies each row of data. This requirement arises because of Microsoft's approach—it is not a factor in regression analysis. But, you must be extremely careful to set this column correctly. It affects the estimation results.

Setting up the data for linear regression is a little tricky. Most traditional linear regression analyses need data at a single level—in one table or query. Additionally, you might need to compute subtotals before running the regression analysis. Hence, it is usually easiest to create a named query to get exactly the data you need. A **named query** is simply an SQL statement stored within a data source view. It is often used to compute subtotals, concatenated key columns, and select or exclude data to be analyzed. It uses the power of SQL to choose exactly the data needed. To illustrate, consider the bicycle case where the goal is to identify

```
SELECT  CAST(dbo.Customer.CityID AS nvarchar) + N' -' + CAST(YEAR(dbo.Bicycle.
OrderDate) AS nvarchar) AS CityYear,          dbo.City.Population2000,
          dbo.City.Income2004, YEAR(dbo.Bicycle.OrderDate) AS SaleYear,
SUM(dbo.Bicycle.SalePrice) AS SaleTotal, dbo.Customer.CityID
FROM   dbo.Bicycle
INNER JOIN  dbo.Customer
          ON dbo.Bicycle.CustomerID = dbo.Customer.CustomerID
INNER JOIN dbo.City
          ON dbo.Customer.CityID = dbo.City.CityID
WHERE (dbo.City.Population2000 IS NOT NULL) AND (dbo.City.Income2004 IS NOT
NULL)
GROUP BY CAST(dbo.Customer.CityID AS nvarchar) + N' -' + CAST(YEAR(dbo.Bicycle.
OrderDate) AS nvarchar), dbo.City.Population2000,    dbo.City.Income2004,
          YEAR(dbo.Bicycle.OrderDate), dbo.Customer.CityID
```

Figure 7.11

A named query to compute subtotals for sales by city and year. Notice the use of the Cast statement to create a unique key that includes both the CityID and the Year. Also, missing data for income and population are excluded.

attributes that affect sales by city. This approach requires the total sales to each city for each year, plus it needs the population and per capita income. Technically, it would be nice to have population and income for each year; but that data is difficult to obtain.

Named queries are created within a data source view. If you do not have one already, you should create a new data source view that includes at least the Bicycle, Customer, and City tables. Then add a new named query. Figure 7.11 shows the SQL for the named query. Notice the use of the Cast function and concatenation to obtain a key column (CityYear) that uniquely identifies each row. The rest of the query is standard SQL for computing subtotals using the GROUP BY statement. You can test the query as you build it to ensure that it works and retrieves exactly the data needed.

Once the data is available, you can right-click Mining Structures in the Solution Explorer and create a new data mining structure. Choose the linear regression option and pick the data source view that holds the named query you created. Select that query as the Case table. Figure 7.12 shows the selection of the attributes. Ensure that CityYear is selected as the key column that identifies the data. Set SalesTotal as the predictable column. For Income, Population, and SaleYear, check the boxes to make them input attributes. Finish the wizard's steps with the default values and give it a name you will remember later (perhaps Sales by City regression). Notice that 30 percent of the observations will be held out to use for testing the model. If you want to come closer to a traditional regression package, you can set this value to zero so that all rows are used in the estimation.

Once models are defined in Microsoft's analysis project, you have to deploy and process them. Then you can browse the results. Right-click the mining structure in the Solution Explorer and choose the option to Process the model. Follow the steps to Run the process and close any windows it opens. Right-click the mining structure again and choose the option to Browse the results. This simple model should evaluate to a single level. The coefficients and the regression equation are displayed in the bottom-right corner of Visual Studio when you select the node.

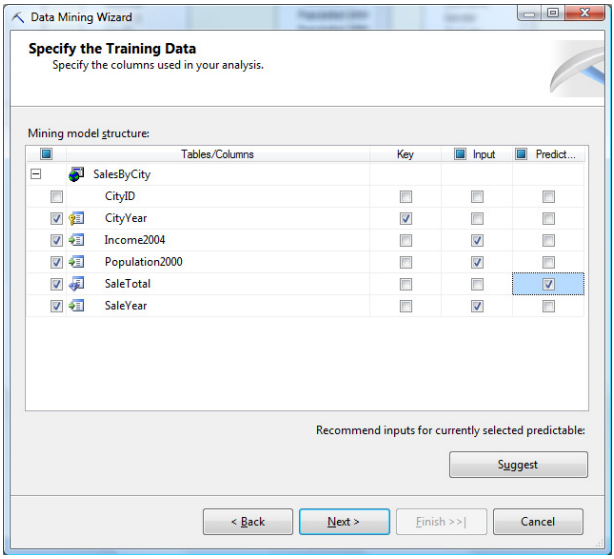


Figure 7.12

Selecting attributes for regression. The key column must uniquely identify a row. The predictable column is the column you want to predict (SaleTotal). The other columns, population, income, and SaleYear are simple inputs.

Results

It is relatively easy to set up and run data mining tools. The real key lies in interpreting the results. By understanding the model it becomes possible to analyze problems and predict possible outcomes based on that model. Technically, linear regression measures the correlation between variables, and correlation does not mean causation. That is, without a formal model and explanation of the underlying effects, it is possible that the results you see are simply one-time correlations of events that happened at the same time. Business and economic models are weaker than physical models. The results can reveal how various attributes move together with a dependent variable, but that does not guarantee the correlation will always exist. It does not mean that when the value of an independent variable is changed that the exact effect will always occur.

Traditional Regression Results

Figure 7.13 presents the results from the traditional regression (e.g., Excel). The three most important elements are: (1) The coefficient values, (2) The T-statistics that evaluate each coefficient, and (3) the R^2 value which indicates the accuracy of the overall model. The values shown in this figure came from the Bicycle sales by city and year, but results might vary depending on which data was used.

The coefficient values are important because they are the slope or change in the Y value that will occur for a one-unit change in the input X-value. In the example, sales increase by 339 each year. An increase of 1,000 in Income increases sales by 14.35. You have to pay attention to the units of the underlying data. Yes, 339 appears to be much larger than 0.0145, but Income is generally discussed in thousands. Although a $1 increase results in less than a penny increase in sales, a $1,000 increase of per capita income is associated with a $14.35 increase in sales.

```
Sales = --674881 + 0.002424 Population + 0.014487 Income + 339.07 Year
          (-175.0)  (45.76)               (10.09)          (176.28)
R2 = 0.4533
```

Figure 7.13

Traditional regression results. Regression provides an estimate of the coefficient values, the T-statistic which is a measure of a coefficient's accuracy, and R^2 which is an overall measure of the accuracy of the model.

Along the same lines, just because a number looks big does not mean it is significantly different from zero. In statistics terms, the coefficient value is a mean. And means are estimations that are subject to error. However, least-squares regression also provides a measure of the standard error of the coefficient. Dividing the coefficient by the standard error gives the T-statistic for the coefficient. This value is used in a simple hypothesis test to see if the coefficient is significantly different from zero. Loosely, if the T-statistic is larger than 2 (in absolute value), then the coefficient is significantly different from zero. Conversely, T-values less than that (e.g., 1.00) indicate that the variation is too high and the coefficient is effectively zero. Hence, the attribute has no important effect on the dependent variable. Technically, T-statistics are evaluated by a T-test which incorporates the degrees of freedom; but with a sufficiently large number of observations, 2 is approximately the value of the critical T-statistic for a test with 5 percent error.

The R^2 value can be interpreted as a percentage of the variation in the Y-values that is explained by the independent X variables. The number ranges from 0 to 1, with 1 representing a complete explanation and a perfectly straight line. Low values indicate that your choice of X variables explains only a small percentage of the variation. How low is low, or how high should the values be? It depends on the problem. In the example, 26 percent seems a little low, but there is a lot of underlying variation in the data. It is unlikely that you will be able to find a better model. Still, you could look to find other attributes that might improve the model. Low R^2 values are indicators that your model might be missing some key attributes.

Taken together, the statistical results of linear regression provide useful information about a process. You can see which attributes have strong (or no) effects on the dependent variable. You have actual measures of the correlation, which you can use to predict future outcomes. And you have a measure of the strength of the model helping you decide if you have examined the proper set of attributes and the linearity.

Microsoft BI Results

Because Microsoft's linear regression tool is based on its decision tree method, it produces somewhat different output than found in least-squares regression. Figure 7.14 shows the results from the linear regression data mining tool. Although not shown, the tool found a single node, which means the coefficients found for the equation should be close to those from traditional regression.

First, notice that the coefficients are presented two ways: In a table and as an equation. The values are the same—except for the intercept term. In the equation, each independent variable is written as a deviation from its mean. The standard intercept value can be computed by plugging in zeros for each of the variables, or it can be read from the top row of the table.

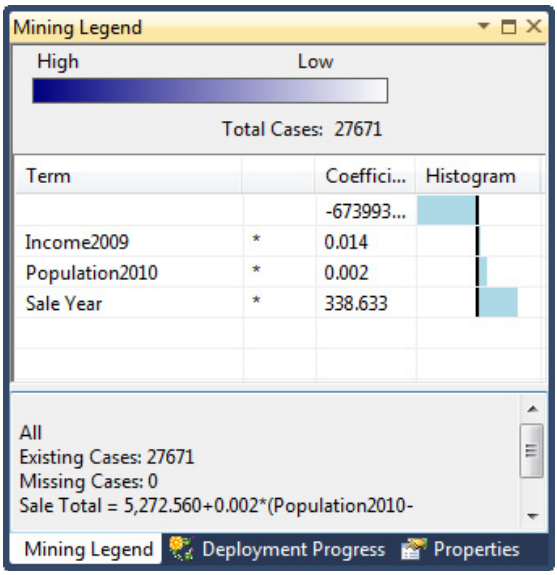


Figure 7.14

Microsoft regression results. With a single node, the decision-tree-based approach returns coefficients similar to those of a traditional regression. But, the tool make its own decision about significance of variables and does not display the standard errors or the R2 value.

Second, observe that the coefficients are close to the values estimated with traditional regression—but not exactly the same; The reason for the difference in the values is because the data mining tool held out some of the data to be used to test the model. The least-squares approach used all of the rows. Because the holdout data was randomly selected, the results will be similar but not exactly the same.

More importantly, the tool does not report standard errors or the R^2, so it is harder to determine the importance of the variables and the model. Instead, Microsoft generates some accuracy charts that use the holdout data set to evaluate the model. These options appear in most of the data mining tools. Click on the Mining Accuracy Chart tab to begin the process. After verifying the input choices (test cases), you can select the Lift Chart and Cross Validation tabs. Because linear regression uses a continuous variable the Lift Chart tab actually shows a scatter chart comparing predicted to actual values. The values are taken from the holdout dataset and the predicted values are computed from the regression equation coefficients. Figure 7.15 shows the scatter plot for the example. In a good model, the points will cluster around the 45-degree line where predicted equals the actual value. Here, the model appears reasonable at low levels but it seems to be truncated and non-linear so that at the higher levels, the predicted values are consistently low. For values less than 10,000, the results appear to be clustered around the equality line. However, the model does not predict well for values greater than about 15,000. The model appears to truncate its predictions and has trouble forecasting larger numbers. Most likely, the model needs an additional attribute or perhaps the coefficient on population is too low. It is possible that the population effect is nonlinear.

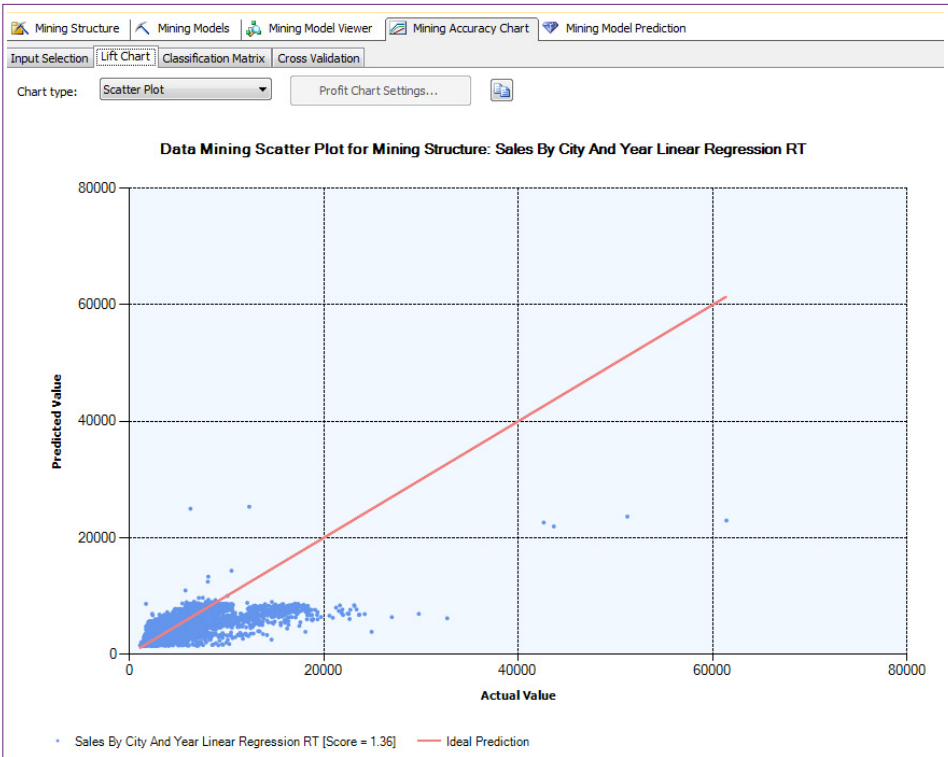


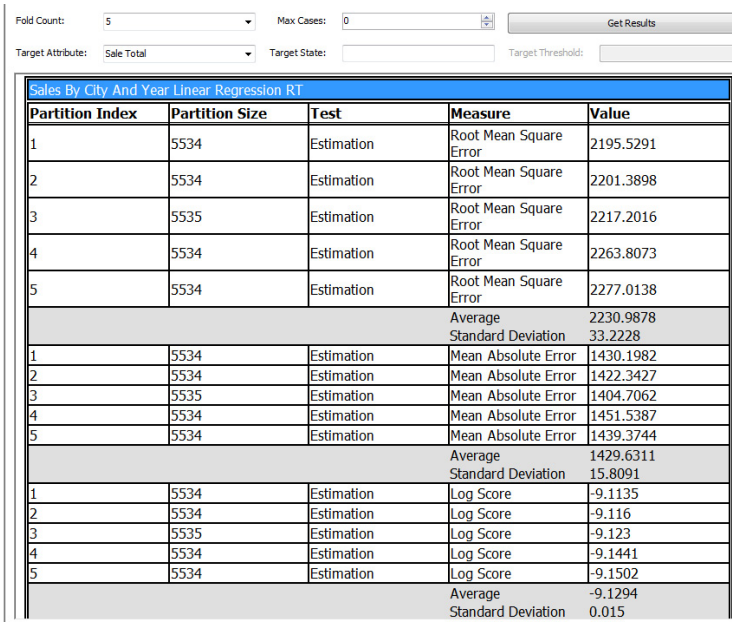
Figure 7.15

Mining accuracy chart. Scatter chart of predicted v. actual Y values taken from the holdout dataset. The model is good if the values are clustered along the 45-degree line (predicted = actual). This model appears to be reasonable at low levels but seems to struggle at higher levels of sales.

Microsoft's data mining tool also uses a cross-validation tool to evaluate the overall performance of a model. The tool has more options for a discrete Y-attribute, but it does have a way to evaluate continuous data used in linear regressions. Essentially, cross validation splits the training data set into partitions or folds. Each partition represents one test—the data from all other partitions is used to build the model, which is then applied to the test data and evaluated. The goal is to determine if a model is susceptible to a specific set of data. By building and testing on different pieces of data, models that work on one segment of data but not others will be easy to spot.

In the Mining Accuracy Chart tab, switch to the Cross Validation tab. Set the Fold Count (number of partitions) to 5, set the Max Cases to 0, which uses all of the non-holdout data. The Target Attribute should already be set to the dependent Y-variable (Sale Total).

A big problem that can arise in data mining is that any model estimated is heavily dependent on the specific data chosen. When you have enough observations, you should hold back some of the data to use for testing. Additionally, Microsoft provides a cross validation tool. Cross validation splits the training data into partitions or folds (ignoring holdout testing data). Each partition will represent one test of the model. Iterating through each partition, the system uses the data in the



Sales By City And Year Linear Regression RT

Partition Index	Partition Size	Test	Measure	Value
1	5534	Estimation	Root Mean Square Error	2195.5291
2	5534	Estimation	Root Mean Square Error	2201.3898
3	5535	Estimation	Root Mean Square Error	2217.2016
4	5534	Estimation	Root Mean Square Error	2263.8073
5	5534	Estimation	Root Mean Square Error	2277.0138
			Average	2230.9878
			Standard Deviation	33.2228
1	5534	Estimation	Mean Absolute Error	1430.1982
2	5534	Estimation	Mean Absolute Error	1422.3427
3	5535	Estimation	Mean Absolute Error	1404.7062
4	5534	Estimation	Mean Absolute Error	1451.5387
5	5534	Estimation	Mean Absolute Error	1439.3744
			Average	1429.6311
			Standard Deviation	15.8091
1	5534	Estimation	Log Score	-9.1135
2	5534	Estimation	Log Score	-9.116
3	5535	Estimation	Log Score	-9.123
4	5534	Estimation	Log Score	-9.1441
5	5534	Estimation	Log Score	-9.1502
			Average	-9.1294
			Standard Deviation	0.015

Figure 7.16

Cross validation results. Data is split into the number of specified partitions (folds) and the model is estimated and tested in each partition. The error terms should be the same across the partitions.

"other" partitions to estimate the model coefficients. It then computes error statistics using the data in the targeted partition. So, you end up with overlapping tests of the model—one test for each partition. If the error measures are the same in each partition, then the model is considered robust and not susceptible to quirks in the data. If you see major differences in the error measures across partitions, then your model is highly dependent on specific data and you should look for a better model—or at least additional attributes.

Figure 7.16 shows the cross validation results using five partitions. Note that Max Cases specifies the number of observations to use—entering 0 tells the system to use all of the training rows. For linear regression, **root mean square error (RMSE)** is the most common measure. It is computed as the square root of the mean (average) squared error. The formula is almost the same as that used to compute the least squares estimates:

$$RMSE = \sqrt{\frac{1}{n}\sum\left(Y - \hat{Y}\right)^2}$$

In fact, least-squares regressions return this value as the standard error. Check Figure 7.10 to see the value of 2234 computed by Excel. The average of 2230 computed by cross validation is close to that value. The cross-validation tool also reports the **mean absolute deviation (MAD)** which uses the absolute value to compensate for negative values instead of squaring the errors:

$$MAD = \frac{1}{n}\sum \text{abs}\left(Y - \hat{Y}\right)$$

The important result in both cases is that the values in the three partitions are almost identical. Technically, you could use the reported standard deviation to conduct simple T-tests if the numbers appear to be different. The conclusion is that the model does not depend on specific groups of data. There is still a chance that the overall data set is uniquely strange, but if it is representative of the population of data, the model should perform consistently in most cases.

Both error measures also convey information about the accuracy of the model. If the measures were zero, the model would have no errors and the data would fit a line perfectly. If you have two or more models using the same data, the model with the lower RMSE is the better model. There is no fixed value of RMSE that is considered good or bad because the number depends on the variation in the Y variable. Technically, the R^2 value can be computed from the RMSE and the sum of squared Y-values (1 – sum of squared error/sum of squared Y), but if you want R^2 just use least-squares regression to compute it for you.

Attribute Evaluation

The R^2, RMSE, and coefficient values are all indicators of the stability of the model. If the R^2 is reasonable and the results show that at least some of the coefficients are significant and have signs that make sense, the coefficients tell you which attributes have important influences on the dependent attribute. The coefficients passing the significance test are the critical ones. These are the items to concentrate one. Pay attention to the signs. A negative sign means increases in the attribute level will decrease the dependent variable. Also, look at the attributes to find ways that the values can be changed.

If an attribute is completely outside of your control, it can be important, but you might not be able to do anything about it. In the example, the sales year is a strong factor, and per capita income of the city is important. Population of the city is also an important factor for increasing sales. At first glance, it might appear that all three variables are outside of management control. The coefficient on the Year attribute simply states that sales for next year depend heavily on what they were in the prior year—for a given city. It is not possible to change the year, but if sales can be increased in the current year, that pattern should carry forward into succeeding years. In other words, increased efforts at selling today will also influence purchases in the future.

City population and per capita income seem even more distant. While it is true that you cannot change either of those values for a given city; you can choose which cities you want to focus on for a marketing campaign. The answer is to focus on larger cities with higher per capita income.

One issue you need to watch when looking at attribute coefficients is the magnitude (size) of the coefficient. A value of 339 on the Year attribute appears much larger than 0.0145 on the Population coefficient. Yet, the size of the coefficient depends on the values used in the attributes. It is often helpful to use a trick from economics and use an elasticity measure instead. **Elasticity** is the percent change in the dependent variable divided by the percent change in the input variable. It indicates how much the dependent variable will change for a one percent increase in the independent variable.

Attribute	Coefficient	Average	Elasticity
Sales		5272.92	
Year	339.07	2004.66	128.908
Population	0.00242	29505	0.014
Income	0.01449	25060	0.069

Figure 7.17

Elasticity. Percent change in Y divided by percent change in X is the slope coefficient times the average X value divided by the average Y value. Elasticity is a pure number that does not depend on the units of the attributes, which makes it easier to compare across attributes.

$$Elasticity = \frac{percent\ change\ in\ Y}{percent\ change\ in\ X} = \frac{\%\Delta Y}{\%\Delta X} = \frac{\Delta Y / Y}{\Delta X / X} = slope \frac{X}{Y}$$

Because the regression coefficient is the slope, elasticity is relatively easy to calculate—particularly at the average point. Simply multiply by the average value of the X and divide by the average of Y. Elasticity is a standalone number and the values can be compared across all of the attributes.

Figure 7.17 shows the computation of elasticity for the sample problem. Looking at the elasticity column, the year is the most important factor—although years can only increase in one-unit amounts, so a percentage increase is hard to understand. The elasticities for population and income are more interesting. Both values are low, but sales are about five times more sensitive to increases in income than to increases in population. So, any marketing campaign should focus on wealthier cities. Note that the effects are additive, so by increasing marketing to cities that are one percent over average in income and in population, sales should increase as a percentage by the sum of the two numbers.

Prediction

Regression is a key tool for predicting outcomes. The estimated linear relationship makes it relatively easy to predict values for the dependent variable for any combination of input attributes. Simply multiply the attribute values by the respective coefficients and add up everything. Make sure to include the constant intercept value. The result is an estimate of the resulting Y value. It is also possible to compute the variance of any predicted value.

$$s_{\hat{y}}^2 = [\hat{x}(X'X)^{-1}\hat{x}']s_e^2$$

The value is written using standard matrix notation for regression. The X matrix consists of rows of data for the independent attributes, with a first column set to 1 for each row to handle the intercept. The smaller consists of the single row of data values to be forecast. And is the square of the standard error of the regression. The resulting variance enables you to compute a confidence interval for the estimated value. A confidence interval is useful because it provides a range of potential values instead of a single point. Values outside the interval are highly unlikely

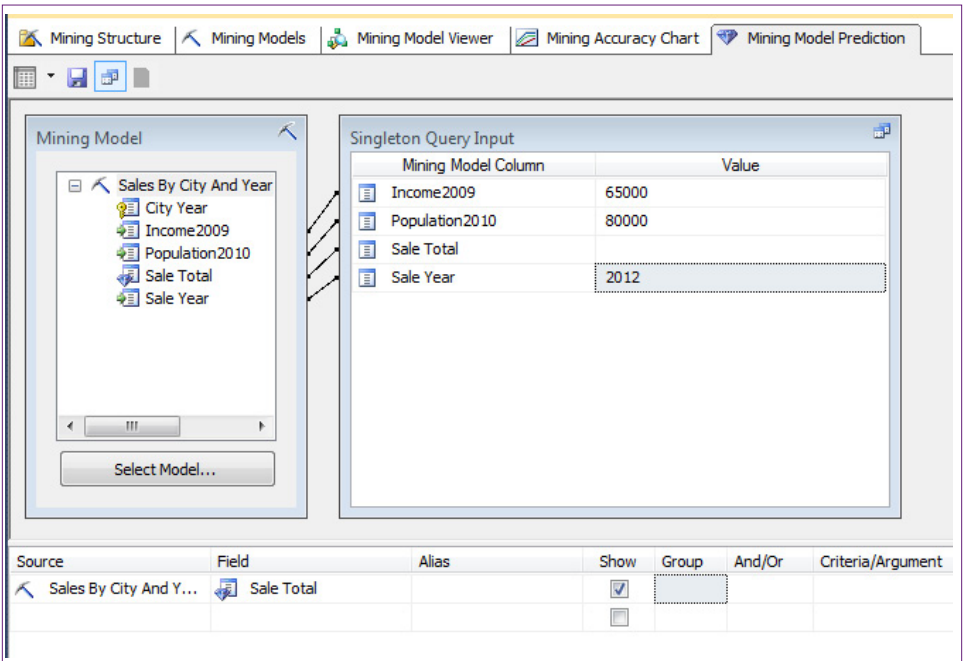


Figure 7.18

Prediction with singleton query. Choose the mining model and click the Singleton query icon. Enter the x-attribute values you want to predict. Choose the mining model (Sales By City) in the Source row and the Y-attribute (Sale Total) as the field to be predicted. Switch to result view to see the predicted total (8446.5).

to occur. For example, a predicted value might be 6924, and a 95% confidence interval might be (6886, 6962). The confidence interval provides a great deal of useful information, because it identifies the most likely range for the prediction. On the other hand, if the confidence interval were (6545, 7304) because of a large standard error, the forecast essentially would be meaningless. In the example, because of the relatively high standard error, the actual confidence interval is (6848, 7000), which is a relatively wide interval, making it difficult to have much confidence in the forecasts.

From the formula, it is clear that the interval depends on the standard error of the regression. If the overall regression has a large error, any individual prediction will have high errors and a wide range. However, the formula reveals that is relatively difficult to estimate the variance of the predicted value. Fortunately, most statistical regression tools provide an option to compute and display both the forecast value and its variance, and often can display the confidence interval as well.

Microsoft BI has the ability to compute predictions—based on a set of X values. Data attributes can be entered into a sample table to be run as a group, or you can switch to a singleton query (toolbar icon) and enter just one row of data directly. Figure 7.18 shows the basic prediction pane for a singleton query. Enter the values of the X-attributes, then pick the mining model (Sales By City) in the Source column. Select the Y-attribute to be predicted (Sale Total). Switch to the Results view to see the computation value (8446 in this case). Notice that the tool

does not compute the standard deviation or confidence interval for the value. With some effort, it might be possible to compute the standard error using the formula and MDX commands. But, ultimately, if you want the statistical error results, it is better to use a complete statistical package.

Logistic Regression

How can attributes for discrete Y-values be estimated? Linear regression is a powerful tool and has many extensions. For instance, it is relatively easy to handle polynomials simply by computing squared, cubic, or higher powers of the X attributes. It is also easy to use discrete X attributes by simply numbering the alternatives. But, linear regression always requires a continuous dependent (predictable) Y variable. Otherwise the results will be biased and predicted values will be difficult to compute correctly. The **logistic regression** method was created to handle discrete Y data. Technically, the binomial logit handles Y values of zero or one, while multinomial logit handles Y data with multiple discrete values. Binary values are common in problems where you are interested in whether some event happens or if a person or object falls into a specific class or not. For instance, if a person defaulted on a loan, or purchased any items, or paid with cash. Multinomial situations arise when an event has several discrete outcomes, such as five choices on a survey, payment methods (such as cash, credit, debit, or check), type of bicycle, or a specified category range (such as one-time, casual, common, or frequent customer).

Goals

The problem is solved by estimating a function that determines the probability that the specified event happens. This probability is based on the independent X attributes. For instance, a marketer wants to know the probability that a person with a certain income, gender, and age makes a purchase. The marketer has observations on many people of various genders, ages, and income groups; as well as the outcome of whether a purchase was made. Early approaches tried to estimate a simple linear regression:

$$P = b_0 + b_1X_1 + b_2X_2 + b_3X_3$$

However, probability values must lie between zero and one and the linear regression forecast could generate almost any value for P. The solution is to transform the data to ensure that it stays between zero and one—regardless of the b and X values. Two common functions are the logistic and cumulative normal distribution. The logistic function is used most often because it is mathematically much simpler:

$$P = \frac{1}{1+e^{-x}}$$

Figure 7.19 shows the logistic function. No matter what value of Z is chosen, the dependent value always lies between zero and one. The function is monotonic (increasing values of X always lead to increasing values of Y), so it does not distort the input values. In terms of estimating probabilities as a function of input attributes (X), the Z variable is written as a linear combination of the attributes:

$$Z = b_0 + b_1X_1 + b_2X_2 + b_3X_3$$

Intriligator [1978] shows another way to approach the problem that results in the same logistic solution. Begin by looking at the odds ratio or the odds in favor of an individual event, but use the natural logarithm:

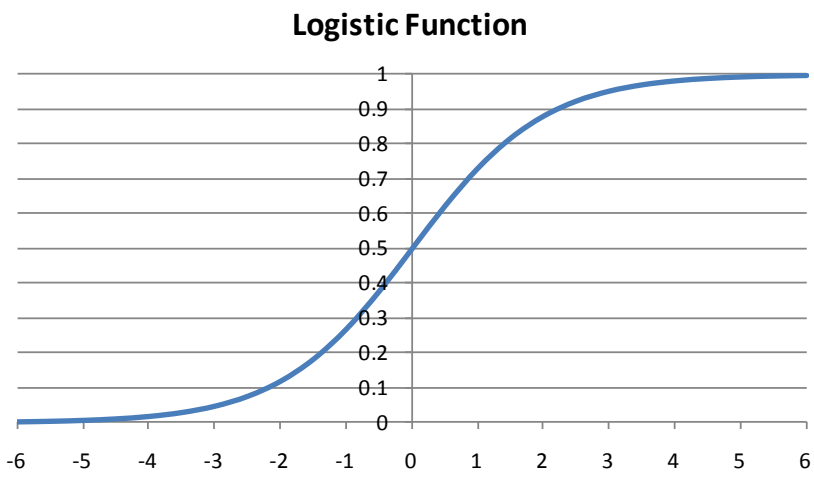


Figure 7.19

Logistic function. For any X values, the logistic function always returns values between zero and one, making it useful to estimate probabilities.

$$\log(\frac{p}{1-p}) = b_0 + b_1 X_1 + b_2 X_2 \ldots = Z$$

Taking the antilog (exp) of both sides and solving for P leads to the logistic function. Not that some algebra is involved, but the approach yields two useful points for understanding logistic regression. First, in the binary (0/1) case, it is possible to estimate the b coefficients by using the log-odds ratio as the dependent variable in a simple linear regression. Second and more importantly, in a traditional logit regression, the values computed from the sum of the linear terms results in the log of the odds ratio in favor of decision occurring. That is, once the b coefficients have been computed, it is easy to determine the odds (e.g., 5-to-1) that a customer with a given set of attributes will make a purchase. However, Microsoft BI does not use traditional logit regression to find the coefficients, so this relationship does not work when using that tool.

When the dependent variable consists of multiple choices, the mathematics and interpretation is similar. For example, managers of Rolling Thunder Bicycles might want to determine which attributes affect the purchase of each model type. Essentially, the traditional problem is written as a collection of binary dependent variables: y1 is set to 1 if a Race bike is purchased, y2 is set to 1 if a Road bike is purchased, and so on. However, this approach leads to estimation of separate equations for each value of the choice variable. In the bicycle example, the results will consist of separate sets of b coefficients for each model of bicycle. Judge [1985] shows how to compute the probability of each outcome. Each outcome j generates a set of B_j coefficients. An individual i has a set of X_i attribute values. To keep the equations easier to read, X and B are matrices, but you can think of the product as just the sum of the multiplications. Then the probabilities of a specific individual selecting outcome j are given by:

$$P_{i1} = \frac{1}{1 + \sum_2^J e^{X_i B_j}} \qquad P_{ij} = \frac{e^{X_i B_j}}{1 + \sum_2^J e^{X_i B_j}}$$

The computation for the first probability is slightly different because the choices are normalized by using differences and assuming that B_1 is zero. There is no value in memorizing these results, but they make it easier to understand the purpose of the regression. Essentially, the estimation of the B coefficients leads directly to the estimate of the probabilities of selecting each of the outcome choices. Typically, you can simply let the regression routine compute the probabilities for you in a prediction.

The key goal is that logistic regression estimates the B coefficients which indicate the importance of each attribute and the equations provide the ability to predict the probability that various choices will be made—based on different values of the X attributes.

Data

Data for logistic regression is similar to that for linear regression—with the difference in the dependent predictable variable. The outcome to be predicted should be a discrete or categorical variable. It can be binary or it can contain multiple choices. In some cases, you can use a continuous dependent variable—by converting it into discrete groups or categories. BI tools, including Microsoft, provide options to discretize the data. However, in most situations, the data should already be in logical categories.

With traditional tools, all of the data will have to be numbers. If necessary, you can recode or use the SQL CASE command to assign numeric values to categories. As an example, consider the Rolling Thunder Bicycle Company with a goal of identifying attributes that affect the sale of the various model types. Model Type will be the dependent variable; independent attributes can consist of Gender, and SaleYear, and city characteristics of population and income. The data will have to be combined into simple columns and rows for export to an external tool. A basic query can be used to combine the data from the three tables: Bicycle, Customer, and City. As shown in Figure 7.20, the query can also compute SaleYear from the OrderDate and assign numbers to the ModelType and Gender variables.

Data for Microsoft logistic regression is simpler because it handles categorical data automatically so recoding is not needed. Ultimately, the tool is easiest to use if all of the data is collected into a simple named query that holds just the columns and rows needed for the analysis. The sample problem uses the Bicycle, Customer, and City tables. If you do not already have one, you need to create a data source view and add the three tables. Then create a named query, call it BikeCityDetail, and use SQL to join the three tables and retrieve at least SerialNumber, Population2010, Income2009, Gender, and ModelType. To check for changes over time, you should also compute a SaleYear column as Year(OrderDate).

Why choose these particular columns? The ModelType will be used as the outcome or predictable column. The others are the only available attributes that might influence the decision of choosing a model type. Additional attributes might be considered, such as SalePrice, EmployeeID, and StoreID. Conceivably, some employees or retail stores might push certain models harder than other ones. Price might be an issue if there are large differences in prices across models. The goal is to select attributes that might have an effect on the outcome; and that might be

```
SELECT  Bicycle.SerialNumber,     City.Population2010/1000 As Population,
        City.Income2009/1000 As Income,
        YEAR(Bicycle.OrderDate) AS SaleYear,
                CASE
                        WHEN ModelType='Race' THEN 1
                        WHEN ModelType='Road' THEN 2
                        WHEN ModelType='Track' THEN 3
                        WHEN ModelType='Tour' THEN 4
                        WHEN ModelType='Mountain' THEN 5
                        WHEN ModelType='Mountain full' THEN 6
                        ELSE 0
                END AS NModelType,
                CASE
                        WHEN Gender='F' THEN 1
                        WHEN Gender='M' THEN 2
                        ELSE 0
                END AS NGender
FROM    Bicycle
INNER JOIN Customer
        ON Bicycle.CustomerID = Customer.CustomerID
INNER JOIN City
        ON Customer.CityID = City.CityID
```

Figure 7.20

Converting categorical data to discrete numbers. The SQL CASE statement makes it easy to assign numbers to categorical data so that the data can be exported and used by external tools.

controllable in some fashion or at least helpful in explaining the decisions of consumers. But, for a first try, it is helpful to keep the list short so the results do not become overwhelming.

Tools

High-end (expensive) statistical packages perform multinomial logit regression without too much effort. Some open-source econometrics packages, such as gretl, support logit (and probit) analyses. In any case, you would need to run the data query to generate numerical attributes, export the data to a format that can be read by the package, then configure and run the analysis within that system. Remember that the systems will return separate estimates of the B values for each outcome variable. These equations can be used to estimate the probability of a particular choice occurring.

Microsoft logistic regression, and often other BI tools, uses a different method to handle logistic regression. It turns out that many neural network systems use a logistic function to transform data for the same reason—to limit output values between zero and one. Consequently, the systems use decision trees and simplified neural networks to search for the best solution. As shown in the next section, the results are useful but often more fragmented. Consequently, the probability interpretations are altered.

Once the data view and named query have been set up, it is straightforward to configure Microsoft's logistic regression. Start a new mining structure and choose

Model	Constant	Pop	Income	Year	Gender
Race	-245.6**	0.00027**	0.0058	0.124**	-0.199**
Road	-225.8**	0.00013	-0.0062	0.114**	-0.203**
Track	1967.4**	0.00007	-0.0015	-0.986**	-0.273
Tour	-111.1**	0.00009	-0.0099*	0.056**	-0.251**
Mountain	5.20	0.00010	-0.0055	-0.001	-0.174**
Mountain full	-212.8	0.00014	-0.0072	0.108**	-0.205**

Cases correctly predicted: 30.3%

Figure 7.21

Logistic results. The asterisks mean that the coefficient is significantly different from zero at the 5 percent level. All but one are actually significant at a 1 percent error level.

the logistic regression tool. Select the appropriate data source view and pick the named query as the Case table. Choose the attribute columns to match the problem: SerialNumber is the key column because the data is arranged by bicycle. ModelType is the Predictable column. For now, stick with a short list of Input (X attribute) columns: Gender, SaleYear, Income2009, and Population2010). Finish the wizard with the default values and give the model a name that you will recognize later.

Results

Results for logistic regression are different for traditional tools versus specific data mining systems, particularly Microsoft's. The difference is not necessarily bad, but some interpretations are harder to obtain. Both approaches do a good job of evaluating the contribution of the independent X attributes. The traditional model is easier to convert to probabilities, while the BI tools are more focused on comparing the effects of the attributes.

Traditional Logistic Regression

Figure 7.21 shows a set of results for the basic logit regression. Observe that one key to understanding the results is to compare the coefficients across the different choices. To keep the coefficients reasonable, the population and income values were converted to thousands. The resulting coefficients are still small. First note that population does not appear to influence the choice of model type, except for Race bikes. Second, income appears to affect only the Tour models. The lack of significance in the population variable seems unusual but it means that customers from almost any size city might purchase any model type. The one significant income effect is negative indicating a slight tendency for customers from lower-income cities to purchase the tour model. The tour bike makes some sense because it tends to be cheaper than race bikes.

In the example, a critical difference in the years is the negative sign on the Track bikes. If you look at the data, it will show that Track bikes sales were discontinued after 1995. The Gender term is instructive. Remember that it is coded as 0 for unknown/missing, 1 for female, and 2 for male. The relatively higher

Hybrid	NULL	9
Hybrid	F	396
Hybrid	M	601
Mountain	NULL	346
Mountain	F	2791
Mountain	M	4381
Mountain full	NULL	965
Mountain full	F	4554
Mountain full	M	7556
Race	NULL	877
Race	F	3743
Race	M	6483
Road	NULL	756
Road	F	3303
Road	M	5660
Tour	NULL	180
Tour	F	848
Tour	M	1370
Track	NULL	3
Track	F	28
Track	M	42

Figure 7.22

Bikes sold by ModelType and Gender. Notice that men buy more of every type. The null values coded as zero might cause problems. Rerunning the logistic regression without customers with missing gender removes the statistical significance from the Gender attribute for all except the full suspension mountain bikes.

coefficients on mountain bikes make sense, but the lower values on the track and tour bikes seem strange. Including the null values might cause some problems. However, it needs further investigation.

A quick way to investigate the situation is to run a GROUP BY query or go back and browse the data cube to examine total number of bikes sold by ModelType and Gender. Figure 7.22 shows the result. Notice that men buy more of all model types. However, the statistical results control for the other variables (population, year, and income), and indicate that after compensating for those effects, women buy more than men. Still, the result appears unusual—perhaps it is a quirk in the data. Another potential problem is the missing values for gender. Some customers used only a first initial and did not indicate a gender. Although those numbers seem low in most categories, it might affect the results. The logistic regression can be re-run after filtering out the customers with those missing values. The results are not shown here, because the coefficients are similar to those in the original regression. However, most of the Gender coefficients are no longer significantly different from zero. Gender remains negative and significant only in the case of full suspension mountain bikes. Intuitively, that is an unexpected result, but it might make sense—it could indicate that women want mountain bikes that are more comfortable and easier to ride on harsh terrain.

For this book, the conclusions regarding bicycle sales are unimportant. The key is to understand how to examine the initial results, look for patterns, and begin interpreting the results. When you see unusual or interesting items, use the tools to drill down and examine the underlying data. Be cautious regarding missing values.

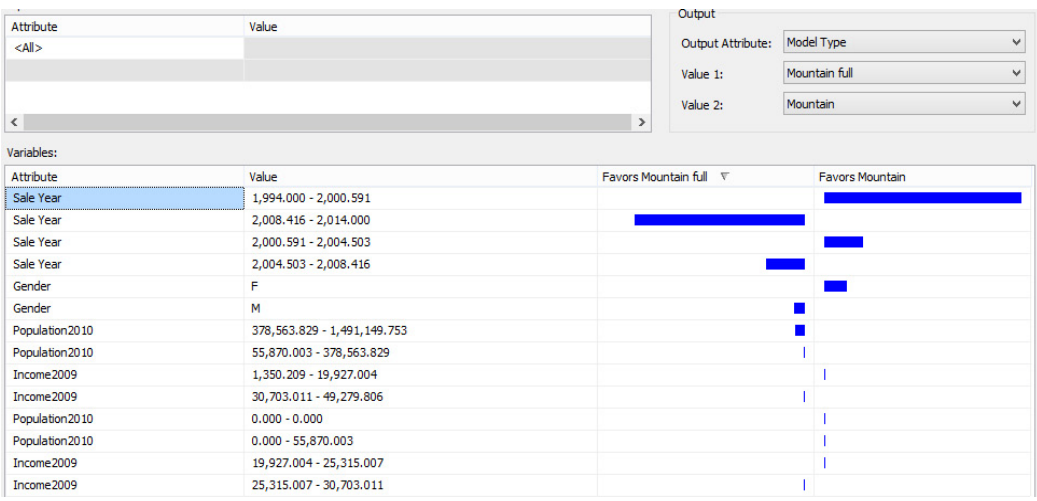


Figure 7.23

Microsoft logistic regression results. The results are shown as comparisons between any two items that you select. This comparison between mountain and mountain full suspension bikes shows the year break when full suspension was introduced. It also shows small effects for income, population and gender.

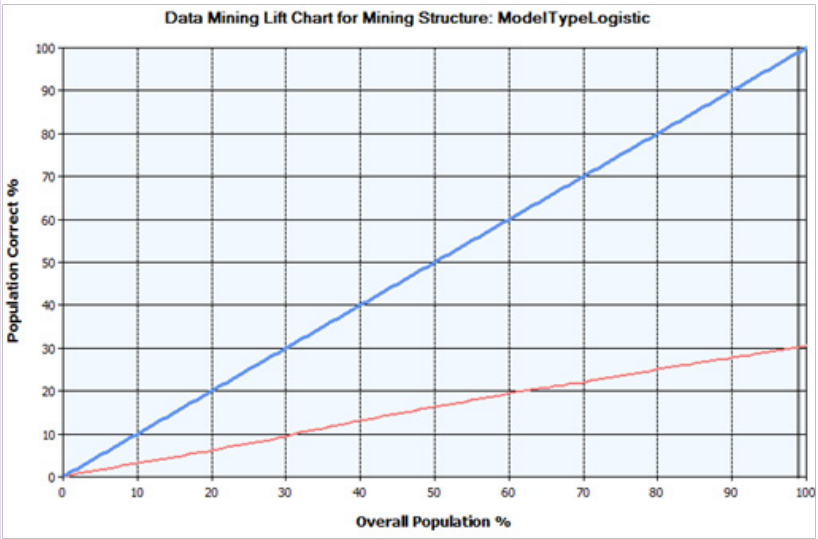
Microsoft Logistic Regression

Microsoft's logistic regression is different from the traditional approach, so the results and interpretation are quite different. The key difference is that the logistic regression uses a decision tree approach; which automatically incorporates interaction effects. With this approach, the system examines the independent X attributes for breakpoint values that will alter the decision path. Yet, like the traditional logistic regression solution, the results depend on the individual model choices.

The results are easiest to understand with an example. However, as you will see, the results are difficult to present in print. Microsoft's presentation results are designed for interactive use. As shown in Figure 7.23, you select any two items from the dependent attribute (ModelType). The side-by-side comparison investigates the question of why consumers choose one model over the other. In this example, the importance of the Year shows that when full suspension bikes were introduced, consumers almost immediately switched to them. However, the gender effect does appear; along with tiny effects for city income and population (larger, wealthier cities lean towards full suspension). However, these city effects are small so they might arise simply because of the large number of sales of full suspension bikes (skew effect).

Note that Microsoft's approach automatically handles the missing gender values as a separate case. Rolling the mouse over one of the bars in the chart causes the system to report the underlying data for that item. In the case Gender for females (F), it reports 36 percent favor full suspension compared to 15.4 percent for mountain bikes. The percentage represents the probability of this range (female) for the specified outcome (full suspension or mountain).

The system also reports the lift for each value. In this situation, Microsoft defines **lift** as the impact of using this particular variable for predicting the speci-



Series, Model	Score	Population correct	Predict probability
Bike City	0.30	30.2%	24.8%
Ideal Model		100.00%	

Figure 7.24

Mining Accuracy Chart. For logistic regression, move the vertical gray bar to the far right (100%). The legend table reports the number of cases classified correctly (30.2%), and a 25% probability that any given person will behave according to the model.

fied outcome. For females, the lift is computed to be 1.1 for full suspension and 1.0 for mountain bikes. These are relatively low values. Compare them to the lift for a sale year from 2008-2014, which yields 1.24 for full suspension and 0.55 for mountain bikes. In terms of developing the model and evaluating the attributes, it is much more important to include sale year than to include gender in the model. The size of the bars in the chart are based on a score computation that measures the effectiveness of the model using normalized data. Essentially, the data attributes are standardized by computing a Z score (X-mean)/standard deviation. The computation relies on probabilities for discrete values, but the effect is the same: it makes it possible to compare the coefficients across various attributes that have different measures. Recall the problem from the traditional regression—with income in the tens of thousands, years in thousands, and gender measured by ones—the coefficients are not directly comparable. Normalizing the data makes the coefficients comparable, and Microsoft generates a score value to show the relative strength of each attribute.

As shown in Figure 7.24, the results also include a Mining Accuracy Chart. Because the dependent variable is discrete, the chart and its interpretation are quite different from the chart created for simple linear regression. First, move the gray vertical bar to the far right so that the legend in the bottom right corner displays the statistics for the entire input population. This table is useful for comparing different models—although only the one logistic model is being used at the mo-

Model	Constant	Population	Income	Year	Gender
Race	-245.6	0.00027	0.0058	0.124	-0.199
Road	-225.8	0.00013	-0.0062	0.114	-0.203
Track	1967.4	0.00007	-0.0015	-0.986	-0.273
Tour	-111.1	0.00009	-0.0099	0.056	-0.251
Mountain	5.2	0.0001	-0.0055	-0.001	-0.174
Mountain full	-212.8	0.00014	-0.0072	0.108	-0.205
	+	*	*	*	*
Sample	1	80	65	2012	1
	Sum: XB	exp(XB)			
Race	4.0876	59.59668788	0.417531477		
Road	2.9724	19.53875638	0.136887571		
Track	-16.7969	5.07223E-08	3.55358E-10		
Tour	0.6847	1.983176793	0.01389404		
Mountain	2.6645	14.36076737	0.100610833		
Mountain full	3.8342	46.25640774	0.324070128		
		141.7357962	0.007005951		
		sum			

$$P_{ij} = \frac{e^{XiBj}}{1 + \sum_2^J e^{XiBj}}$$

$$P_{i1} = \frac{1}{1 + \sum_2^J e^{XiBj}}$$

Figure 7.25

Predicting probabilities. For traditional logistic regression, multiply the coefficient times the sample data and sum the results. Compute this XB value for each model type. Calculate exp(XB) for each model and add up the total. The probability of the base (Hybrid in this case) is 1/(1+exp-sum). The other probabilities are the exp(XB) times the base value. In this example, the prediction is for the purchase of a full suspension mountain bike at 100 percent probability.

ment. Notice that the percentage of cases classified correctly is 30.1 percent which is close to the 30.2 percent value from the traditional logistic tool. The Predict probability value of 24 percent represents the probability that an actual person selected from the group would respond as predicted by the model. Both percentages are lower than you might prefer, so it might be worthwhile to search for a better model. Still, the model has some value and it has provided some insight into the importance of the various attributes.

Attribute Evaluation

The examples make it clear that logistic regression does a good job of identifying the strength of the X attributes. The coefficients on the traditional logistic equation identify the strength of each attribute. However, they do not provide measures of slope or elasticity because of the nonlinear form of the model. More importantly, the logistic regression approach shows the breakdown of how the individual attributes will affect each of the outcome values, making it easy to compare and identify how input attributes can have different effects on each outcome.

Microsoft's logistic regression tools go even further in evaluating attributes for each outcome. The Neural Network Model Viewer is designed to identify attributes and rank their influence on side-by-side comparisons of outcomes.

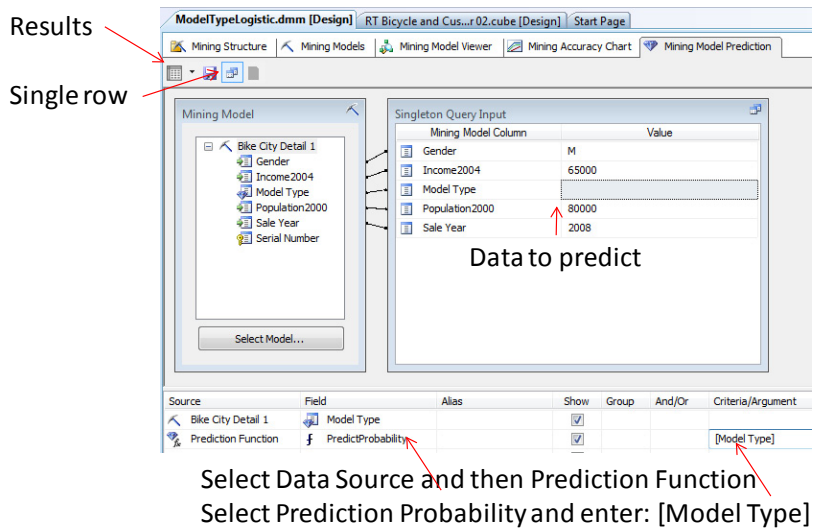


Figure 7.26

Microsoft single row prediction. Click the Single Row icon and enter the values to forecast. In the bottom grid, choose the model and ModelType attribute. Select the Results icon to see the forecast value, which is Race in this example.

Prediction

The traditional logistic regression can be used to estimate the probability that a person with a specified set of independent attributes will make a certain decision. In fact, the regression coefficients can be used to compute the probability of each of the outcomes occurring. Figure 7.25 shows the manual way to compute the probabilities based on the probability equations. For each model type, multiply the model's regression coefficients by the sample data and sum the results for that model to get XB (the sum of the x-value times the matching b coefficient). Compute exp(XB) for each model type and add the values. The probability of the base event (Hybrid) is 1/(1+sum-exp). The probabilities of the other events are found by multiplying the respective exp(XB) times the base probability. In this example with relatively high income and high population, the model predicts a 43-percent chance of purchasing a road bike with a 31-percent chance of a full suspension mountain bike.

Microsoft BI has an automated option to predict the outcome based on values of the input attributes. If you want to examine many data points, you can place them in a new table and have the system predict all of the rows in one pass. For a handful of values, it is easier to use the single-row option and compute them interactively one at a time. Figure 7.26 shows the basic process. Once you pick the Singleton Query option, you simply plug in the values of the X-attributes in the table. The second step is a little trickier. Select the data source in the bottom grid (Model Type Logistic), and set the predicted field Model Type as the Field. To see the probability value, you need to select Prediction Function on the second row. Enter [Model Type] including the brackets in the Criteria column. The results button computes and displays the most likely outcome (highest probability) for the case. In the example, the result is similar to the traditional logistic regression, which had Mountain full as a secondary choice with a slightly lower probability.

Predicted	MTN full (actual)	Track (actual)	Tour (actual)	Mountain (actual)	Hybrid (actual)	Race (actual)	Road (actual)
MTN full	**3326**	0	483	1530	138	2696	2460
Track	0	**0**	0	0	0	0	0
Tour	0	0	**0**	0	0	0	0
Mountain	444	21	186	**671**	158	567	426
Hybrid	0	0	0	0	**0**	0	0
Race	111	1	14	83	11	**78**	57
Road	0	0	0	0	0	0	**0**

Figure 7.27

Mining Accuracy Classification Matrix. Values on the diagonal were the number of cases predicted correctly. The other values are the number of items incorrect for each prediction. For example, the model was wrong 444 times while predicting mountain bike purchases that were actually full suspension purchases.

Microsoft also provides a method to examine the potential accuracy of the model. As you plug in various values for the X-attributes, you will begin to see a pattern—the model almost always predicts Mountain full or mountain bikes. At that point, you should become a little suspicious of the model and investigate its accuracy.

The Microsoft Accuracy Chart has an option to display the **Classification Matrix**, which uses the holdout data to show exactly where the model predicts correctly and incorrectly. The results for the ModelType are shown in Figure 7.27. When building a model to predict outcomes, it is always useful to know where the model is likely to be right and what happens when it is wrong. The numbers on the diagonal show the number of items that were predicted correctly. For example, 3326 times the model correctly predicted the purchase of full suspension mountain bikes, but 1530 times it predicted a full suspension purchase that was actually a mountain bike purchase. Check the rows for Tour and Road bikes and you will see the real weakness of the model—it predicted zero purchases of those model types. It appears that the model needs another factor to identify when people will purchase tour and road bikes instead of mountain bikes.

Naïve Bayes

How do you begin an analysis when you know little about the data? The **naïve Bayes** approach is a good tool for examining data when you have little background about the relationships. The method begins by assuming most of the data are unrelated (hence the name "naïve"), and it has proven to be relatively robust in finding initial relationships. It is not the most accurate or most powerful tool—it can miss details and it requires relatively broad categories of data. But, it is easy to implement, fast to run, and the results are relatively easy to understand. One important catch is that the Bayes method requires that the X-attributes and the Y-variable be discrete. Microsoft BI can convert continuous data to discrete groups, but the Y-attribute works best if it is legitimately a discrete set of values. Otherwise the results are highly dependent on how the discretization is handled.

Gender/Purchase	Yes	No	Total
Female	89	65	154
Male	172	139	311
total	261	204	465

Figure 7.28

Simple joint distribution. Observations on the number of people entering a store and whether or not a purchase was made. Goal is to find probability a purchase will be made if a new person enters the store.

In its simplest form, Bayes' Theorem is relatively easy to understand and use. Yet, Bayes' Theorem leads to an entirely different way to understand probability—and it provides a useful approach to machine learning. The theorem is a simple mathematical relationship of conditional probabilities. A **conditional probability** defines the probability of an event (B) occurring, given that another event (A) has already happened. It is written P(B | A) and pronounced "the probability of B given A."

Goals

The primary goal of the naïve Bayes data mining tool is to identify which attributes have a strong effect on the outcome variable. The approach uses an advanced version of Bayes' Theorem, where the system begins with a relatively neutral probability and uses data observations to estimate an updated posterior probability distribution function.

Simple Version

Before trying to understand that last sentence, it is best to begin with a simple description of Bayes' Theorem using a single binary attribute. Figure 7.28 shows a **contingency table** or collection data on a group of customers based on two attributes. The fictional customers were observed to be either male or female and the outcome consists of whether or not the person purchased an item during that visit. The outcome variable (purchase) is binary and the x-attribute (gender) is also binary. The table represents a complete tabulation of the joint distribution because it covers all 2 x 2 (= 4) possible cases. The table also shows the margin totals because they are important to Bayes' Theorem. Each cell represents the combination of both attributes. For example, 89 people were observed to be Female **and** make a purchase. In terms of frequency, the P(Gender=female and Purchase=yes) = 89/465. To simplify notation, this probability will be written as P(F ∩ Y).

The margin totals have their own interpretation—they are used to compute the conditional probabilities. To understand the term, answer the question: If you observe a representative female shopper, what is the probability that person makes a purchase? In notation the probability is written: P(Y | F). The answer is readily available from the table. Once you are given the gender, simply cover up or ignore the row of Male data. It is clear that 89/154 of these women made a purchase, which is the conditional probability. Similarly, it is easy to compute the P(M | N)—the probability the shopper was a man given that no purchase was made. Look at the information provided (No purchase) and cover up the rest, so the answer is 139/204.

The conditional probabilities work in any direction—the mathematics does not know which piece of information you are given first. So if P(Y | F) is 89/154, what is the probability that it was a female shopper if you are told that a purchase was made P(F | Y)? Because the table shows all of the possible combinations, again you look only at the data you are given (Purchase=yes) and find the answer directly as 89/261. The genius of Bayes was to realize that sometimes data is received in sequence and you do not have all of the parts—it is still possible to compute the conditional probability. First, the simple statement of Bayes' Theorem:

$$P(Y \mid F) = \frac{P(F \mid Y)P(Y)}{P(F)}$$

You want to predict the probability that female shopper will make a purchase, but all you have from past observations is that if someone did make a purchase, you know the number of times (frequency) that the shopper was female which is P(F|Y). You also know the conditional probability P(F | N)—the probability the customer was female given no purchase was made. Additionally you know the probability that any shopper will make a purchase. From these values you can compute P(F) and P(Y | F). Note that the denominator is always the sum of all possible numerators in the equation. In this case:

P(F) = P(F|Y)P(Y) + P(F|N)P(N)

The formula is easy to apply from the data in the table:

$$P(Y|F) = \frac{\left(\frac{89}{261}\right)\left(\frac{261}{465}\right)}{\left(\frac{89}{261}\right)\left(\frac{261}{465}\right)+\left(\frac{65}{204}\right)\left(\frac{204}{465}\right)} = \frac{\left(\frac{89}{465}\right)}{\left(\frac{89}{465}\right)\left(\frac{65}{465}\right)} = \frac{89}{154}$$

It is clear that this approach gives the same answer as that found directly from the table. Bayes' Theorem is relatively easy to prove using the definition of conditional probability:

P(A|B) = P(A Ç B) / P(B)

Simply write the corresponding definition of P(B|A) and combine the two equations.

Bayes' Theorem for Updating Probabilities

The bigger question is why does Bayes' Theorem matter? In the sample problem, it was easier to read the answer directly from the table instead of trying to remember the formula and plug in the values to compute the conditional probability. The answer to this question explains the value of the Bayesian data mining method. The answer consists of two factors: (1) It is rarely possible to obtain all of the data needed to generate the complete tabulation, and (2) Bayes' theorem has a critical interpretation for applying new information and learning.

First consider the issues of size. The sample problem had the binary outcome and a single binary attribute, leading to a 2*2 table of values. What happens when another binary attribute is added? Essentially, the table would have to expand into a third dimension, multiplying the number of cells by 2 again, for a total of 2^3 = 8 cells. In general, with n binary attributes, the complete joint distribution would contain 2^{n+1} cells. With even 9 binary attributes, the problem contains 1024 cells. If the attributes are not binary and contain additional attributes, the number of

cells can be substantially larger. For example, if each attribute has 4 options, the number of cells in a joint probability distribution with 9 attributes is $(2^2)^{10}$, which is slightly over one million cells. You should begin to see why dimension reduction becomes important.

The number of cells is just the beginning. Estimation of the distribution requires observations within each of those cells, and a sufficient number of observations to hold down the bias. Even the small binary problem with a tiny 10 observations per cell would require at least 10,000 evenly distributed data points. A more realistic problem quickly jumps to 10-100 million. Because these levels of data are rare, Bayes' Theorem becomes more important and the data mining tools need special assumptions to compute probabilities.

The real importance of Bayes' Theorem is found by slightly restating it and focusing on the interpretation of the elements:

$$P(Y|F) = P(Y)\frac{P(F|Y)}{P(F)}$$

In this version, P(Y) is the **prior distribution** (or *a priori* in Latin). P(Y|F) is the **posterior distribution** (*a posteriori*) . The prior distribution is simply the initial guess for the solution—it is typically assumed to be a neutral distribution. It could be a uniform distribution or perhaps an early measure of data. The known conditional distribution P(F|Y) is the likelihood measure obtained from data observations. It is normalized by the P(F) value which is also observed in the data. Consequently, Bayes' Theorem provides a way to take a prior belief about the probability and modify it with observed information to obtain a new, better belief about the probability. In Bayesian terms, probability is subjective and decision makers use new information to revise their subjective probabilities using this formula. As documented by Zellner (1971), this approach can be used to define statistical theory. However, it also illustrates the foundations of data mining and machine learning: Begin with a general estimate of probabilities, use observation on conditional data to compute a refined conditional probability that can be used for prediction.

Estimation

The naïve Bayes approach is a relatively simple method to examine the effect of discrete X-attributes on a discrete predictable variable. Its primary objective is to estimate the probability of each Y-value outcome given each of the independent X-attribute values.

Figure 7.29 illustrates the goal with the results of a Bayesian model. Each value of the Gender attribute is matched to each value of the ModelType outcome attribute. Each number is an estimated probability that a person with the specific attribute will purchase the listed model type. For instance, there is a 61.5 percent probability a man will purchase a Track bicycle. The process and the results chart become more complex when the problem has multiple attributes—particularly when the attributes have several values. Also, note that the results are rough, so remember that the goal is to provide an initial idea of potential attributes.

The probabilities in Figure 7.29 are the conditional probabilities from Bayes' Theorem (e.g., P(Gender=Male | ModelType=Track). Keep in mind that these probabilities are only a small portion of the overall results. The problem ultimately has more attributes (such as city population and income). Adding more attributes

Model/Gender	Male	Female	Missing
All	0.584	0.352	0.064
Track	0.615	0.346	0.038
Race	0.591	0.331	0.078
Tour	0.570	0.360	0.069
Road	0.579	0.348	0.073
MTN full	0.577	0.355	0.068
Hybrid	0.611	0.387	0.002
Mountain	0.591	0.371	0.038

Figure 7.29

Bayes objective. Compute the probability each attribute value (Gender) leads to a specific outcome (Model Type). In this partial table, 58.4 percent of the customers are Male. There is a 61.5 percent chance a man will buy a Track bike.

dramatically increases the size and complexity of the problem. One of the first steps to reduce the problem is to assume that the various x-attributes are independent. For instance, gender might affect the model choice and income might affect the model choice, but gender is not related to income and there is no interaction effect on the choice of model. This simplifying assumption leads to the "naïve" name. But, it means that the joint probabilities can be estimated with simple multiplications. From probability theory, if two events A and B are independent:

$$P(A \cap B) = P(A)P(B).$$

Consequently, Bayes Theorem for multiple X-attributes can be written as

$$P\left(Y|X1\ldots Xn\right) = \frac{P(Y)\prod_i^n p(Xi \mid Y)}{P(X1\ldots Xn)}$$

Much like the values in the simple tabular example, the conditional probabilities in the numerator can be estimated from frequency counts within each cell. However, it is difficult to obtain enough data to fully fit even the naïve independence model. Cells with few observations can bias the results—particularly with a large number of cells. Consequently, most tools estimate a probability density function instead. Assume that each probability arises from some underlying, relatively neutral distribution. Use the mean and variance to estimate the parameters of the distribution. Covariances are not needed because of the independence assumption. Often the data is smoothed to reduce problems with limited data. Different tools use different smoothing and estimation techniques, so final results can vary by tool.

Despite the naïve independence assumption, Bayesian classification has been shown to perform reasonably well in many cases. The actual probability numbers can be off, but the classification or relatively importance of the x-attributes is generally good—particularly with datasets that contain noise or complex relationships. In the end, the naïve Bayes tools produce a set of conditional probability estimates shown in Figure 7.29—for each attribute.

```
SELECT  dbo.Bicycle.ModelType, dbo.Bicycle.SalePrice,
        dbo.Bicycle.SerialNumber, dbo.Bicycle.FrameSize,
        YEAR(dbo.Bicycle.OrderDate) AS SaleYear,
        dbo.Bicycle.LetterStyleID, dbo.Bicycle.StoreID,
        dbo.Bicycle.EmployeeID, dbo.City.Population2000,
        dbo.City.Income2004, dbo.Customer.Gender
FROM      dbo.Bicycle
INNER JOIN dbo.Customer
        ON dbo.Bicycle.CustomerID = dbo.Customer.CustomerID
INNER JOIN dbo.City
        ON dbo.Customer.CityID = dbo.City.CityID
```

Figure 7.30

Data query. Combine Bicycle, Customer, and City tables. Compute SaleYear, and retrieve at least ModelType, Gender, Population, and Income attributes. SerialNumber is needed as the key column.

Data

Data for the naïve Bayes classification must be in discrete categories. In particular, the outcome values should be a discrete list. It is possible to use tools to create categories for continuous data for X-attributes. It does not make much sense to use a continuous variable as the predictable (Y) attribute—because the results would be highly dependent on the categories. It could be possible to use a continuous Y-attribute if a clustering algorithm was able to find strong groupings.

To illustrate the technique, assume Rolling Thunder Bicycle Company managers want to determine which types of people buy each model type. The company does not have much personal information about customers—it knows their gender, the year, and the model type purchased. All of these are discrete values. With some research through the Census Bureau, the company can find the average income of people living in the same cities as each customer, along with the size of the city. These last two attributes are continuous, but Microsoft BI can automatically define discrete groupings of these values. If predefined categories exist (perhaps for wealthy, typical, and low-income), the categories could be defined manually. If necessary, create a new data source view that contains the Bicycle, Customer, and City tables. Create a named query that links the tables and selects the desired columns. As shown in Figure 7.30, it should include a SaleYear column as Year(OrderDate).

Tools

For tiny problems, you might be able to use subtotals from a cube browser to compute all of the joint probabilities needed to fill a small table. From there, it would be possible to compute the Bayesian conditional probabilities directly. However, specialized naïve Bayes tools are almost always used to estimate the probabilities—both because they are efficient at handling large amounts of data and because they automatically solve complications such as smoothing.

Microsoft's naïve Bayes model is relatively easy to create and run. Add a new Mining Structure and choose Microsoft Naïve Bayes. Pick the appropriate data source view and select the named query you created as the Case table. Choose the ModelType as the predictable column, verify that SerialNumber is set as the key

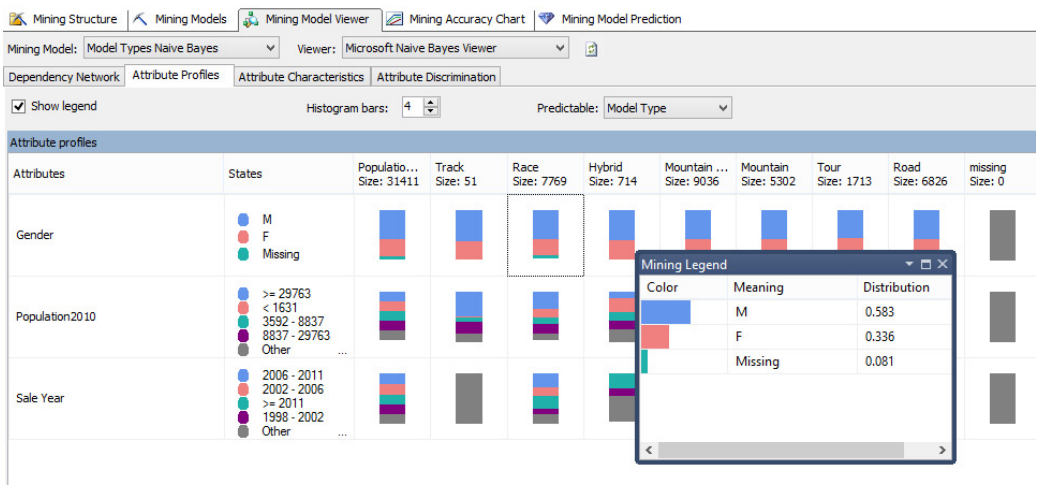


Figure 7.31

Bayesian classification results. Microsoft BI uses charts to highlight the differences and similarities across attributes. Opening the data legend provides the probability estimates.

column, and add the Gender, Income2009, Population2010,and SaleYear columns as input attributes. You will be given the option to set the data type for each column. Income and Population should be Discretized because they are continuous variables. SaleYear should be a Discrete variable. For some problems, year can become an issue—if a company has been collecting data for dozens of years, it will probably be necessary to group them into categories to reduce the complexity of the problem. Once the model is built, right-click its name in the Solution Explorer and choose the Option to Process and then Run the model.

Results

Figure 7.31 shows the results of browsing the Attribute Profiles from the Bayesian classification. It would be difficult to show all of the probability numbers for every attribute value on one page. Consequently, Microsoft BI uses small charts to show the probabilities. The charts make it easier to look for similarities and differences across attributes. You can open the data legend for a specific attribute to see the probability details. Glancing at the charts, it appears that Gender and Income might have strong effects on the choice of models—the probabilities are high and they appear to vary across model types. Population might be less important, but it appears to be different for Track and Race bikes.

Attribute Evaluation

The Bayesian tool provides more detailed evaluations of attributes within the results under the Attribute Characteristics tab. In Figure 7.32, notice the importance of the Gender attribute values, followed by the effect of city size (population). In particular, it appears that people from larger cities have a higher probability of purchasing the full suspension mountain bike. Use the drop-down list to select different model types (outcome attributes). Also, rolling the mouse cursor over one of the bars pops up the actual probability value.

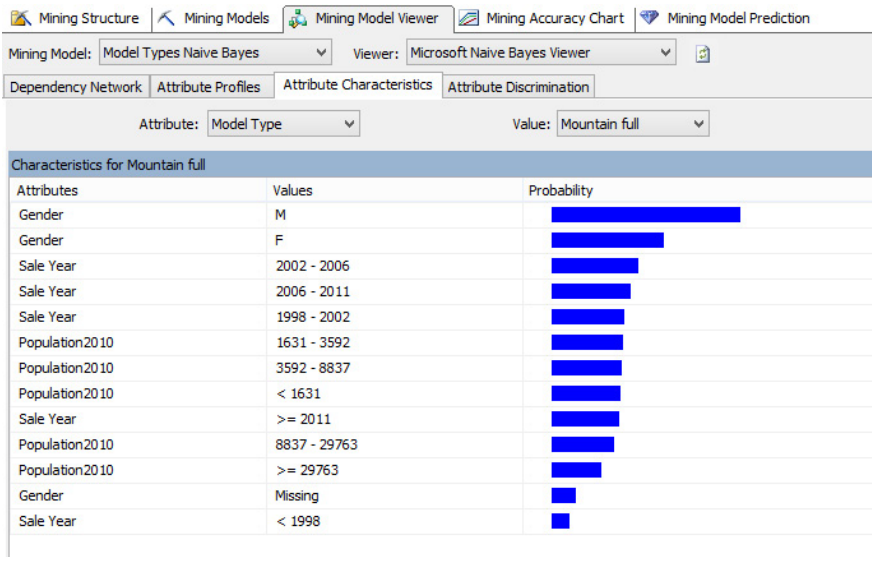
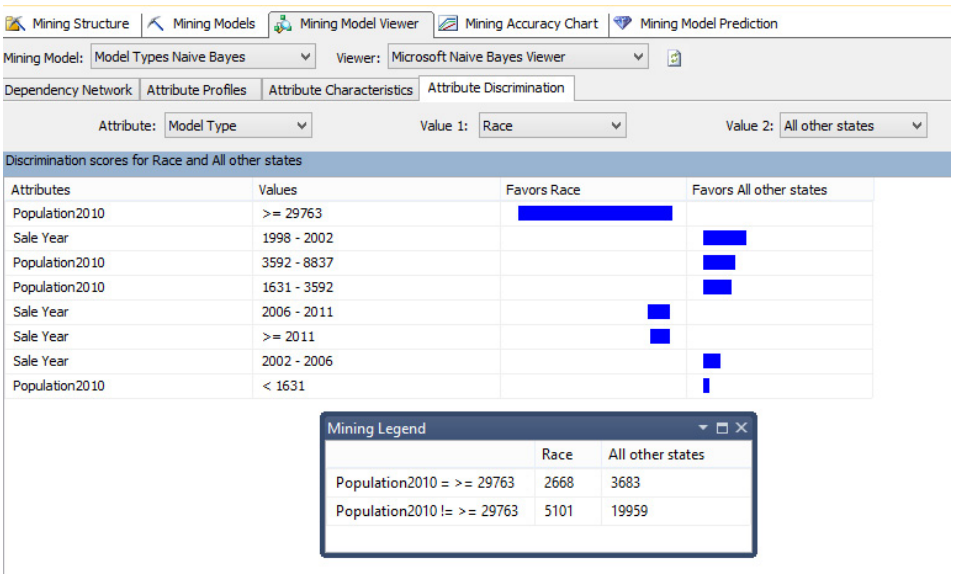


Figure 7.32

Attribute evaluation. Details for each model type value provide more information about the conditional probabilities. The attributes at the top of the list have a stronger effect on the outcome.

Figure 7.33

Attribute discrimination. The ability of attributes to discriminate among the outcomes is important in analyzing the data. This chart enables you to focus on one value (e.g., Race) and compare it to all other values or to a second specific value.



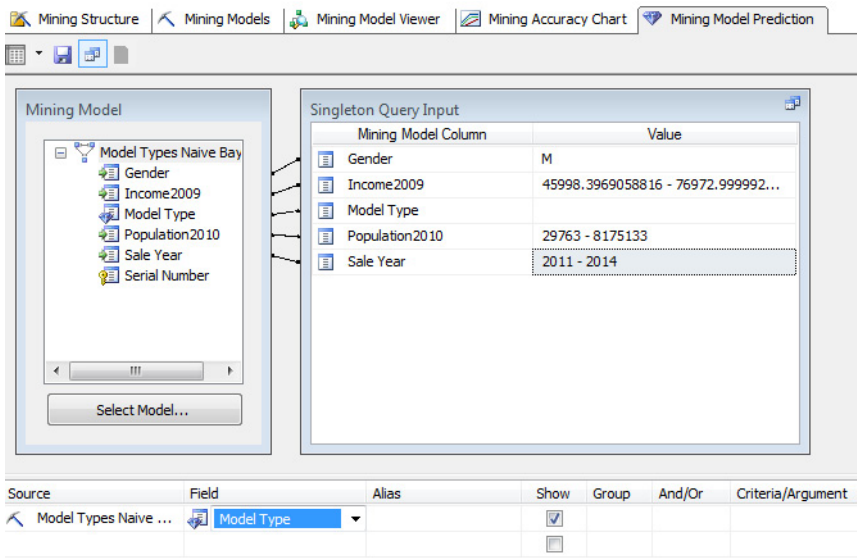


Figure 7.34

Microsoft single row prediction. Click the Single Row icon and enter the values to forecast. In the bottom grid, choose the model and ModelType attribute. Select the Results icon to see the forecast value, which is Race in this example.

Microsoft BI provides another chart to help evaluate attributes. Click the Attribute Discrimination tab to compare each model type to other model types—both as a group or individually. Figure 7.33 shows a discrimination chart for the Race model type versus all others. The drop-down boxes can be used to compare to another specific model, or to change the main value. Attribute discrimination is important when analyzing the data. It represents the ability of the model to differentiate among the selections. For example, an analysis of attribute importance might reveal that gender and population are important for every model type. An evaluation of discrimination ability might find that sale year or a specific population group is critical to explaining which group buys mountain bikes instead of road bikes. If a model has low discrimination ability, the analysis will lack detail. It might be able to predict increases or decreases in total sales, but it will not have the ability to specify exactly which model types will be sold. In this example, large cities are clearly key to predicting sales of race bikes. Examine the other model types and see if large cities play the same role. You will find that it is strongest for the race bikes, which is a useful piece of information when managers begin devising a marketing strategy.

Prediction

The naïve Bayes model can be used for prediction. Microsoft BI provides the same prediction tool it uses for other models. You can create a separate table of X-attribute values that you want to predict, or you can enter data for a single set of values and interactively get a prediction. However, you should keep in mind that the Bayesian probability estimates are typically relatively rough. The category

Predicted	MTN full (actual)	Track (actual)	Tour (actual)	Mountain (actual)	Hybrid (actual)	Race (actual)	Road (actual)
MTN full	**2959**	0	386	1384	102	1667	1982
Track	0	**0**	0	0	0	0	0
Tour	0	0	**0**	0	0	0	0
Mountain	151	5	145	**438**	118	237	297
Hybrid	0	0	0	0	**0**	0	0
Race	837	16	155	394	62	**1430**	614
Road	0	0	0	0	0	0	0

Figure 7.35

Mining Accuracy Classification Matrix. Values on the diagonal were the number of cases predicted correctly. The other values are the number of items incorrect for each prediction. For example, the model was wrong 233 times while predicting mountain bike purchases that were actually full suspension purchases.

classifications tend to be good, but the actual probability numbers can be wild-particularly when the independence assumption is violated. See Rish (2001) for a study of the performance of the naïve Bayes classifier.

Figure 7.34 shows the process for generating a simple forecast. Note that because all of the attributes are discretized, you must choose from among the values, you cannot enter random numbers. Keep this constraint in mind if you choose to build a data table with values to be predicted. The predicted outcome of these values is the Race model type. Recall that large-city population is a strong discriminator for this model, so virtually any input values using the large-city choice are going to result in the Race outcome. Yet, the data show that the company sells road and mountain bikes to large cities as well, so the forecast is a little biased.

As shown in Figure 7.35, the classification matrix is a useful way to examine the model forecasts. The row represents the forecast item and the column is the actual number. For example, the Bayesian model correctly predicted the sale of 2959 full suspension mountain bikes, but 386 times when it predicted mountain full, the sale was actually tour bikes and 1384 times it was wrong when it predicted full suspension sales that were actually purchases of hard tail mountain bikes. More importantly, notice that the model predicts zero sales of Track, Tour, Hybrid, and Road bikes. The first three might be valid because those models were sometimes discontinued; however, zero sales for road bikes is a big problem. Clearly, the model lacks the ability to differentiate road bikes from other model types—particularly full-suspension mountain bikes.

Decision Trees

Is there a way to organize the attributes to see how they explain the decision? A **decision tree** is a tree or graph that models how attribute values influence a categorical outcome variable. Each node classifies a break point in an attribute that has a different effect on the outcome variable. Trees begin with a single starting node and follow a path to new nodes. Splits are made based on the values of an attribute. If the attribute values have significantly different effects on the outcome variable, a separate node and path are created. Decision trees

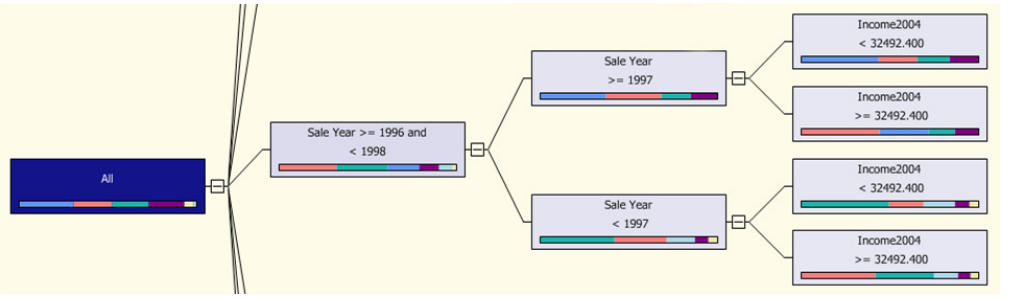


Figure 7.36

Partial decision tree. The nodes represent split points where an attribute has a significantly different effect on the outcome variable (Model Type). Following a path from the top (left) node represents a distinct classification of the impact of data on choosing a model type.

can be descriptive, but in data mining, the ultimate goal is to discover a tree that best predicts the outcome variable. Each node split leads to a different outcome.

Figure 7.36 shows a portion of a decision tree. Following a path from the starting (left) node yields a classifier for how the attribute values affect the model type outcome variable. For example, follow the top-most path in the figure to see that (1) SaleYear between before 1996, (2) SaleYear before 1995, and (3) Gender not missing (null) leads to a specific purchase decision. The combination of model types unique to that path is shown by the small, color-coded bar chart on the node. Based on results presented in a later section, that particular path has a 29 percent probability of purchasing a race bike (the largest, orange bar). Simplifying that particular path: Before 1995 (that is, 1994) customers with an identified gender (M/F) 29 percent chance of purchasing a race bike; followed by a 26 percent chance of purchasing a mountain bike. Because the value of SaleYear is set to a single value, this particular node is not very useful for predicting what will happen next year. However, other nodes cover different years, and if managers have additional or subjective information about those years, the conclusion can still be useful. Decision trees are probably the most popular data mining tool because the results are easy to understand and the model is relatively easy to create. Microsoft BI selects decision trees as the default method and many of the other routines use at least portions of the method.

Goals

The decision tree model is one of the important attribute classifier tools available for data mining. It requires minimal supervision by the analyst—simply choose the Y-attribute outcome and select a handful of X-attributes that might influence the outcome variable. The outcome variable should hold categorical values. The input attributes can be discrete or continuous. The primary goal of the method is to determine subgroups of the attributes that have different effects on the outcome variable. The model created can be used to predict the outcome. Simply use the classifications to identify the most appropriate group based on the input attributes and the probability of the outcome is provided.

Decision trees are built iteratively. From any node, the system searches for correlations between input attributes and the output variable. When an important dif-

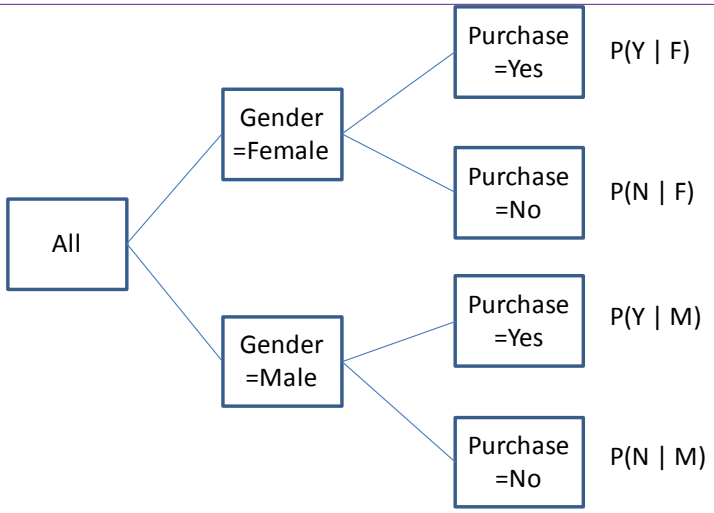


Figure 7.37

Decision tree as Bayesian network. Each lower node is condition on the node above (to the left of) it. So probabilities at any node are actually conditional probabilities and depend on all nodes above that point.

ference is identified, the node is split into new paths. The number of paths emanating from a node depends on how the data affects the output variable. Various algorithms have been defined to create trees with differences largely in how correlation is measured and how the gain from splitting a node is defined.

A decision tree is sometimes considered to be a Bayesian network, because the lower nodes on the tree are conditional on the prior nodes. Figure 7.37 illustrates the basic principle. Any probability at a node depends on the nodes that came before it. The assumption of independence of attributes essentially means that the nodes do not cross, leaving a relatively simple tree.

Microsoft exploits these relationships and the Microsoft decision tree algorithm uses a Bayesian approach by default. Microsoft BI offers two variants of the Bayesian approach—it can use a diffuse prior which assumes no bias and all options are equally likely, or the parameters can follow a Dirichlet distribution. Microsoft explains the Dirichlet approach and other statistical details of the algorithm in a white paper (Heckerman, Geiger, and Chickering 1995).

Decision trees are typically built up one node at a time. At each point, the system uses the underlying data to determine if splitting a node beyond that point will improve the network's ability to explain the results. The data consists primarily of the counts of the observations within each category, which is applied to estimating the underlying probability density functions. In a Bayesian context, the search systems typically use a **maximum likelihood estimator (MLE)** to see if a new variation will improve the likelihood measure. The MLE measure is typically computed as the log of the probability function.

A slightly different approach to choosing whether to split a tree is to use **Shannon's entropy** or information measure. Shannon was a scientist who developed several important theories regarding the transmission of data. These theories have been useful in evaluating the information content of data, and provide a method to

x	p(x)	-p log(p)	p2(x)	-p log(p)
1	0.25	0.150515	0.7	0.108431
2	0.25	0.150515	0.05	0.065051
3	0.25	0.150515	0.2	0.139794
4	0.25	0.150515	0.05	0.065051
		0.60206		0.378328

Figure 7.38

Example of Shannon entropy. The first distribution is diffuse, evenly distributed data. The entropy is high with little information. The distribution is boring. The second example is more interesting where some values have higher probabilities. The entropy is lower, the information is higher.

compare models. The basic information definition is straightforward. Given a set of events (x) defined by their probabilities (p), the information content is:

$$H(X) = -\sum_i p_i \log(p_i)$$

Many decision tree systems use this information measure to decide how to split nodes. Microsoft BI offers it as an option. To understand the measure, Figure 7.38 shows two versions of a distribution. The first is a rectangular diffuse distribution where each point has the same probability. Plotted, it is a flat line. Its entropy is relatively high and there is little information in the data. The second distribution is more interesting because some values have substantially higher probabilities. The entropy measure is much lower so the information content is higher. The measure is easy to compute and the change in its value from one model to the next provides a good measure of the information gained.

Data

Traditionally, the data for a decision tree consists of discrete values; however, continuous values can be used for the X-attributes. It is possible to use a continuous variable for the predictable variable, but then the model becomes a regression model tree. This is the method Microsoft uses to solve the basic linear regression problem. So, this section considers the dependent variable to hold categorical data. With Microsoft BI, attributes with too many discrete values are often automatically compressed to fewer categories using feature selection. A large number of attributes with multiple values can present problems in terms of the size of the estimation and in terms of over fitting the model. Microsoft BI automatically checks the data and reduces the number of dimensions if needed.

To compare this approach with the others, you can use the same data source. The objective is to build a decision tree model that can predict the sale of the various model types. The basic attributes to be tested consist of Gender, SaleYear, and income and population taken from the City table. If the data source does not yet exist, create a new one and add the Bicycle, City, and Customer tables. Create a new named query that joins those three tables and uses SQL to select the main attributes: SerialNumber, ModelType, Gender, Population2010, and Income2009. Create SaleYear as a new column using the expression Year(OrderDate).

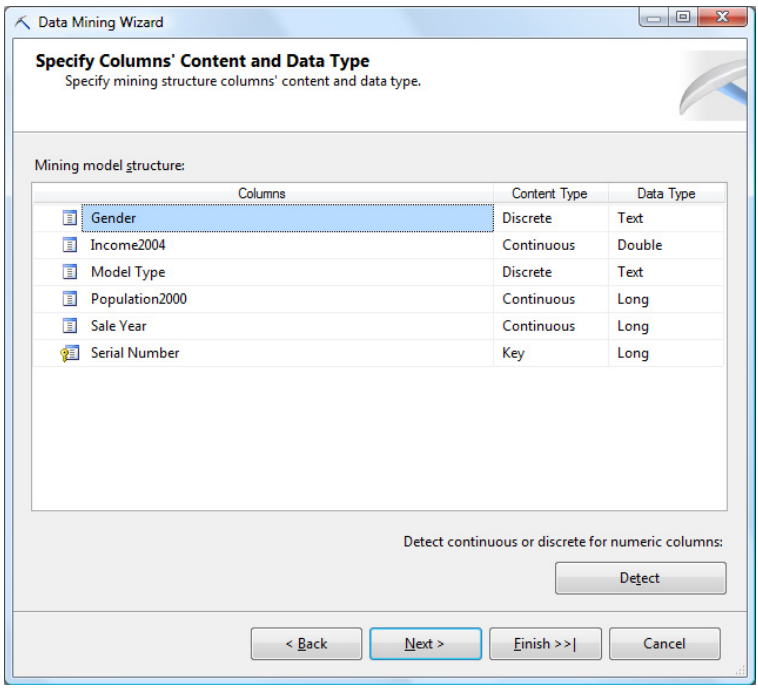


Figure 7.39

Microsoft decision tree setup. SaleYear is seen as continuous. Leaving it that way enables the tool to find ranges of values instead of focusing on single years.

Tools

The Microsoft decision tree tool is relatively easy to configure. Once the data has been established, add a new Mining Structure and choose the decision tree tool. Select the data source view that contains the named query holding the data. Pick that query as the Case table. Choose the columns and assign them to the appropriate categories. SerialNumber is the unique key column because the analysis is focused at the bicycle level. ModelType is the predictable variable. The other columns are simply input attributes. The wizard shows you the data types. Notice in Figure 7.39 that SaleYear is evaluated as continuous data, even though it holds integer values. With a smaller number of years, it might be useful to leave this attribute as a discrete type. However, to reduce the number of dimensions, it is more flexible to treat it as a continuous variable. This approach will also enable the tool to identify important ranges of years instead of examining one year at a time.

When the model has been defined, it is straightforward to select the model, and Process it to deploy it to the analysis server and Run it to obtain the base results. Once the model is processed, it can be Browsed to see the results.

Results

Figure 7.40 shows the results presented as the entire tree. One problem with decision trees using many attributes is that the trees can become large and complex. They are difficult to view on paper. The Tree Viewer contains options not shown in the figure to zoom in, or collapse the tree to a limited number of levels. The

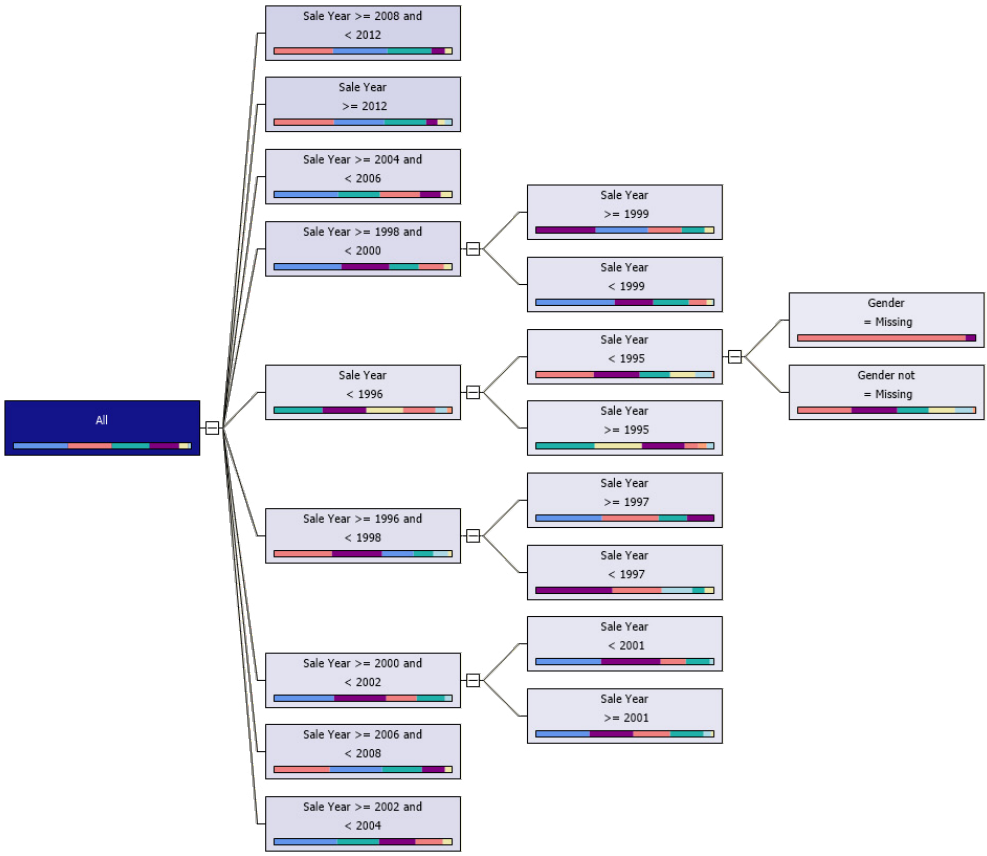


Figure 7.40

Decision tree results. The number of leaf (final) nodes shows the number of distinct classifications (14 in this case). Rolling the cursor over any node reveals the count of the number of each model type.

bar chart on each node indicates the distribution of model types within that node. Rolling the mouse cursor over a node will show the actual count by model type for that node in a popup box. If you checked the box to Enable Drill-Through as one of the final steps in the wizard, you can right-click any node and choose the Drill Through option to see a table of the underlying data that matches the conditions of the chosen node.

To get more details about any node, right-click a node and choose the option to Show Legend. As shown in Figure 7.41, the legend provides the number of cases in the node for each model type and the estimate of the percentage of items that would choose each model type. In this example, the customers were likely to purchase a race with a 29 percent probability. Examining the legends for various end nodes should help you see the conditions that lead to each of the outcomes. If managers have additional subjective information about the nodes, such as marketing campaigns or weather, it can help explain how customers behave.

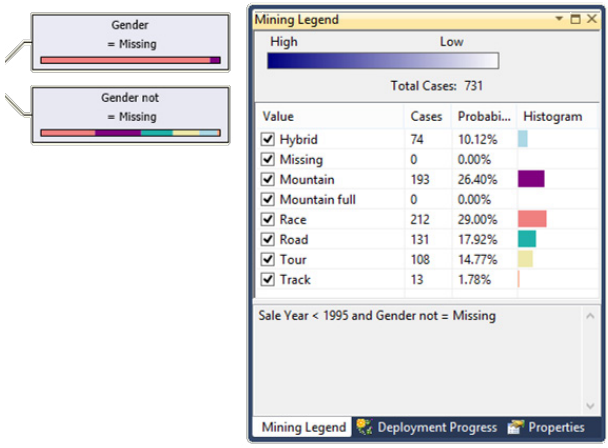


Figure 7.41

Mining legend. Right-click a node to see the legend for that node. Notice that it provides the classification probabilities. In this node, customers meeting the node conditions of year and income purchased mountain bikes with a 48 percent probability.

Attribute Evaluation

The main purpose of the decision tree is to find classifications that describe how combinations of the attributes affect the outcome variable. The decision tree itself provides a visual way to see patterns and how the attributes combine to affect the outcome. When examining the tree, you should look for similarities and differences in the node bar charts. It would be nice if the viewer made it easy to sort and reorganize the tree to group similar items together. As it is, you get to use those skills honed as a child searching two similar comics to find the seven differences between images.

You might want to focus on one specific model type at a time. For instance, find the nodes that predict a relatively high number of full suspension mountain bikes (medium-blue bar), then look for similarities in the attributes. Most of the differences appear to be driven by the year. Starting from the left-most node, the first division appears to be sales between 2004 and 2006. Full suspension bikes appear to be in the lead for those years. After 2006, the sales are split and depend mostly on year but somewhat on gender. In the later years, race bikes appear to have become more popular. When looking through the decision tree, remember that each node is dependent on its parent nodes—the conditions apply in a chain.

Prediction

In many ways, decision trees are automatically built to enable prediction. The easy way is to start at the top (left) of the tree and apply the if-then classification conditions to the desired data. Follow the path that matches the data to an end node. Check the probabilities within that node to get the most-likely outcome as well as the probabilities of the other outcomes. Figure 7.42 shows how to apply the sample data to obtain a prediction. The first level is set by the sale year. The applicable node (Sale Year >= 2006) has only one split into two children based on

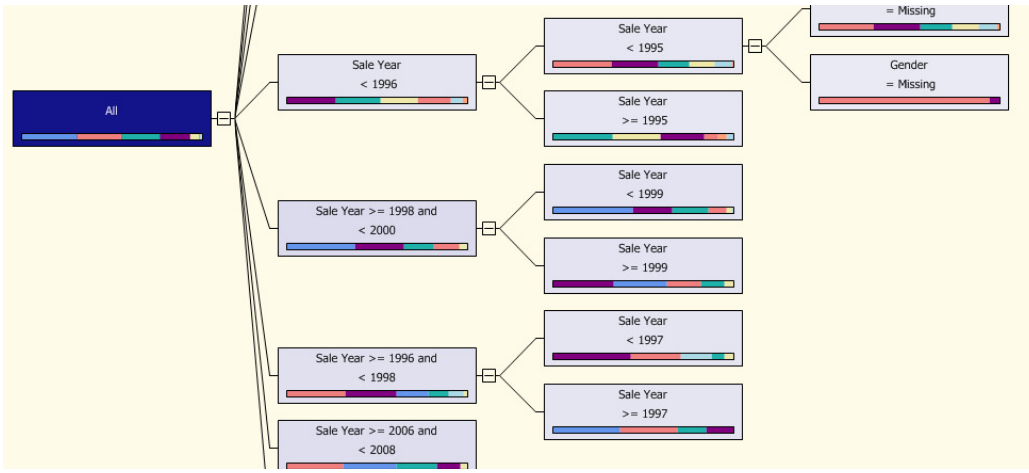


Figure 7.42

Prediction by following the tree nodes. X-attributes are year=1997, income=60,000, population=80,000, and gender=male. The path is short and determined only by year. The full suspension model at 36 percent is the most likely outcome.

income. The node (Income >= 32,492) matches the sample data and it is the end node. Notice that the gender and population attributes do not affect the classification or prediction. Examining the legend details for the ending node reveals that Race is the leading outcome with a 39 percent probability, followed by mountain full at 27 percent, and road at 19 percent probability. This manual path through the tree is easy to use and easy to explain to others. It is available with all tools that generate decision trees.

Microsoft BI also provides a prediction method where data rows to be predicted can be entered into a special table. The method predicts all rows in one batch and is useful when several outcomes need to be predicted and compared. The table results can be analyzed separately or exported to other tools. Figure 7.43 shows the basic setup for entering the X-attribute values interactively. Switching to the Result View reveals the prediction of the Mountain full model type. However, it does not provide the probability details or information about other model types. Consequently, for decision trees, this tool is more useful for batches of data. For single rows, it is usually better to simply trace through the tree.

Microsoft BI provides several common tools to evaluate the accuracy of the model including the Lift Chart which compares the predicted values to a perfect model. It can also be used to compare variations of models to each other. As shown in Figure 7.44, the Classification Matrix is a tool that compares individual forecasts to the actual results. Simply glancing at the matrix reveals that in at least one area it does a better job than some of the other tools. It predicts some sales for Road bikes, while some of the other tools always predict zero.

Decision trees often have a problem with over fitting. The models tend to adapt too much to the sample data used. If the data are somewhat limited or do not accurately match the population, the over fitting can cause serious prediction errors. Some tools use techniques to reduce the problem. For example, the tools can automatically divide the data into smaller groups and fit decision trees to each group.

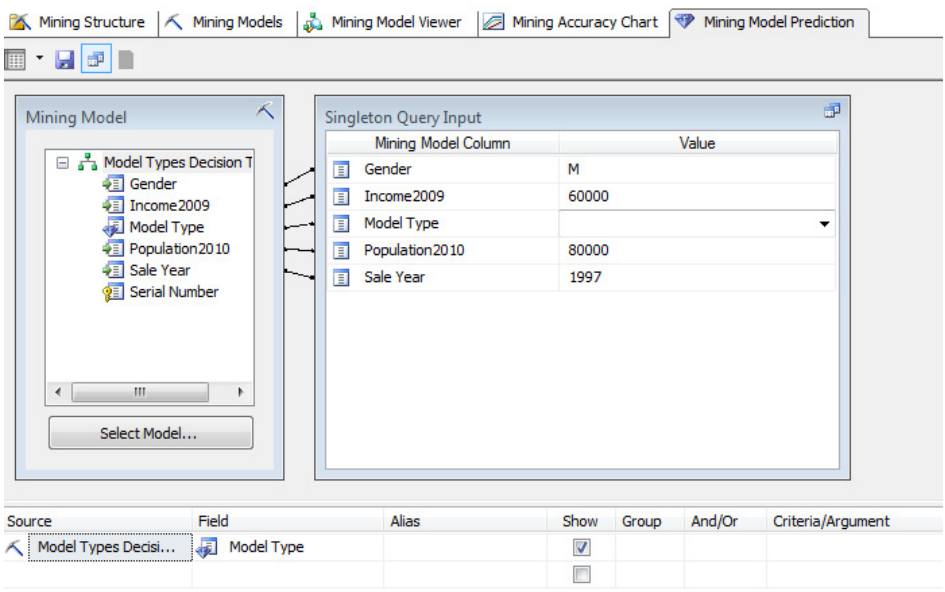


Figure 7.43

Prediction of decision tree. Be sure to set the model in the drop-down list at the bottom and choose the Model Type outcome variable. Switch to Result View to see that the system predicts the most likely outcome as Mountain full.

Figure 7.44

Mining Accuracy Classification Matrix. Values on the diagonal were the number of cases predicted correctly. The other values are the number of items incorrect for each prediction. For example, the model was wrong 339 times while predicting race bike purchases that were actually full suspension purchases.

Predicted	MTN full (actual)	Track (actual)	Tour (actual)	Mountain (actual)	Hybrid (actual)	Race (actual)	Road (actual)
MTN full	**1901**	0	212	1029	53	1074	1147
Track	0	**0**	0	0	0	0	0
Tour	0	0	**0**	0	0	0	0
Mountain	239	0	76	**530**	110	374	166
Hybrid	0	0	0	0	**0**	0	0
Race	1773	2	339	577	135	**1849**	1494
Road	0	19	93	83	25	37	**124**

Error	Pass	Fail	Log Like.	Lift	RMSE
1	2092	4189	-1.4654	0.1198	0.6676
2	2097	4185	-1.4662	0.1189	0.6677
3	2097	4186	-1.4664	0.1190	0.6674
4	2095	4187	-1.4658	0.1202	0.6674
5	2092	4191	-1.4693	0.1167	0.6676
Avg	2094.6	4187.6	-1.4666	0.1189	0.6675

Figure 7.45

Cross validation summary. Five different error measures were computed on five folds of data. The Pass/Fail numbers represent the number of cases predicted correctly and incorrectly (out of about 6282). Because the errors are relatively consistent, the model is reasonably robust and not over fit.

The final tree is an average of results from each group. If the amount of data is limited, some systems use **bootstrapping** to generate additional data points that match the characteristics of the sample with some degree of randomness. The Microsoft decision tree algorithm does not support bootstrapping.

Use the Cross Validation tool to check for over fitting. The tool divides the data into sets (typically choose 5 or 10 folds). It then fits a decision tree to all of the data less one of the sets. It repeats this process until each set has been held out. The cross validation tool computes error measures for each application. If the error measures vary radically across the sets, then the model is not very robust and is susceptible to over fitting.

Figure 7.45 summarizes the results of the cross validation tool applied to 5 folds or partitions of data. The pass/fail measures are counts of the number of predictions correct or incorrect. Each fold has about 6282 observations. The log likelihood value is derived from the probability density function. Likelihood is the probability of each probability arising and is computed as the product of all of the probabilities. Lift is the gain in predictive value of the model compared to random selection. Microsoft's technical notes state that it is measured from the log of the actual probability. RMSE for a categorical variable is slightly different from that used for continuous data. It is the square root of the mean of the squares of the complement of the probability scores (logs). The details of the definitions are not critical. The key is that each of the measures should be consistent across the various folds. If the results revealed significant differences, it would indicate that the model fitting is highly dependent on specific data points and would be less useful in predicting values on any other set of data. In this case, all of the measures are highly consistent, so over fitting does not appear to be a problem.

Neural Network

How can the modeling process be automated even more and handle nonlinear relationships? At several points in the discussion and development of computers, people have asked questions comparing computers to humans. Computers are amazingly faster and more precise than humans at computations and retrieving specific items from memory. Yet, people are incredibly fast at tasks involving pattern recognition and retrieving associated items from

memory. These discussions led to the question of how people think, and what it would take to make computers think the same way. After a few wrong turns and many years, one of the most important answers to these questions led to the development of neural networks. Check out Rumelhart and McClelland (1986) for details on the early development concepts.

The name **neural network** comes from the description of the way the human brain is built: as a large interconnection of neurons. Think of a neuron as a single cell that has inputs and an output that emits an electrical signal. When total input values reach some level, the neuron "fires" and emits its own output signal. The network consists of connections among many of the cells. The network holds patterns and images. Fortunately, you do not need to become a specialist on human brains to use the tools. In fact, the tools probably depart significantly from human anatomy, because the tools have been adapted to computer processing and to solve computational problems.

From a data mining perspective, the best way to think about neural networks is in terms of the output or goals. Essentially, a neural network defines a relationship between the input X-attributes and the output Y-attribute. Conceptually, this relationship is similar to a regression relationship where the system minimizes errors. However, neural networks have the ability to define highly non-linear relationships. The non-linearity makes it possible to estimate complex interactions among the attributes, but it also makes it difficult to understand and analyze these relationships. In many ways, the result can become a black box, where the system might produce accurate predictions, but the workings of the relationships are essentially hidden. This tradeoff becomes an important decision when you use the tool. Are you willing to give up the ability to understand the relationships to gain an improvement in prediction? If you cannot completely understand the relationships, will the predictions be valid as input data changes? The answers depend on the details of the problem being studied. You can certainly run a neural network on almost any problem, and sometimes the predictive ability is high and the model is easy to understand. But, as models become more complex, you should always take a step back and ask yourself if the model is going to work for the specific situation you are studying.

Goals

The ultimate goal is to define a relationship between the input attributes and the output variable. The primary purpose is to create a model with high predictive accuracy. The technique for achieving this goal appears a little unusual when you first encounter it, and you need to understand some of the terminology to work with the results. As shown in Figure 7.46, neurons or perceptrons are the foundation of the method. Think of a neuron as a smart light bulb. It receives input energy (v) from other cells. Each input is weighted by some value (w) and the result is totaled. If the input energy exceeds a certain threshold (T), this neuron lights up and emits a signal. Technically, the signal emitted is modified by a sigmoid (s-shaped) function to ensure that the values range from zero to one. Microsoft BI uses the most common sigmoid functions: hyperbolic tangent (tanh) for the hidden layer, and a logistic function for the output layer. The immediate goal of this tool is to find the values of the weights (w) that provide the best fitting model.

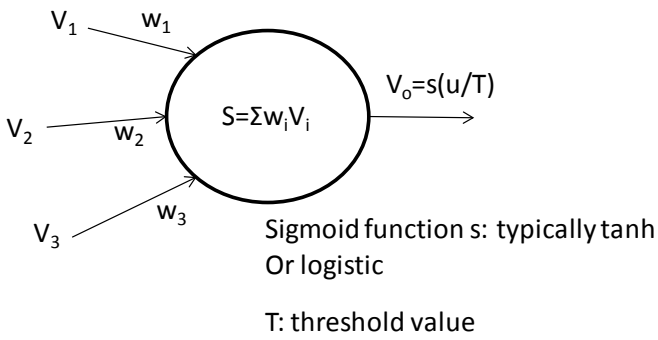


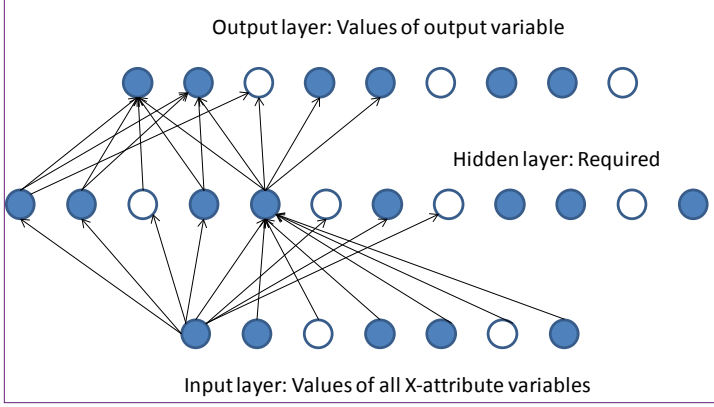
Figure 7.46

Neuron or perceptron. A neuron receives inputs from other neurons. These signals (v) are weighted (w) and summed to get the total input value. If the value is greater than a threshold number, the neuron emits a signal. This signal is modified by a sigmoid function to keep its value between zero and one.

Neurons exist in groups because the patterns are stored within the network. A key feature of neural networks is that they need three layers of neurons: Input, Output, and Hidden. As indicated in Figure 7.47, the neurons in the hidden layer are connected to every neuron in the input layer and every neuron in the output layer. In a data mining application, each input neuron represents one of the X-attributes. If the attribute is categorical, separate input neurons are created for each categorical value. The input values are normalized to keep them within reasonable ranges. For example, continuous values are converted to a version of a Z-score ([value-mean]/standard deviation). An output neuron is created for each

Figure 7.47

Neural network layers. The input layer consists of values from the X-attributes. The output layer contains the outcomes to be predicted. The hidden layer is critical and provides the main flexibility in the relationships. Only a few connections are shown here. Typically, every hidden neuron is connected to every input and every output neuron.



possible outcome of the predictable variable. If there are too many outcome variables, the system typically splits into a second network. With Microsoft BI, 500 is the definition of "too many." In the Rolling Thunder Bicycles example, each output neuron represents one model type. The hidden layer is a buffer that is crucial to provide the flexibility to model complex problems.

The neural network method uses the sample data to train the network—essentially finding the weights and threshold values that lead to the best predictions. Along the way, the system has to determine the number of neurons in the hidden layer. A larger number improves the predictive ability but increases the complexity of the model. If too many neurons are added, the model becomes over fitted and the model will work only with the sample data. Microsoft BI provides you some control over the number of neurons in the hidden layer through parameters. The HIDDEN_NODE_RATIO parameter defaults to 4.0 and provides an initial estimate of the number of neurons by multiplying by the square root of the number of input cells times the number of output cells.

Data

Neural networks can analyze almost any type of data. The method is general and has been used successfully for many complex problems, including predicting continuous and categorical variables. The technique is also commonly used for pattern-matching problems, such as text and speech recognition. The most important step is to identify the outcome variable. If the dependent (predictable) variable holds continuous data, Microsoft BI converts it to discrete bins.

The Rolling Thunder Bicycle company case is easy to configure as a neural network problem. Again, the goal is to predict selection of model type based on the limited customer attributes available. The easiest approach is to create a data source view that contains the three main tables: Bicycle, Customer, and City. Then build a named query that includes at least the SerialNumber, ModelType, Gender, SaleYear computed from OrderDate, Income, and Population columns. This query is the same as the one used in the other sections of this chapter.

Tools

Configuring Microsoft BI to estimate the neural network is straightforward. Add a new Mining Structure and choose the Neural Network method. Pick the data source view that contains the data. Set the named query as the main case table. For choosing columns, be sure that SerialNumber is set as the key because the data is organized by bicycle. Set ModelType as the predictable column, then select Gender, Income2004, Population2000, and SaleYear as input columns. Technically, it is possible to select a column as both predictable and input, but choosing that approach makes the results difficult to understand and use for prediction.

Results

Because of their underlying nature, results from a neural network can be difficult to comprehend. What does a weight on a hidden neuron really mean? Predicting outcomes is usually reasonable, and Microsoft BI provides a useful tool for predicting outcomes based on various input data. It is more difficult to understand relationships and the impact of input variables on the outcomes. Some tools convert the internal weights into a nonlinear equation from the input attributes to the outcome variable. Although the equations are often complex, they can aid in understanding the relationships. Microsoft BI does not attempt to provide equations.

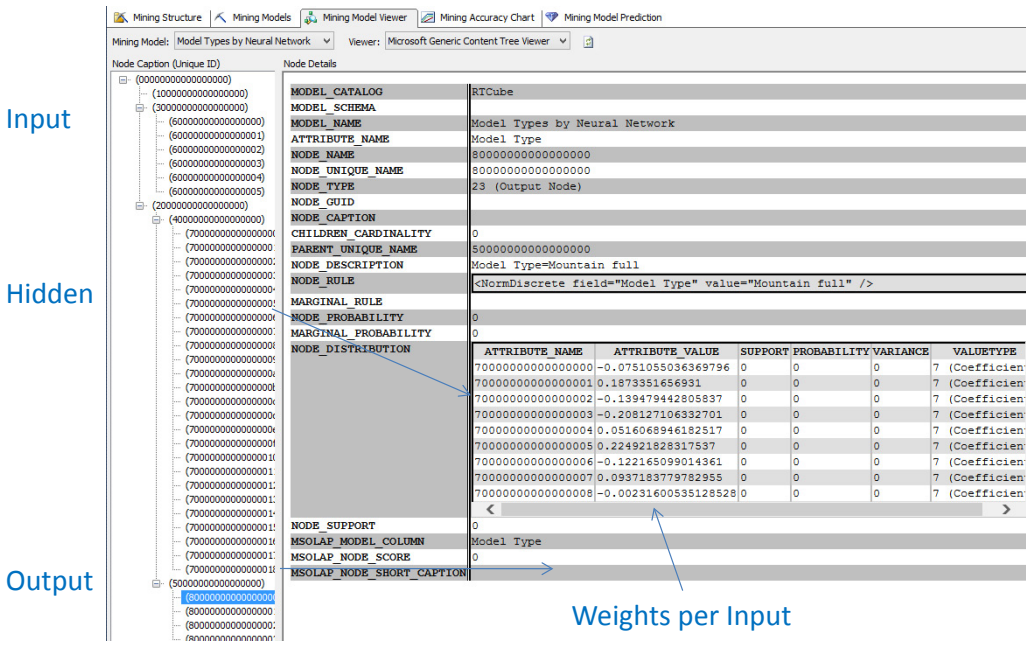


Figure 7.48

Neural Network detailed results. Microsoft's Generic Content Tree Viewer lists the estimated weights for every neuron connection. Choose the neuron in the list on the left and scroll through the weight table. This example shows the weights on the hidden layer neurons connected to the output neuron for the mountain full model.

Instead it provides the values of the weights estimated for each neuron. However, Microsoft does provide a tool to compare outcomes side-by-side.

Figure 7.48 shows some of the detail weights for the output neuron for the mountain full model type. This view is available in the Mining Model Viewer by selecting the Generic Content Tree Viewer and expanding the nodes in the tree list. Remember that these weights come from the 19 neurons in the hidden layer. By themselves, the numbers are difficult to understand. By hand, it would be possible, but difficult, to track the input weights through all hidden-layer neurons to the weights on the outcome variable. Microsoft BI provides a second viewer to help explore the importance of the various attributes.

Attribute Evaluation

Microsoft BI provides the Neural Network Viewer as the primary tool to explore the strength of various input attributes. The most important attribute values are listed at the top, so exploring the tables for various combinations can provide insight into the strength of each attribute. You can select any pairs of outcomes. The combination shown in Figure 7.49 should be useful to managers wishing to decide if customers are going to switch more to full suspension bikes instead of hard tail mountain bikes. Clearly, year is important—because of when full suspension bikes were introduced. An income effect shows that customers from cities with lower per capita incomes favor hard-tail bikes—which makes sense because suspension bikes are more expensive. A small population effect indicates that people from

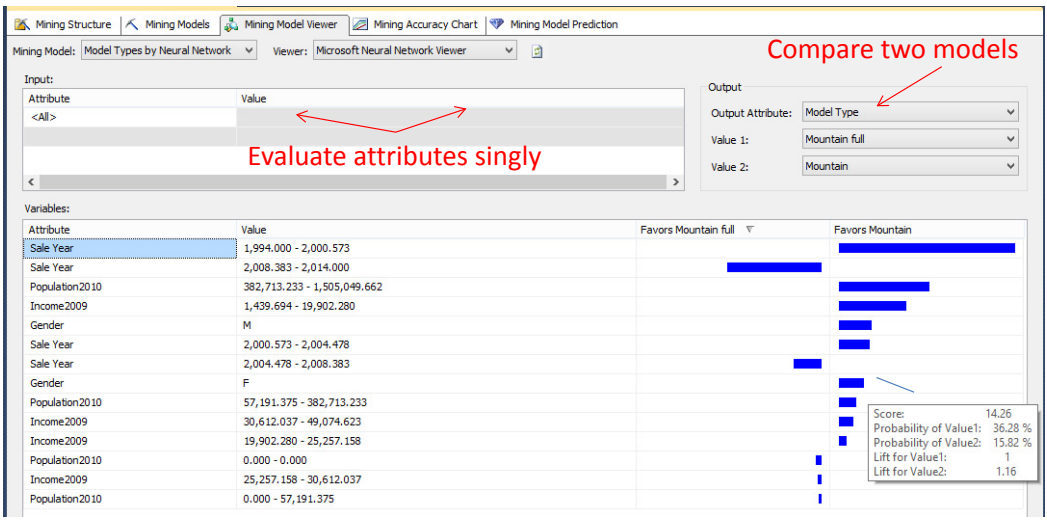


Figure 7.49

Neural Network attribute exploration. This viewer shows the side-by-side comparison of two output values to help judge the impact of various input attributes. The analyst can choose any two attributes for comparison. Rolling over a bar in the chart provides the details in the popup box. It is also possible to specify a single attribute filter and see the impact of the remaining attributes.

larger cities also favor hard tails. To examine the situation of higher-income cities in more detail, change the <All> attribute in the top-left table to Income2009 and select the highest income category in the Value column. You will see that although other attributes influence the strength of the relationship, every case favors simple mountain bikes instead of full suspension, except for years 2008 – 2014.

Prediction

Most tools provide a method to predict the outcome value for a given set of input attributes. The process basically feeds the attributes to the network input neurons and reads the probabilities from the output neurons. It is possible to perform the calculations manually using DMX queries to retrieve the estimated coefficients. However, in most cases, it is easier to use the Microsoft Prediction tool.

Figure 7.50 shows the basic process for the Microsoft prediction tool. For predicting many different input combinations, it is easier to create a special table and enter the input values into that table. Then run the prediction to return the results to that table. SQL and other exploratory tools can be used to examine the results. However, you can right-click and use the singleton query to enter one set of attributes at a time. Switching to the Results view reveals that these attributes predict the customer will purchase a race bike. The Prediction Function (PredictProbability) will supply the probability (35.9%).

The Mining Accuracy Classification Matrix provides more details on the accuracy of the predictions. Figure 7.51 shows the results for the Rolling Thunder Bicycles model type predictions. The biggest issue is the prediction of zero sales of Road bikes. The predictions for Mountain full, Mountain, and Race are not great either, but at least the highest number of cases for each type is correct. Because

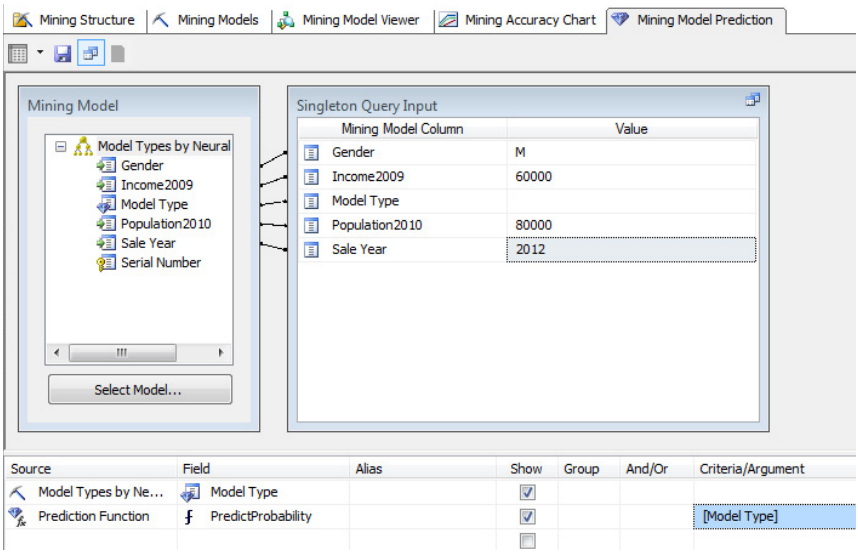


Figure 7.50

Prediction with neural network. Choose the singleton query to enter X-attribute data interactively. Switching to the results view reveals the answer of Race with a probability of 32.8 percent.

Figure 7.51

Mining Accuracy Classification Matrix. Values on the diagonal were the number of cases predicted correctly. The other values are the number of items incorrect for each prediction. For example, the model was wrong when predicting full suspension sales (row) of 1337 that were actually simple mountain bike purchases (column).

Predicted	MTN full (actual)	Track (actual)	Tour (actual)	Mountain (actual)	Hybrid (actual)	Race (actual)	Road (actual)
MTN full	**3015**	0	470	1337	123	2523	2237
Track	0	**0**	0	0	0	0	0
Tour	0	0	**0**	0	0	0	0
Mountain	554	23	206	**708**	149	570	525
Hybrid	0	0	0	0	**0**	0	0
Race	189	1	43	189	12	**295**	201
Road	0	0	0	1	0	1	**1**

Predicted		MTN full (actual)	Track (actual)	Tour (actual)	Mountain (actual)	Hybrid (actual)	Race (actual)	Road (actual)
Full	Logistic	**3326**	0	483	1530	138	2696	2460
	Bayes	**2959**	0	386	1384	102	1667	1982
	Tree	**1901**	0	212	1029	53	1074	1147
	Neural	**3015**	0	470	1337	123	2523	2237
Track	Logistic	0	**0**	0	0	0	0	0
	Bayes	0	**0**	0	0	0	0	0
	Tree	0	**0**	0	0	0	0	0
	Neural	0	**0**	0	0	0	0	0
Tour	Logistic	0	0	**0**	0	0	0	0
	Bayes	0	0	**0**	0	0	0	0
	Tree	0	0	**0**	0	0	0	0
	Neural	0	0	**0**	0	0	0	0
MTN	Logistic	444	21	186	**671**	158	567	426
	Bayes	233	6	144	**438**	118	237	297
	Tree	239	0	76	**530**	110	374	166
	Neural	554	23	206	**708**	149	570	525
Hybrid	Logistic	0	0	0	0	**0**	0	0
	Bayes	0	0	0	0	**0**	0	0
	Tree	0	0	0	0	**0**	0	0
	Neural	0	0	0	0	**0**	0	0
Race	Logistic	111	1	14	83	11	**78**	57
	Bayes	837	16	155	394	62	**1430**	614
	Tree	1773	2	339	577	135	**1849**	1494
	Neural	189	1	43	189	12	**295**	201
Road	Logistic	0	0	0	0	0	0	**0**
	Bayes	0	0	0	0	0	0	**0**
	Tree	0	19	93	83	25	37	**124**
	Neural	0	0	0	1	1	1	**1**

Figure 7.52

Comparison of accuracy by method. Notice that only the Decision Tree method predicts any sales of Road bikes, and the Ree correctly predicts more Race bike sales than the other methods.

of the flexibility of the neural network, it is unlikely that any other method will do substantially better. In other words, the model is likely to be improved only by collecting data on additional attributes.

As always, it is a good idea to run cross validation on the model to ensure that the model is not over fit and too dependent on specific observations. The results are not shown here, but the error rates are strongly consistent across the partitions, so over fitting is unlikely to be a problem.

Model Comparisons

How do the model results compare to each other? The ultimate question is whether one modeling approach is better than the others at identifying attributes and making predictions. In general, there is no single answer. However, if the outcome variable holds continuous data, linear regression has several nice properties; it is easy to use; it is easy to forecast; and the results are easy to explain.

Method	# Correct	Percent	RMSE
Logistic	1915.6	30.5	0.647
Bayes	2175.2	34.6	0.657
Tree	2094.6	33.4	0.668
Neural	1928.8	30.7	0.622

Figure 7.53

Comparison of model predictions. Notice that the Naïve Bayes approach does a better job overall than the other methods on this data.

Prediction

It is easiest to compare models in terms of predictions. Several measures including RMSE and lift are used to evaluate models. It is important that the same items are being measured in each model. Be even more cautious if different software tools are used to compute the measures because they might be defined and computed differently in each tool. Microsoft's prediction matrix is another useful tool when comparing forecasts. Figure 7.52 summarizes the data for the Model Type forecasts created with the four methods for discrete data covered in this chapter. Notice the similarities of the forecasts, except that the decision tree is the only one to predict any sales of Road bikes, and the Decision Tree approach correctly predicts Race bikes much better than the other methods.

Figure 7.53 summarizes the models by examining the number of correct predictions. These predictions were generated from the holdout data sets in the Microsoft BI tool. Notice that the Naïve Bayes and Decision Tree approaches have clearly predicted model types better than the other methods. This result might not hold for other data sets. Notice that overall, none of the models does a great job of predicting the choice of model type. But, the data is somewhat limited. More and better input attributes would be useful—but they can be difficult to obtain.

Another way to examine the predictions is to look at the forecast of each model type for each method. Managers of Rolling Thunder Bicycles probably need to

Figure 7.54

Comparison of model predictions by Model Type. Notice the prediction problems with Race and Road types and the over prediction of full suspension sales.

	Logistic	Bayes	Decision Tree	Neural Net	Actual Sales
MTN full	79.0	63.0	40.2	72.6	28.41
Track	0.0	0.0	0.0	0.0	0.0
Tour	0.0	0.0	0.0	0.0	4.62
Mountain	18.4	10.9	11.1	20.5	7.32
Hybrid	0.0	0.0	0.0	0.0	4.58
Race	2.6	26.1	45.8	6.9	31.02
Road	0.0	0.0	2.8	0.0	24.05

Logistic	Sale Year	Gender	Population	Income
Bayes	Gender	Sale Year	Population	
Decision Tree	Sale Year	Gender		
Neural Net	Sale Year	Population	Income	Gender

Figure 7.55

Summary of attributes by method. The most important attribute is listed first. This list was derived by examining the comparison between Mountain full and Mountain model types.

know which models will be popular next year. It is one of the reasons for undertaking this analysis. Knowing which models will sell the most will make it easier to order the correct number of components at the start of the year. To obtain the best forecast, you should first estimate all of the background variables (income, population, gender, and sale year); then enter those values into the prediction equations for each model. However, for an initial rough pass, Figure 7.54 shows the average values taken from the classification matrix data which uses all years. The actual sales by model type were computed using a simple SQL query for 2012. The purpose of this table is to look at the prediction with less detail. If managers do not care about income, population, gender, and year; are some of the models more accurate than others? The answer is challenging. All of them are over predicting the sales of Mountain full bikes. Logistic and Neural network both over predict the sales of Mountain bikes at the expense of Race bikes. All of the models are missing the sales of Road bikes. The Decision Tree predicts the most number of race bikes, but basically over-predicts those as well.

The basic problem with all of the predictions is that the underlying model needs more attributes and better data. The methods are doing as well as possible, but ultimately more data needs to be collected for additional attributes to improve the model. Alternatively, it is possible that the underlying consumer decision of which model to purchase has a large random element. Some decisions and events simply cannot be predicted with much accuracy. In this particular case, more data on individual customers would be nice but it is not available. However, there are other ways to get a better prediction model. Looking at the average sales for 2012 seems to provide better estimates than anything else, which implies that a better model could be built by examining sales over time. These techniques are covered in the chapter on time series. If you assume that sales for next year will be roughly based on sales for the current year, the forecasts should improve dramatically. But the point of this chapter was to examine how specific attributes affect the choice of model types, not to provide the most accurate forecast possible.

Attribute Evaluation

With Microsoft BI, each of the estimation tools has a slightly different method of presenting the effect of the X-attributes. These differences provide different perspectives on the data and examining all of them should help the analyst see a bigger picture of the data. However, the differences make it more difficult to compare the methods. If all of the methods returned numerical coefficients on the attributes—similar to traditional logistic or linear regression, a formal comparison of the attributes is easy.

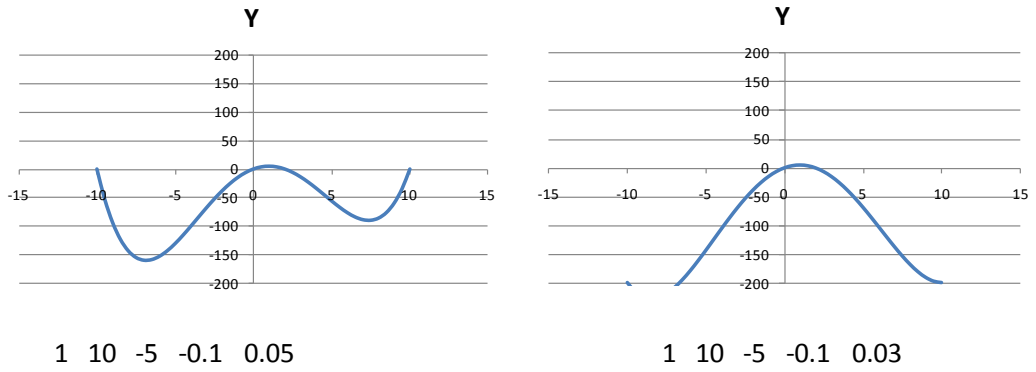


Figure 7.56

Instability with nonlinear coefficients. The charts show two versions of similar quartic functions. The only difference is 0.02 in the last coefficient. If your estimate of that coefficient is slightly off, you will obtain radically different forecasts.

Figure 7.55 summarizes the attribute evaluations of the four main discrete-attribute methods examined in this chapter. The most important attribute is listed first. For some methods, specific values of the attributes are more important than others—such as years after 1997 for Mountain full. The list was generated by finding the most important attributes when comparing Mountain full to the Mountain model type. In most of the methods, these rankings are relatively consistent across model types, with some variation in the lower level spots. However, Naïve Bayes is an exception. In the general model, Gender takes the top spot for influencing the choice of model type. This result is important to know because Naïve Bayes was better than the other methods for predicting sales of Race bicycles.

Nonlinear Complications

Neural networks are powerful tools that can estimate complex relationships. However, nonlinearity can cause problems. To begin, nonlinear relationships are difficult to understand and hard to explain to others. A basic exponential or log function might be close enough to linear to work, but complex interactions among attributes quickly become meaningless.

More importantly, higher-order nonlinear equations can be unstable in the sense that large changes in the outcome variable arise from tiny changes in the coefficient estimates. Figure 7.56 illustrates the problem with a basic fourth-degree polynomial function. The two equations are different by only 0.02 in the last coefficient. Even tiny errors in estimating the function can lead to drastic changes in interpretation and prediction. And this chart uses only a single variable. The problems magnify when the attributes intermix through a nonlinear equation.

Forecasting is the other classic problem with nonlinear functions. Many start-up firms exhibit exponential growth for the first few quarters or even a couple of years. A nonlinear function does a good job of predicting the historical growth. But, exponential growth patterns are extraordinarily hard to maintain. At some point, the organization simply cannot maintain exponential change and it reverts

to a linear growth (or crashes completely). Consequently, even if a nonlinear function does a great job of predicting past growth, it is often safer to fit a conservative linear function for prediction. Ultimately, the regression, naïve Bayes, or decision tree forecast can be more accurate than a nonlinear forecast generated from a neural network. This statement does not mean you should never use a neural network—but you should be cautious when using the model for predicting beyond the immediate future.

Summary

It is important and helpful to identify which attributes influence a predictable outcome variable. Knowing which attributes and which values influence others is a critical step in building models and predicting what will happen in the future. If the relationships are strong enough, you might be able to recognize causality. If the overall model is good enough (has a sufficiently small error), you will be able to predict what will happen in new situations. As long as you can measure the X-attributes, you can plug them into the estimated model and forecast a value or find the probabilities of any outcome occurring. For example, if you develop a model that relates personal attributes such as job, income, and education to loan payments; you can use the model to predict whether the next person applying for a loan will make payments on time. Almost any business model raises similar questions. The key is to identify the outcome variable and then find data on various attributes that might affect that variable. Once enough data is collected, the data mining tools will find and measure correlations and provide prediction estimates.

Linear regression is a powerful tool used in science to test hypotheses and evaluate claims. It is heavily used in economics and forecasting. In the context of data mining, it provides good measures of attribute coefficients, but the outcome variable must measure continuous data. Simple regression does not work if the dependent variable contains discrete categorical values. Logistic regression is one of the first tools developed to handle categorical dependent variables. The traditional method estimates linear equations for the effect on each outcome of the dependent variable. The Naïve Bayes tool uses Bayes' Theorem to use data observations to update a prior distribution and create a posterior distribution that more accurately describes the data. The probability distribution can identify the importance of each input attribute and can be used to predict the probability of any given outcome. Decision Trees are created to define a path through the attribute values. Each path results in a different outcome and highlights the values of the X-attribute data nodes that have a significantly different impact on the outcome. Decision tree results are good for understanding and explaining relationships.

Neural networks are probably the strangest evaluation method of the group. Loosely based on human brains, a neural network consists of three layers of neurons: Input, Hidden, and Output. Training the network consists of applying the sample data to estimate the weights of the connections between layers and to define the threshold firing value for a given neuron. These estimated weights define a nonlinear network that relates the input values to the outcomes of the predictable variable. The weights make it difficult to understand the contribution of each attribute, but the network can be used to predict the likely outcome of any new data.

All of the tools in this chapter are commonly used in data mining. Often, an analyst will apply all of the tools to the same data set—because each tool provides a slightly different perspective on the relationships. Comparing the results both in terms of prediction and the importance of each X-attribute provides insight and knowledge into the entire process.

Key Words

bootstrapping
causality
classification
classification matrix
conditional probability
contingency table
correlation
decision tree
decision tree
dependent
discretized data
elasticity
independent
lift
linear regression
logistic regression
maximum likelihood estimator (MLE)
mean absolute deviation (MAD)
model
naïve Bayes
named query
neural network
posterior distribution
prediction
prior distribution
root mean square error (RMSE)
Shannon's entropy

Review Questions

1. Why is it useful to evaluate the effect of attributes on outcome variables?

2. How is missing data handled by each of the tools in this chapter: regression, logistic regression, naïve Bayes, decision trees, and neural networks?

3. How is data organized for all of the tools in this chapter?

4. What is the role of the key attribute in data sets for Microsoft BI tools?

5. What do the coefficients mean in linear regression results?

6. How is logistic regression different from linear regression?

7. What is the fundamental assumption of the naïve Bayes method?

8. What is the primary strength of the decision tree method?

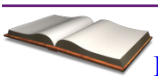
9. How is the decision tree method different from the naïve Bayes approach?

10. How is a neural network result different from the other tools?

11. What is the problem of over fitting and how is it tested?

12. Should one of the methods in this chapter be preferred over the others?

Exercises



Book

1. Set up and run the linear regression example from the chapter, using both a standalone program and the data mining tool. Summarize the results and interpretation.

2. Set up and run the logistic regression example from the chapter. Summarize the results and interpretation.

3. Set up and run the naïve Bayes example from the chapter. Summarize the results and interpretation.

4. Set up and run the decision tree example from the chapter. Summarize the results and interpretation.

5. Set up and run the neural network example from the chapter. Summarize the results and interpretation.

6. For the main problem in the book with ModelType as the predictable variable, identify other attributes that would be good candidates for analysis. For items not in the original database, how would you obtain values?

Rolling Thunder Database

7. Expand the analysis of total sales by city and see if you can identify attributes that reliably predict sales. What additional data might you want to collect?

8. Looking at the customers who purchased more than one bicycle from the Rolling Thunder Bicycles. What attributes do they possess—particularly which ones could be used to target a marketing campaign at other customers?

9. Are there attributes that affect whether customers buy bicycles through retail stores or directly from the company? Look at StoreID values of 1 and 2 versus the rest.

10. Examine the patterns of sales for carbon fiber versus aluminum and steel bikes. The bike can be classified by the type of material used in the down tube.

11. Managers want to increase prices and profit margins on Campagnolo (Campy) equipped bicycles. Who will be affected by this change? Or, what attributes lead people to purchase Campy-equipped bikes over Shimano?

Diner

12. The managers want to know how to increase the sales of desserts (without changing the price). Who buys desserts now? Who does not?

13. What factors affect the total amount of a bill?

Corner Med

14. What factors affect the total amount of revenue per visit? Hint: Consider at least the number of visits, procedures per visit, patient demographics, and the insurance company.

15. Are there factors that affect which physician treats a patient? For example, is it affected by diagnosis code or procedure?

Basketball

Note: Every team is listed twice in the Games table. Load and create the view: TeamGameTotals which links to BaseTeam to eliminate the duplicates. Also, pick one season to answer each question.

16. What attributes affect whether a team wins a game?

17. What attributes affect the number of points scored in a game? In particular, how many points is the home court worth?

18. Are some divisions better or worse than others in terms of winning? What about in terms of total points scored?

19. Which players and player characteristics were key to wins by the LA Lakers?

20. Did importance factors change for the Lakers in the playoffs? For example, were some players more (or less) important in the playoffs than in the regular season?

Bakery

21. Determine how the month and day-of-week (DOW) impact the sales of products by category.

Cars

22. Do any of the attributes affect the price of the vehicles?

23. Which attributes affect the acceleration (SecTo60)?

Teamwork

24. Using the basketball database, each person in the group should choose one team and determine which player statistics affect that team's ability to win.

25. Using the Rolling Thunder Bicycles database, assign one technique from this chapter to each team member and find the best model for predicting the selection of model type.

Additional Reading

Heckerman, David, Geiger, Dan, and David M. Chickering, 1995, *Learning Bayseian Networks: The Combination of Knowledge and Statistical Data*, Microsoft Technical Report MSR-TR-94-09, Microsoft: Redmond. http://research.microsoft.com/apps/pubs/default.aspx?id=65088. [The technical description of Microsoft's Bayesian and Decision Tree methods.]

Intriligator, Michael D. 1978, *Econometric Models, Techniques, and Applications*, Prentice-Hall, Englewood Cliffs. [Basics of econometrics with good coverage of logistic and probit methodologies.]

Judge, George G., W.E. Griffiths, R. Carter Hill, Helmut Lütkepohl, and Tsoung-Chao Lee, 1985, *The Theory and Practice of Econometrics, second edition*, Wiley: New York. [A classic complete work on econometric theory for those who want to know how to handle problems that arise with regressions. Graduate level.]

Rish, Irina, 2001, "An Empirical Study of the Naïve Bayes Classifier," IBM Research Report RC 22230 (W011-014). http://www.research.ibm.com/people/r/rish/papers/RC22230.pdf. [An analysis of factors that affect the performance of the Bayes classifier.]

Rumelhart, David E. and James L. McClelland, 1986, *Parallel Distributed Processing, Vol. 1 and 2*, MIT Press: Cambridge, Massachusetts. [A classic collection of work on the foundations and start of neural networks. Includes descriptive and technical content.]

Trevor Hastie, Robert Tibshirani, and Jerome Friedman, 2009, *The Elements of Statistical Learning/2e*, Springer: New York. [An outstanding book on data mining, with an emphasis on theory. A graduated-level book that requires a strong mathematics background.]

Zellner, Arnold, 1971, *An Introduction to Bayesian Inference in Econometrics*, Wiley: New York. [A classic book on Bayesian theory, particular focus on subjective probabilities and how they can define traditional analyses. Graduate level with mathematics.]