Chapter 3 Predictive Modelling

3.1 Introduction

Data mining explores data and attempts to discover patterns, trends and relationships. By so doing, it can transform raw data into useful information for decision making. Many tools are available in data mining to analyse data. As discussed in the previous chapter, these include description and visualisation tools, association analysis tools and clustering tools.

This chapter focuses on predictive modelling. In particular, it discusses regression (a traditional statistical method), neural networks (an artificial intelligence model) and decision trees (a machine learning technique). The next chapter discusses data mining issues that are encountered in the predictive modelling context.

The most common and important applications in data mining usually involve predictive modelling, which can be further categorised into two major categories. Classification refers to the prediction of a target variable that is qualitative (i.e., categorical) in nature (e.g., predicting fraud versus non-fraud, high-risk versus low-risk or purchase versus non-purchase). Estimation, on the other hand, refers to the prediction of a target variable that is quantitative (i.e., continuous) in nature (e.g., predicting the amount spent, duration of a call or account balance).

More precisely, whether a variable is qualitative or quantitative depends on the scale of measurement. Measurement can be considered an assignment of numbers to characteristics, attributes or features. For example,

we can measure a customer's age, gender and propensity to purchase by assigning numbers to these.

The nominal scale of measurement assigns numbers that serve only the purpose of classification or identification. For instance, in measuring gender, the number "1" can be assigned to females and the number "2" to males. These numbers identify the gender of a person by classifying the person as female or male. It is not meaningful to subject these numbers to the basic mathematical operations of addition, subtraction, multiplication and division (e.g., it is not meaningful to say that the average gender among a group of customers is 1.58).

The ordinal scale of measurement assigns numbers that also serve the purpose of classification. In addition, these numbers can be meaningful ranked. In measuring age, for example, the number "1" can be assigned to young respondents (less than 25 years old), the number "2" to middle-aged respondents (from 25 years old to less than 50 years old) and the number "3" to old respondents (who are 50 years old and above). These numbers classify the age of a person as one of three groups. Further, the three age groups can be ranked in some meaningful ascending or descending order. As in the case of nominal measurement, it is not meaningful to subject the numbers in ordinal measurement to the basic mathematical operations of addition, subtraction, multiplication and division.

Variables that are measured on a nominal or ordinal scale are qualitative or categorical variables. They can also be referred to as non-metric variables. The term "non-metric" will be used from this point onwards to refer to such variables.

The interval scale of measurement assigns numbers such that the intervals between numbers can be meaningfully interpreted. For example, budget performance can be measured as the difference between actual sales and budgeted sales. Suppose managers A, B, C and D have budget performance of $800, $600, $300 and $100, respectively. In this illustration, the

difference in budget performance between managers A and B is $200 (i.e., $800 - $600) and the difference in budget performance between managers C and D is also $200 ($300 - $100). The interval of $200 can be meaningful interpreted and the first $200 (i.e., the difference between the budget performance of managers A and B) is the same as the second $200 (i.e., the difference between the budget performance of managers C and D).

In an interval scale of measurement, while intervals between numbers (i.e., addition and subtraction) can be meaningfully interpreted, ratios of the numbers (i.e., multiplication and division) cannot be meaningfully interpreted. The reason is that the zero point (0) is arbitrary and/or non-natural. In the above illustration, zero for A is not the same as zero for B because it is determined relative to the budgeted sales (which may not be the same for A and B). The implication is that changing the zero point does not affect intervals between numbers but it does affect ratios of numbers measured on an interval scale. For example, when budgeted sales equals $1000, actual sales of $4500 and $3000 indicate a difference in budget performance of $1500 (i.e., [4500 – 1000] – [3000 – 1000]) and a ratio of 1.75 (i.e., 3500/2000). However, if the budgeted sales is $2000, the difference is still the same (i.e., [4500 – 2000] – [3000 – 2000] = $1500) but the ratio is now 2.5 (i.e., 2500/1000).

The ratio scale of measurement applies where there is a non-arbitrary and/or natural zero, where zero reflects the absence of a characteristic, attribute or feature. For instance, a customer's expenditure can be measured on a ratio scale because zero expenditure means no expenditure. Ratios can be meaningfully computed for variables measured on a ratio scale. Therefore, the ratio of 2:1 means that the first expenditure is two times the second expenditure for the following pairs of numbers: $200 and $100, $700 and $350, or $4000 and $2000. Therefore, in the case of ratio measurement, it is meaningful to subject the numbers to the basic mathematical operations of addition, subtraction, multiplication and division.

Variables that are measured on an interval or ratio scale are quantitative variables. They can also be referred to as metric variables. The term "metric" will be used from this point onwards to refer to such variables.

Given the above, classification refers to the prediction of a non-metric target variable and estimation to the prediction of a metric target variable. Both classification and estimation fall under the major category of predictive modelling.

Generally, predictive modelling attempts to predict a target variable (also called a dependent variable) on the basis of one or more input variables (also called independent variables). For ease of reference, the terms "target" and "input" variables will be used from this point onwards.

Three data mining tools are commonly used for predictive modelling, namely, regression, neural networks and decision trees. Usually, all three are used and the results are evaluated to identify the best model.

3.2 Regression Models

Regression is a traditional statistical method of analysis that can be used for predictive modelling. In particular, multiple regression can be used to predict metric target variables while logistic regression can be used to predict non-metric target variables. Input variables are used in a regression model to do the prediction. For example, the input variables age, gender and income can be used to predict the amount of expenditure on a particular product (metric target variable) or whether a customer will purchase the product (non-metric target variable).

3.2.1 Multiple Regression

In the simplest form, a regression model has a target variable and an input variable. Suppose that an organisation wishes to predict the amount of

expenditure on a particular product on the basis of the age of customers. Suppose further that expenditure and age data have been collected for a sample of customers and can be plotted in a scatter plot as shown in Figure 3.1.

The scatter plot suggests a linear (i.e., straight line) and positive relationship between expenditure and age. The relationship is positive in that an increase in age is associated with an increase in expenditure, and vice versa. The regression model attempts to draw a line through the points to capture this relationship. The regression model can be expressed mathematically as $\hat{y} = b_0 + b_1X_1$, where $\hat{y}$ = predicted expenditure, X_1 = age and b_0, b_1 = regression coefficients. Graphically, the regression line can be drawn as a straight line passing through the points, as shown in Figure 3.2. It is noted that b_0 is the intercept and b_1 is the slope (i.e., gradient) of the regression line.

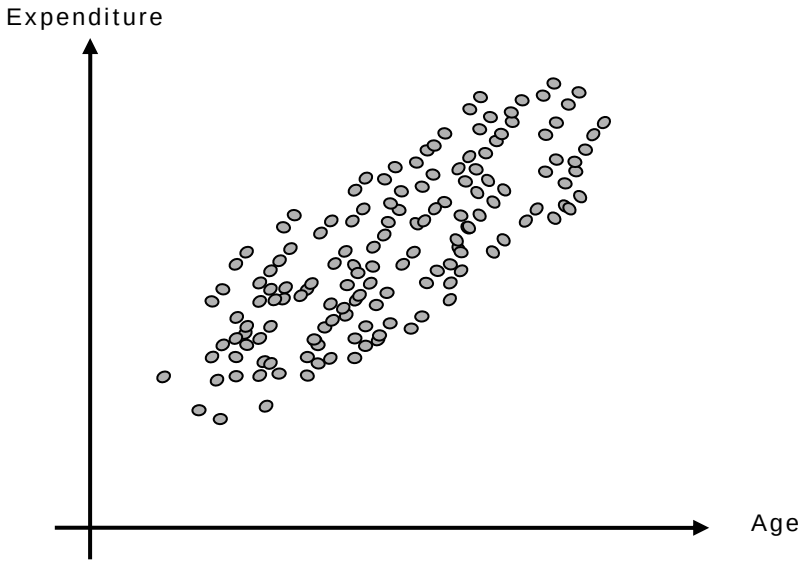


Figure 3.1 Scatter Plot of Expenditure and Age

The intercept (b_0) can be interpreted as the value of y (i.e., expenditure) when X_1 (i.e., age) is zero. However, this interpretation is appropriate only if the points in the scatter plot spread across the point $X_1 = 0$. Otherwise, there is no empirical evidence to support a particular value of y when X_1 is zero. Also, sometimes it is not meaningful for an input variable to be 0. In such instances, the intercept can be interpreted as an adjustment factor to improve the prediction of y. The slope (b_1) can be interpreted as the change in y (i.e., expenditure in this example) when there is a one-unit change in X_1 (i.e., age).

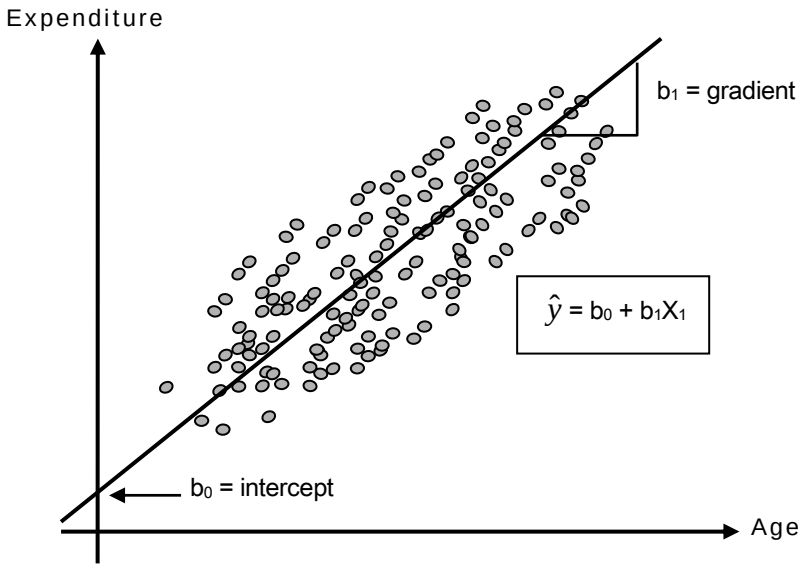


Figure 3.2 Regression of Expenditure on Age

The regression coefficients b_0 and b_1 are estimated so as to minimise the sum of squared errors. Hence, the estimation procedure is commonly referred to as Ordinary Least Squares (OLS). An error (e) can be defined as

the difference between the actual value of y and the predicted value of y. That is:

$$e = y - \hat{y}$$

Therefore, a squared error is:

$$e^2 = (y - \hat{y})^2$$

and the sum of squared errors is:

$$\sum e^2 = \sum (y - \hat{y})^2$$

where the summation is made over all the points in the scatter plot (i.e., all the observations in the data set).

It is noted that the sum of squared errors $\sum e^2$ is a function of the regression coefficients b_0 and b_1 (via $\hat{y}$). In other words, given a particular set of values for b_0 and b_1, $\hat{y} = b_0 + b_1X$ can be computed and hence e can be determined. That is, $\sum e^2$ can be determined. Therefore, calculus (i.e., differentiation) can be applied to find the values of b_0 and b_1 that will minimise $\sum e^2$. This concept underlies OLS estimation.

With the regression model $\hat{y} = b_0 + b_1X_1$, it is possible to predict the values of y (i.e., expenditure) given values of X_1 (i.e., age). However, before using the regression model for prediction, it is appropriate to first assess the adequacy of the model. One good adequacy measure is the statistical significance of the regression model. If no regression model is available to help predict y, then the best prediction of y is the average of y (denoted as $\bar{y}$). That is, the average expenditure of the sample is the best estimate of an individual's expenditure (assuming that the sample is reflective of the population of interest).

On the other hand, the regression model discussed here can predict y (expenditure) on the basis of X_1 (age). Hence, one way to test the adequacy of the regression model is to evaluate if the regression model (i.e., $\hat{y}$) gives significantly better predictions of expenditure as compared to just using the average expenditure ($\bar{y}$) to predict expenditure. This is done in regression analysis by the model F test. The p-value associated with this test indicates the degree of statistical significance of the regression model. Traditionally, p-values

of 0.05 or lower show a significant regression model. Other benchmarks such as 0.01 or 0.10 can also be used. The smaller the benchmark (denoted by α in statistics), the more stringent the assessment of model adequacy. The model F test is equivalent to testing if all the regression coefficients (not including the intercept) are equal to zero. If all the regression coefficients in the regression model (not including the intercept) are equal to zero, then the regression model is not useful in predicting y. Generally, a statistically significant regression model suggests an adequate model.

Another good adequacy measure is R-square (also called the coefficient of determination). In the current example, R-square indicates the changes in y (i.e., expenditure) that can be explained by the changes in X_1 (i.e., age). Thus, R-square looks at how well changes in age can help predict (or explain) changes in expenditure. It ranges from 0 to 1, with 0 indicating no explanatory power and 1 perfect explanatory power. If R-square is 0, the points on the scatter plot will be randomly distributed; if R-square is 1, the points on the scatter plot will all lie on the regression line. A possible rule-of-thumb is to consider a regression model as adequate if R-square is at least 0.6.

R-square increases when more input variables are added to the regression model and/or when the number of observations is decreased. Hence, to take into account the number of variables and the number of observations, the adjusted R-square can be computed. This adjusted measure is appropriate when comparing regression models with, say, different numbers of variables.

Suppose that for the current example, the regression model is:

Predicted Expenditure = 500 + 30*Age

Then, if an individual is 40 years of age, the predicted expenditure is $1700 (i.e., 500 + 30*40). The coefficient of 30 for age can be interpreted as the expected increase in expenditure given a 1-unit increase in age (in this case, a 1-year increase in age). It is also possible to test if the regression coefficient is statistically significant; that is, if it is significantly different from zero. A zero

coefficient would mean that age has no effect on expenditure. The coefficient test can be done by either a t test or an F test (they are equivalent in that the test statistics $F = t^2$). As in the case of a model test, a p-value is associated with the coefficient test and it can be used to assess statistical significance by comparing it to α (see earlier discussion). A statistical test is significant if its p-value is equal to or less than α. In such an instance, the regression coefficient is significantly different from 0 and the regression model is useful in predicting (or explaining) the target variable.

A regression model can be easily expanded to include more than one input variable. For example:

Predicted Expenditure = $b_0 + b_1$*Age + b_2*Income + b_3*Gender

Regression can incorporate non-metric input variables (e.g., gender) provided these variables are coded as dummy variables that take on only values of 0 and 1 (e.g., 0 = female and 1 = male). In this respect, a non-metric variable with c number of categories needs only (c – 1) number of dummy variables. To illustrate, the races Chinese, Malay, Indian and Others can be coded as follows:

	Dummy Variables		
	D1	D2	D3
Chinese	1	0	0
Malay	0	1	0
Indian	0	0	1
Others	0	0	0

As expected, dummy variables D1 to D3 take on only values 0 and 1. In this coding scheme, “Others” is the reference group (where all the dummy variables equal zero) and the coefficients of D1 to D3 are interpreted relative to the reference group. Therefore, the coefficient for D1 represents the average difference in the target variable (e.g., expenditure) between Chinese and Others. Other coding schemes for dummy variables are also possible besides the standard (0, 1) coding.

More generally, a regression model when expanded to include more than one input variable can be expressed as follows:

$$\hat{y} = b_0 + b_1X_1 + b_2X_2 + \ldots + b_pX_p$$

where there are p number of input variables. This model is commonly referred to as multiple regression. The regression concepts and statistics discussed earlier (e.g., OLS estimation, model F test, R-square, individual t tests … etc.) also apply to multiple regression.

Where there are many input variables, stepwise regression procedures are available to select the most significant variables in the final model. For example, the forward selection procedure selects the most statistically significant variable (deemed to be the most important variable) at the first step, the next most statistically significant variable at the second step and so on. This procedure stops when there is no more statistically significant variable to select or when the number of input variables in the final regression model is deemed sufficient. Alternatively, instead of selecting the most important variables, the backward elimination procedure progressively removes the least important variables, starting from a full model with all the potential variables included in the model. This procedure stops when there is no more insignificant variable to remove.

Finally, it is noted that the regression model assumes that the errors (i.e., [y - $\hat{y}$]) are distributed identically and independently as a normal distribution with equal variances. More discussion on the regression assumptions can be found in statistics textbooks such as Afifi and Clark (1996).

3.2.2 Logistic Regression

Multiple regression can be used to predict a metric target variable. If the target variable is non-metric, however, then logistic regression is appropriate. One way to view logistic regression is to think of it as a modified version of multiple regression. Suppose that the target variable in the following regression model

is coded as 0 and 1, which is taken to be the probability of an event occurring (say, the purchase of a particular item):

$$\hat{y} = b_0 + b_1X_1 + b_2X_2 + \ldots + b_pX_p$$

Given a data set, the regression coefficients can be estimated but there will be two problems with such a regression model. Firstly, the assumptions underlying the regression model (e.g., normality of the target variable) are likely to be violated. This may lead to incorrect results. Secondly, there is no assurance that $\hat{y}$ (as probability estimates) will be within the range of 0 to 1. That is, $\hat{y}$ from the regression model can take on values below 0 and above 1. This reduces the usefulness of $\hat{y}$ as probability estimates as the probability of an event occurring ranges only from 0 to 1. Although, it can be argued that all $\hat{y}$ with values below 0 can be set to 0 and all $\hat{y}$ with values above 1 can be set to 1, this is not a satisfactory solution as it biases the estimates of the regression coefficients.

One good way to circumvent the problems discussed above is to interpret $\hat{y}$ as a theoretical index that is related to the probability of an event occurring and not as a probability estimate in itself. That is, the model:

$$\hat{y} = b_0 + b_1X_1 + b_2X_2 + \ldots + b_pX_p$$

predicts a theoretical index which is not limited by the range of 0 to 1. In addition, this theoretical index can be transformed to a probability estimate via a cumulative probability distribution. In particular, logistic regression uses the cumulative logistic distribution. This is shown in Figure 3.3.

As can be seen, all values of the theoretical index $\hat{y}$ can be transformed into a probability estimate within the range of 0 to 1. Recall that the objective of logistic regression is to predict a non-metric target variable. Suppose that a logistic regression model is constructed to predict whether an individual will be a purchaser of a particular product. From the logistic regression model, a theoretical index can be computed. This index can then be transformed via a cumulative logistic distribution into a probability estimate of that individual being a purchaser. To give a classification (i.e., non-metric

prediction), the estimated probability of being a purchaser can be compared to a cut-off probability. For simplicity, assume that 0.5 can be used as a cut-off probability. Then, any individual whose estimated probability is 0.5 or above will be predicted as a purchaser. Conversely, any individual whose estimated probability is below 0.5 will be predicted as a non-purchaser.

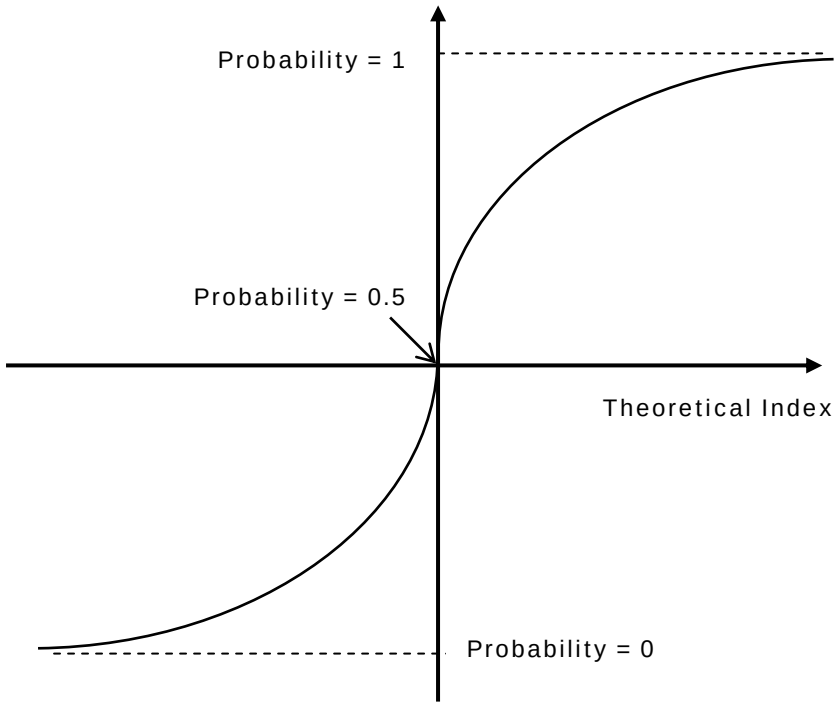


Figure 3.3 Transformation via a Cumulative Logistic Distribution

Each estimated probability is associated with a particular value of the theoretical index and versa vice (see Figure 3.3). Therefore, a cut-off probability is also associated with a cut-off value of the theoretical index. This

means that to classify an individual as purchaser/non-purchaser, there is no need to transform the theoretical index into an estimated probability. Comparing the estimated theoretical index predicted by the logistic regression model (i.e., $\hat{y}$) with the cut-off theoretical index is sufficient for the classification. However, computing the estimated probability generates additional information. In practice, the computation of the (predicted) theoretical index and estimated probability is done by computers.

In logistic regression, the theoretical index has a special interpretation. It is the natural logarithm of the odds. Odds is the ratio of the probability of an event occurring (e.g., a purchase) to the probability of the event not occurring (e.g., a non-purchase). For example, if the probability of purchase is 0.8 and non-purchase is 0.2, then the odds of purchase to non-purchase is 4 (i.e., 0.8/0.2). Taking natural logarithms, ln(4) is 1.3863, which is the value of the theoretical index. (Natural logarithms are logarithms taken to the base of e, which is a mathematical constant that is equal to 2.7183. This "e" is different from the error term e in multiple regression). While multiple regression uses linear estimation (i.e., the OLS estimation procedure), logistic regression uses non-linear estimation. In particular, it uses the maximum likelihood estimation procedure. More information of this procedure can be found in Greene (2003).

Mathematically, in the logistic regression model, the predicted probability of the event occurring is:

$$\text{Predicted Probability of Event} = 1/[1 + e^{-(b_0 + b_1X_1 + b_2X_2 + \ldots + b_pX_p)}]$$

The predicted probability of the event not occurring (i.e., the non-event) is just one minus the predicted probability of the event occurring. As noted earlier, e = 2.7183.

As in the case of multiple regression, it is appropriate to first assess the adequacy of the logistic regression model before using it for prediction. The statistical significance of the logistic regression model can be evaluated by looking at the model chi-square test statistic and its associated p-value. The model chi-square test evaluates if all the model coefficients are zero, in which

case the model is useless in predicting the target non-metric variable. If no logistic regression model is available to help predict y (say, purchase versus non-purchase), then the best prediction of y is the prior probabilities of y (i.e., the probabilities of purchase and non-purchase in the population of interest). Hence, the model test can be deemed to be a test of whether the logistic regression model classifies observations significantly better than a crude "model" that is based on prior probabilities.

The p-value associated with the model chi-square test indicates the degree of statistical significance of the logistic regression model. It is usually compared with the traditional benchmarks of 0.01, 0.05 or 0.10. The smaller the benchmark (or α), the more stringent the assessment of model adequacy. The model is statistically significant (and hence useful) if the p-value is equal to or less than α.

In addition, measures similar to R-square in multiple regression are also available in logistic regression. As before, the higher the R-square measures, the better the model fits the data and hence, the better the model can classify the observations (i.e., predict the non-metric target variable). Another very common and good way to assess the adequacy of the logistic regression model is to look at its accuracy rates in classification. The accuracy rates are usually presented in the form of a classification table (also called a confusion matrix). A typical classification table is shown in Table 3.1. The occurrence of interest (e.g., purchase or fraud) is designated as an event.

As shown in Table 3.1, A and D represent the number of correct predictions (or classifications) for events (e.g., purchase) and non-events (e.g., non-purchase), respectively. Therefore, the accuracy rate of events is A/(A + B) and the accuracy rate of non-events is D/(C + D). The total number of correct predictions is (A + D); that is, the overall accuracy rate of the logistic regression model is (A + D)/(A + B + C + D). It is also possible to compute error rates instead of accuracy rates. In this case, the focus will be on the B (events

incorrectly predicted as non-events) and C (non-events incorrectly predicted as events).

Table 3.1 Classification Table of Logistic Regression Model

Actual Status	**Predicted Status**		**Total**
	Event	Non-event	
Event	A	B	(A + B)
Non-event	C	D	(C + D)
Total	(A + C)	(B + D)	(A + B + C + D)

To crystallise the concepts, suppose that a logistic regression model is constructed to classify individuals as purchasers or non-purchasers of a particular product. Then, the classification table summarising the model results might look like Table 3.2. As shown, the accuracy rates for purchase and non-purchase are 75% and 80%, respectively. Conversely, the error rates for purchase and non-purchase are 25% and 20%, respectively. The overall accuracy rate of the model is 76.92%; that is, the overall error rate is 23.08%. The organisation developing the prediction model will have to decide if the accuracy rates are acceptable before using the model. More discussion on accuracy rates will be presented in the next chapter.

Individual chi-square tests are also available to test if each of the model coefficients is statistically significant; that is, if it is significantly different from zero. A zero coefficient would mean that the variable is not useful in predicting y. Again, the p-value associated with the coefficient test is used to assess statistical significance by comparing it to α. A statistical test is significant if its p-value is equal to or less than α. This indicates that the model coefficient is significantly different from zero and therefore that input variable is useful in predicting or explaining the non-metric target variable y.

Table 3.2 Accuracy and Error Rates of Logistic Regression Model

Actual Status	Predicted Status		Total
	Purchase	Non-purchase	
Purchase	600 (75%)	200 (25%)	800 (100%)
Non-purchase	100 (20%)	400 (80%)	500 (100%)
Total	700	600	1300
Overall accuracy rate = 1000/1300 = 76.92%			

In logistic regression, the interpretation of model coefficients is less straightforward compared to the case of multiple regression. Suppose that a logistic regression model to predict purchase/non-purchase has only two input variables: age (measured in number of years) and gender (coded as 0 for male and 1 for female; i.e., male is the reference group). Then, the model coefficient for age (a metric input variable) relates to the change in the (natural) logarithm of the odds (of purchase versus non-purchase) for a 1-unit (i.e., 1-year) change in age. Also, the model coefficient for gender (a non-metric input variable) relates to the change in the (natural) logarithm of the odds (of purchase versus non-purchase) for a female compared to a male (i.e., the reference group). While analysing the magnitude of the impact of an input variable on the target variable is not so straightforward, determining the direction of the impact is easier. Generally, a positive coefficient indicates a positive impact on the probability of the event and a negative coefficient indicates a negative impact on the probability of the event. In any case, the interpretation of magnitude and direction is meaningful only for variables that are statistically significant.

Finally, the logistic regression model can be expanded to incorporate a multichotomous target variable (i.e., a target variable with more than two categories). For example, credit cardholders can be classified as having a low,

medium or high risk of default. Also, stepwise logistic regression procedures are available too. A good discussion of logistic regression can be found in Afifi and Clark (1996).

3.2.3 Illustration of Regression Model

This illustration continues with MailPurchase, a mail order company with a database of 1400 customers. Recall that for each customer, the following data are captured:

1) Status: whether the customer has purchased a promoted product in any of the quarterly marketing campaigns last year;
2) Expend: average monthly expenditure on the company's products last year;
3) Numpur: average number of purchases per quarter last year;
4) Age: age of customer as at 1 January last year;
5) Gender: gender of customer;
6) Income: annual income of customer as at 1 January last year (in $'000);
7) Race: race of customer;
8) Marital: marital status of customer as at 1 January last year; and
9) Member: whether the customer is a member of the loyalty card programme last year.

(More details are given in section 2.2.1 of Chapter 2).

Suppose that to develop the next marketing campaign, MailPurchase is interested to target only existing customers with a high probability of purchase. Hence, it is interested to classify existing customers as likely purchasers or non-purchasers. To construct this prediction model, MailPurchase has decided to use "status" as the target variable and the other variables as input variables. That is, the input variables comprise both purchasing patterns (namely, expend and numpur) and demographic characteristics (namely, age, gender, income, race, marital and member). From the prediction model, MailPurchase will be

able to predict the probability of purchase and hence be able to classify existing customers into the purchaser and non-purchaser groups.

For this application, MailPurchase has decided to use SPSS Clementine (in particular, the logistic regression node) to construct the prediction model. The results are summarised in Figures 3.4 and 3.5. The results in Figure 3.4 (see lower right table) indicate that the logistic regression model is statistically significant with a p-value of 0.000 (rounded to three decimal places). That is, not all the model coefficients are equal to zero and hence, there is at least one variable that contributes significantly to the prediction of purchase and non-purchase status.

In addition, the equation in Figure 3.4 (see upper right panel) shows that the logistic regression model constructed based on the database is:

$$\text{Predicted Probability of Purchase} = 1/[1 + e^{-(\text{ Theoretical Index})}]$$

where the theoretical index is:

$\hat{y}$ = 1.426 – 0.02332*Age + 0.0009866*Expend + … – 0.2849*Race(=Malay)

Here, dummy variables are used to represent non-metric variables such as marital status and race.

The above logistic regression model can be used to predict the probability of purchase and non-purchase. This predicted probability when compared to the cut-off probability (default = 0.5) can be used to classify customers as purchasers or non-purchasers. However, before the model is used, it is prudent to further assess its adequacy and accuracy.

Figure 3.5 (upper right table) shows the individual test results. As shown, age, gender, marital status, membership and race are statistically significant with p-values lower than α of 0.05. That is, these variables are significantly associated with the purchase/non-purchase of promoted products by the customers.

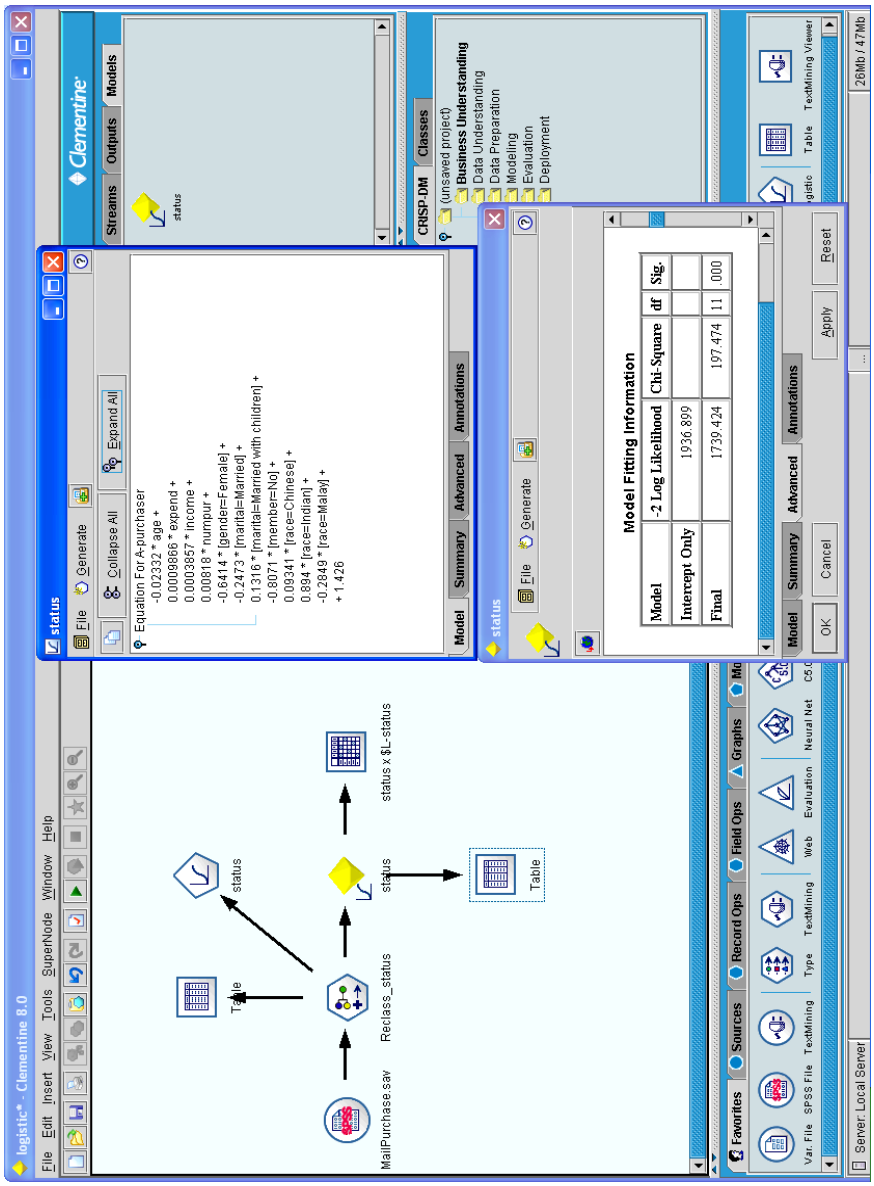


Figure 3.4 Logistic Regression Model and Model Test Result

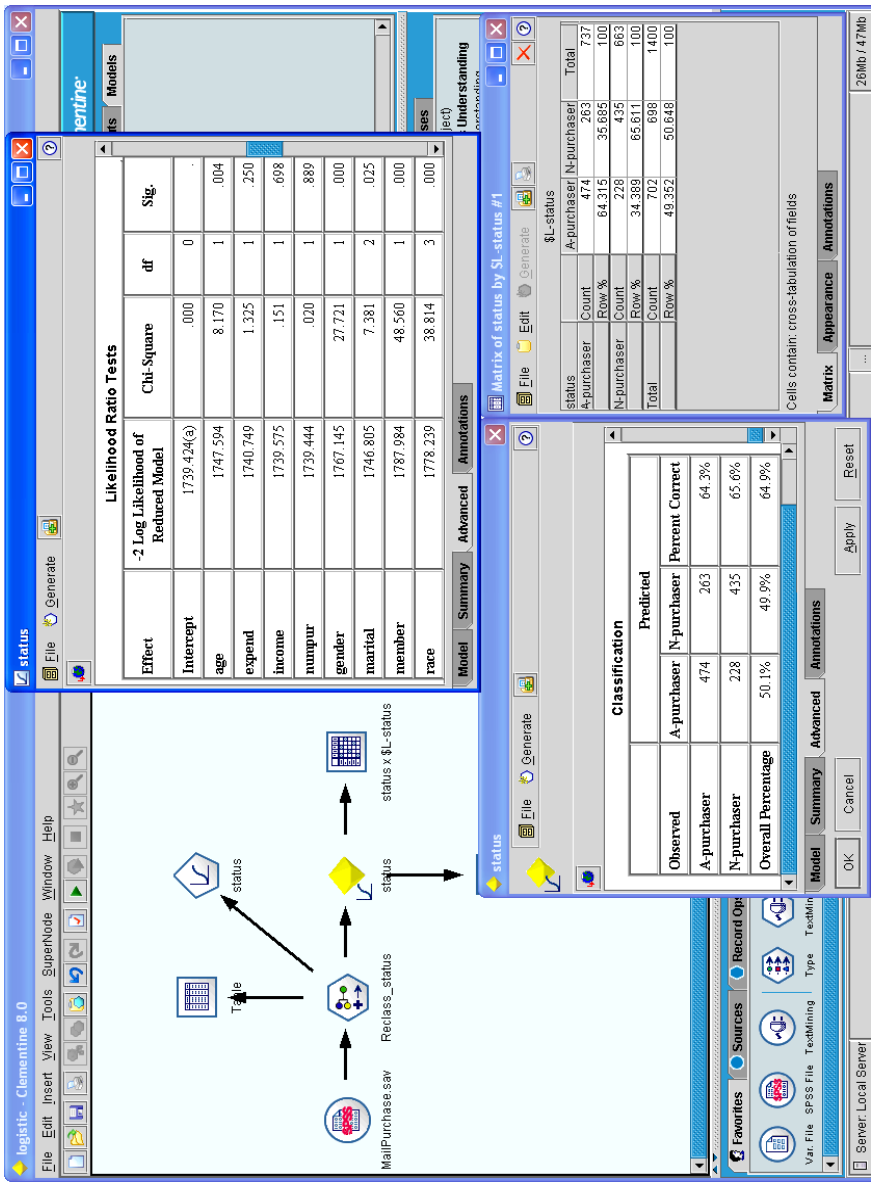


Figure 3.5 Individual Variable Test Results and Accuracy Rates

On the other hand, purchasing patterns such as the average monthly expenditure on MailPurchase's products and the average number of quarterly purchases are not significantly associated with the purchase of promoted products during the quarterly marketing campaigns. Also, annual income does not appear to affect the purchase or non-purchase of promoted items. These findings can help MailPurchase understand its customers better.

In particular, interpretation of the coefficients of the significant variables can help MailPurchase analyse the direction and magnitude of the impact of these variables on the purchase/non-purchase of promoted products.

As shown in Figure 3.5 (bottom half), the accuracy rate for purchase is 64.3% and the accuracy rate for non-purchase is 65.6%. The overall accuracy rate is 64.9%. Assuming that these rates are acceptable to MailPurchase, the logistic regression model can be used for targeting the most likely customers for the next marketing campaign. The next chapter will discuss accuracy rates in greater detail and also the financial evaluation of model results.

3.3 Neural Networks

Neural networks are frequently referred to as universal approximators as they can often model complex relationships in data well. Hence, they are useful for recognising patterns in data and for predictive modelling. Complex relationships include non-linear relationships as well as interaction effects. With interaction effects, the effect of an input variable on the target variable depends on the level of another input variable. For example, age may favourably affect expenditure when income is high; however, age may not affect expenditure when income is low. Neural networks can be used to model both metric and non-metric target variables.

Neural networks are modelled after the human brain, which can be perceived as a highly connected network of neurons (called nodes in neural networks terminology). One way to understand the concepts underlying neural networks is to think about how a child learns to play basketball (Dhar and Stein, 1997). In such a learning process, there is a fair amount of trial-and-error (e.g., throwing the basketball at different locations using different amounts of strength), adjustments (e.g., by varying the direction and strength of a throw) and generalisation (e.g., by "knowing" how to throw a basketball from new locations and under different conditions). In a similar way, trial-and-error, adjustments and generalisation are involved in constructing neural network models.

In addition, the architecture of a neural network mirrors that of the nervous system. As explained by Dhar and Stein (1997), the nervous system consists of a network of nerve cells or neurons. These neurons receive different pieces of information (i.e., stimuli) and process them. The information then travels through the nervous system (i.e., network) via neurotransmitters (i.e., connections). Neuron connections can be strengthened or weakened over time and with experience. Through this learning process, new responses to stimuli are developed, old ones are modified and unused ones are removed.

The concepts mentioned above can be translated into a neural network model in the following way. Suppose that a target variable is to be predicted on the basis of five input variables. Then, the target variable can be represented by an output node in the output layer of a neural network and the five input variables by five input nodes in the input layer of the neural network. This representation is shown in Figure 3.6, where O_1 is the output node and I_1 to I_5 the input nodes. Compared to the nervous system, the nodes are the neurons and the input variables the stimuli. The stimuli are transmitted between neurons via connections.

Two important operations take place in a neural network. Firstly, the five input nodes are aggregated by using weights. Let weight w_{ij} be the weight

connecting node i to node j, where node i and node j are nodes in two different layers. With reference to Figure 3.6, w_{11} refers to the weight connecting node I_1 to node O_1. Similarly, w_{51} refers to the weight connecting node I_5 to node O_1. Aggregation is done by multiplying each input by its weight and summing them up. Mathematically, the aggregation of the input nodes I_i can be expressed as $\sum I_i w_{ij}$. Frequently, a bias term (say, I_0) is also included in the aggregation. This is similar to having an intercept term in regression. Denoting the aggregated sum as A_j (i.e., an input value feeding into node j), then:

$$A_j = I_0 + \sum I_i w_{ij} \text{ for } i = 1 \text{ to } i$$

Here, the weights correspond to connections in the nervous system.

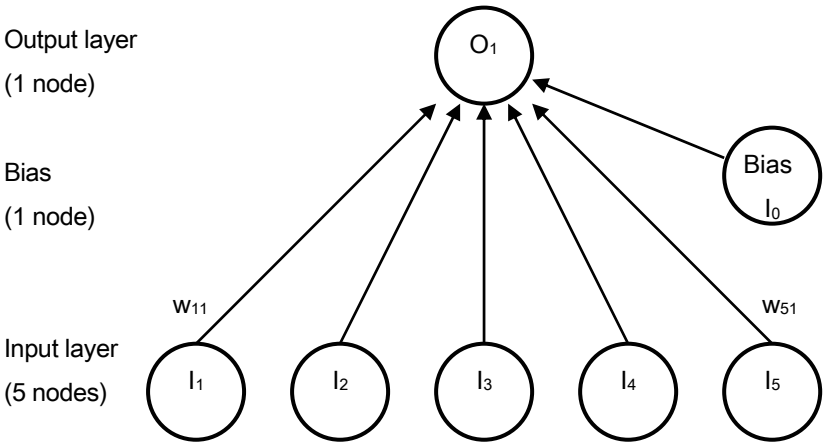


Figure 3.6 Neural Network with One Input Layer and One Output Layer

Secondly, when an aggregated input value is fed into a node, it is transformed to a new value via a transfer function. For Figure 3.6, A_j (the aggregation of the input nodes plus the bias) is fed into the output node O_1. Let the transformed output value be denoted by $\hat{y}$, the predicted output. If the transfer function is linear, the simplest of which is $\hat{y} = A_j$ (i.e., the linear

transformation has a coefficient of 1), then ignoring the subscript j and assuming p input variables I_1 to I_p:

$$\hat{y} = A_j = I_0 + \sum I_i w_{ij}$$

$$\Rightarrow \hat{y} = I_0 + I_1 w_1 + I_2 w_2 + \ldots + I_p w_p$$

This is equivalent to multiple regression where I_0 is the intercept and w_1 to w_p the regression coefficients. In this sense, multiple regression can be considered a special (and very simple) case of neural network models.

If the transfer function is a logistic function, then:

$$\hat{y} = 1/[1 + e^{-A_j}]$$

$$\Rightarrow \hat{y} = 1/[1 + e^{-(I_0 + I_1 w_1 + I_2 w_2 + \ldots + I_p w_p)}]$$

which is equivalent to the logistic regression model. Hence, the logistic regression model can also be considered a special case of neural network models.

The neural network architecture shown in Figure 3.6 is a very simple one. In most neural networks, there is at least one hidden layer of hidden nodes (H_i). Figure 3.7 incorporates one hidden layer with two hidden nodes H_1 and H_2.

The extension of the workings of the neural network in Figure 3.6 to that in Figure 3.7 is relatively straightforward. The first step is to aggregate the input layer (and bias) to generate an input value to be fed into the hidden layer nodes H_1 and H_2. Let these aggregations be denoted by A_1 and A_2, respectively. Then, the second step is to transform A_1 in H_1 and A_2 in H_2 via some transfer function. Let the transformed results be B_1 and B_2, respectively. These represent values going out from the hidden nodes. The third step is to aggregate B_1 and B_2 and the bias ($Bias_3$) to generate an input value to be fed into the output node O_1. Finally, at the fourth step, this value is transformed through a transfer function to generate the predicted value $\hat{y}$.

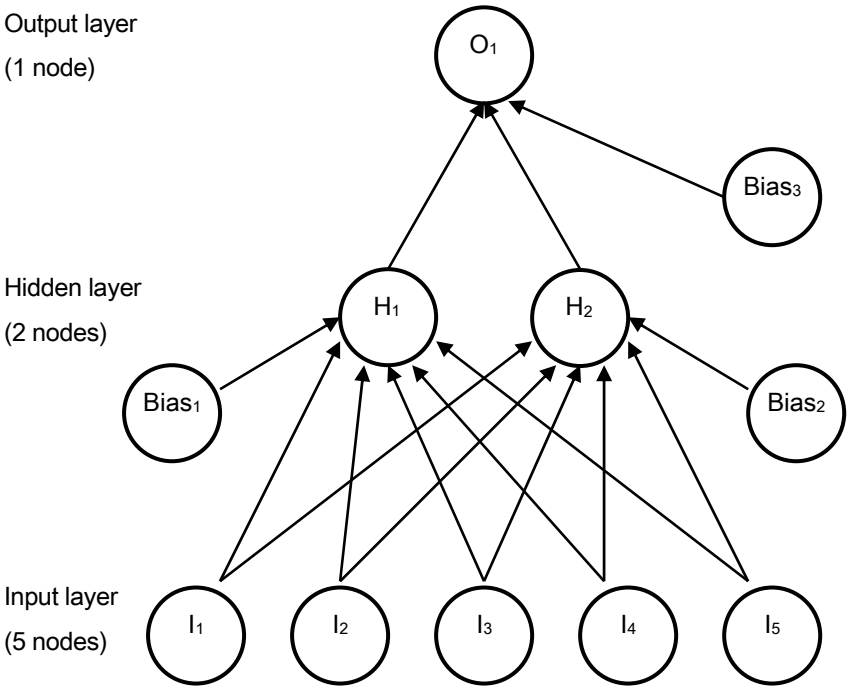


Figure 3.7 Neural Network with One Hidden Layer

Since there is a weight associated with every connection of a pair of nodes in a neural network, there will be a lot more estimates in neural networks (i.e., weights) than in regression models (i.e., coefficients). Further, it is possible to have several hidden layers and several hidden nodes for each layer. However, rules-of-thumb indicate that neural networks can predict well with only one or two hidden layers and with only five or fewer hidden nodes per layer. An excessively large neural network tends to over-fit the data (by "memorising" unique patterns in the data). Hence, it predicts poorly on data outside of the data that are used to construct the model (also called the training data in neural network terminology). In other words, an over-fitted model lacks generalisability.

While neural networks are universal approximators and hence very good prediction models, they are often criticised as "black-box" models. That is, they do not indicate how input variables affect the target variable. This shortcoming is not surprising given the aggregation and transformation of values throughout the entire network. One way to mitigate this shortcoming is to perform sensitivity analysis. This usually means varying a particular input variable from its mean and observing how the predicted value changes. This will give some clue as to the effect of changes in an input variable on the target variable. Another way to mitigate the "black-box" problem is to model the predicted values of the neural network using the input variables. This will give some indication as to how the input variables are associated with the neural network predictions.

There are a few algorithms that neural networks can use for learning (i.e., to estimate the weights). The most popular algorithm is the back-propagation algorithm. The general principles are as follows. All the weights are initially set to some small random values. Then an observation (known as an example in neural network terminology) is presented to the neural network for processing and prediction. The predicted target value is compared to the actual target value and the error (i.e., actual – predicted) is computed. Next, this error is fed back (or back-propagated) to the network and weights are then adjusted so as to minimise the error. This process of learning and adjustment continues with every observation presented to the neural network. There can be several cycles of all the observations being presented to the neural network. The training process stops when a pre-determined number of cycles has been completed or when the weights do not change significantly.

The adjustment of weights can be summarised by the following formula:

$$w_{ij,(t+1)} = w_{ij,(t)} + \lambda(\varepsilon w_{ij})(I_i) + \alpha(w_{ij,(t)} - w_{ij,(t-1)})$$

where t is the number of times the neural network is updated. Here, λ is the learning parameter or learning rate that determines the speed of learning. That is, a larger λ imposes a bigger adjustment to the weights, and vice versa.

Further, α is the momentum. It reduces the current adjustment when previous adjustments are getting smaller. This fine-tunes the weight adjustments to move towards minimum prediction error. The weight adjustment also depends on the input value of the node (i.e., I_i) and the sensitivity of the output value of the node to a change in the weight (i.e., εw_{ij}).

As noted earlier, neural networks can be used for predicting metric and non-metric target variables. That is, they can be used for either estimation or classification.

3.3.1 Kohonen Networks

The discussion so far assumes a predictive modelling context where a target variable is to be predicted on the basis of input variables. This form of neural networks is called supervised training. The neural networks are supervised in the sense that the actual target values serve as a "teacher" to guide learning (via the weights). Neural networks can also be unsupervised – that is, there is no target variable to guide learning. In unsupervised neural networks, all variables are of the same status and are not differentiated as target or input variables.

Kohonen networks (also called Kohonen nets and self-organising maps or SOM) are unsupervised neural networks that perform the function of clustering. (Recall that Kohonen networks are discussed briefly under the topic of clustering in Chapter 2). They cluster or segment data on the basis of patterns of the input variables so that similar patterns (i.e., observations) are grouped together.

A Kohonen network is a neural network that is arranged as an n-dimensional grid or array of nodes. Usually, only two dimensions, or at most three dimensions, are used. Thus, a 2x3 Kohonen network has six nodes (or clusters). Each node is connected to all the input variables. Each node is also connected to other (so-called neighbourhood) nodes. This architecture is very

different from the multi-layer architecture discussed in the previous section. A graphical representation is shown in Figure 3.8.

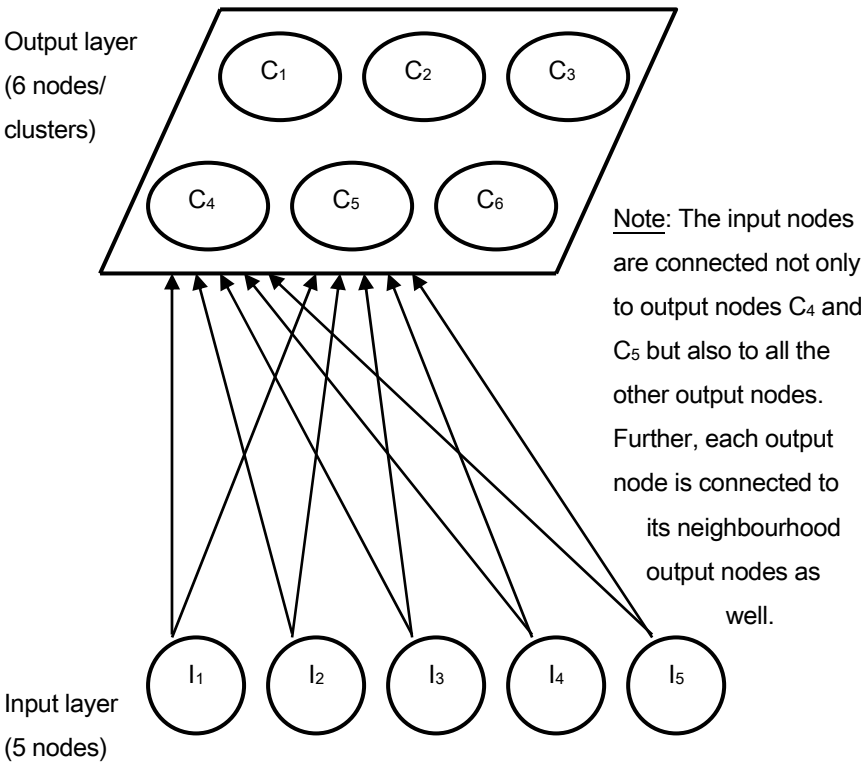


Figure 3.8 Kohonen Network with Five Input Nodes and Six Output Nodes

The learning algorithm proceeds as follows. All the weights are initially set to some small random values. When an observation (or example) is presented to the Kohonen network, its pattern of input variables is compared to the weight pattern of the nodes in the grid. The node with weights that are most similar (or the least dissimilar) to the values of the input variables "wins" the observation. Here, dissimilarity is measured by the distance d_j as defined below:

$$d_j = \sqrt{\sum}(x_i - w_{ij})^2$$

where x_i denotes the input values and w_{ij} the weights for connections from the input variables to node j. The most similar node has the smallest distance or dissimilarity.

Next, the weights of this winning node are adjusted to make them more similar to the pattern of input variables. In addition, the weights of surrounding (neighbourhood) nodes are also likewise adjusted, but to a lesser extent than the weight adjustments for the winning node. Nodes that are defined as neighbourhood nodes of the winning node are pre-specified (e.g., neighbourhood nodes can be defined as nodes that are at most two nodes away from the winning node). The adjustment formula can be summarised as follows:

$$w_{ij,(t+1)} = w_{ij,(t)} + c(x_i - w_{ij,(t)})$$

where c captures the learning rate as well as the specification of neighbourhood nodes. Further elaboration of this can be found in Smith and Ng (2003).

This process of learning and adjustment continues with every observation and there can be several cycles of all observations being presented to the Kohonen network. The process stops when a pre-determined number of cycles has been completed or when the weights do not change significantly. The end result is a map of nodes or clusters, where each cluster contains similar observations and clusters with similar patterns (or profiles) are closer together on the map. Also, clusters with dissimilar profiles are further away from each other. As mentioned in the previous chapter, this is a desirable property in clustering applications.

3.3.2 Illustration of Neural Network Model

To illustrate neural networks, suppose that MailPurchase wants to re-perform the illustration in section 3.2.3 using a neural network model instead of logistic regression. Recall that MailPurchase is interested to classify existing customers

as likely purchasers or non-purchasers. For this prediction model, purchase status is the target variable and the input variables comprise purchasing patterns (i.e., expend and numpur) and demographic characteristics (i.e., age, gender, income, race, marital and member). For this application, MailPurchase has decided to use SPSS Clementine (in particular, the neural network node) to construct the prediction model. The results are summarised in Figures 3.9 and 3.10.

As shown in Figure 3.9 (lower right table), the accuracy rates for purchasers and non-purchasers are 66.62% and 64.56%, respectively. These give an overall accuracy rate of 65.64% for the neural network model (see upper right table in Figure 3.9). These accuracy rates are similar to those of the logistic regression model. (The comparison of results across models will be discussed in the next chapter). From the relative importance of the input variables, the top three most important input variables associated with purchase status are race, marital status and age. It is noted that no statistical tests are associated with these results. Instead, these results are derived from sensitivity analysis (see earlier discussion on neural networks). The weights of the neural network model are given in Figure 3.10. The prediction (i.e., classification) of the model is shown under the column "$N-status" and the confidence of the prediction is given in the column "$NC-status". No probability estimates are generated by the neural network model.

As in the case of the logistic regression model, assuming that the accuracy rates are acceptable to MailPurchase, the neural network model can be used for targeting the most likely customers for the next marketing campaign. These are customers who are predicted as "A-purchaser" with a high "$NC-status" score.

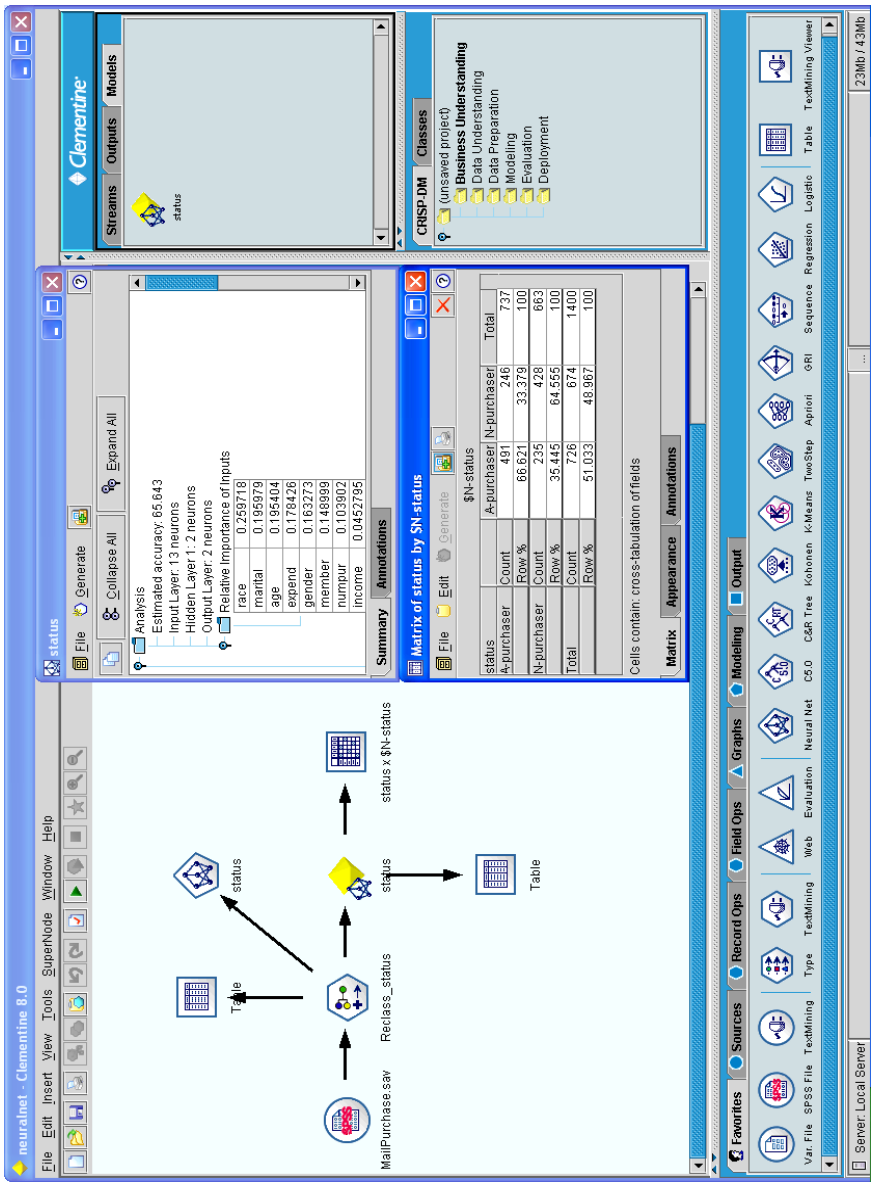
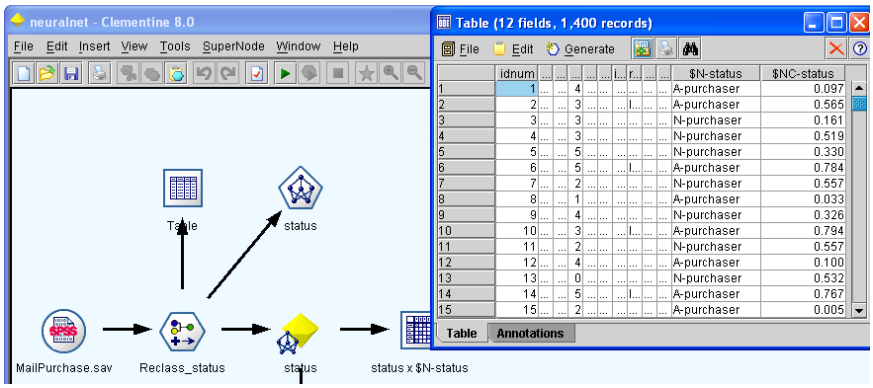


Figure 3.9 Neural Network Results




```
<Neuron id="13" bias="-3.1362004423">
<Con from="0" weight="-1.77514997601"/>
<Con from="1" weight="-1.15693985345"/>
<Con from="2" weight="3.38720931463"/>
<Con from="3" weight="-1.81908488353"/>
<Con from="4" weight="1.23603810593"/>
<Con from="5" weight="-0.0436623106591"/>
<Con from="6" weight="-4.07963675578"/>
<Con from="7" weight="0.54602806527"/>
<Con from="8" weight="1.03582441993"/>
<Con from="9" weight="-2.24659341588"/>
<Con from="10" weight="3.34532475509"/>
<Con from="11" weight="-1.72788821414"/>
<Con from="12" weight="-1.6831201078"/>

<Neuron id="14" bias="-2.20961300422">
<Con from="0" weight="-1.09137607335"/>
<Con from="1" weight="0.177069335488"/>
<Con from="2" weight="5.87462903845"/>
<Con from="3" weight="-0.20523378221"/>
<Con from="4" weight="-2.3223812605"/>
<Con from="5" weight="0.531199561613"/>
<Con from="6" weight="-1.4038129393"/>
<Con from="7" weight="7.15735688805"/>
<Con from="8" weight="0.578548091229"/>
<Con from="9" weight="1.60371047408"/>
<Con from="10" weight="-2.28636766866"/>
<Con from="11" weight="-1.9796015086"/>
<Con from="12" weight="-1.1998552929"/>

<Neuron id="15" bias="1.25816246348">
<Con from="13" weight="-2.19327319962"/>
```

Figure 3.10 Neural Network Weights and Predictions

3.4 Decision Trees

Besides regression and neural networks, decision trees can also be used for predictive modelling. Generally, regression can be considered a statistical method, neural networks an artificial intelligence model and decision trees a machine learning technique.

Decision trees either estimate a metric target variable or classify observations into one category of a non-metric target variable by repeatedly dividing observations into mutually exclusive and exhaustive subsets. Hence, the algorithm used to construct decision trees is also referred to as recursive partitioning algorithm.

In a decision tree, each observation is eventually assigned to a node (also called leaf) that has a predicted value or classification. The end product can be graphically represented by a tree-like structure (called a decision tree), which is a compact explanation/representation of the data. The end product can also be represented by explicit decision rules (similar to the association rules discussed in the previous chapter). The resulting visual representation and explicit rules make decision trees easy to interpret and use. In addition, decision trees can model complex non-linear and interaction relationships reasonably well.

Many algorithms are available to construct decision trees. The more common ones are CHAID (chi-square automatic interaction detection), C5.0 (a proprietary algorithm) and CART (classification and regression tree). Some algorithms are used for metric target variables only, some for non-metric target variables only and still some for both. Decision tree algorithms are very intensive (i.e., a lot of computations are performed to construct the tree). To better understand the decision tree methodology, the chi-square test of independence is first discussed.

3.4.1 Chi-square Test of Independence

Suppose that there are two non-metric variables: risk of fraud (defined as high-risk and low-risk) and location of transaction (defined as local and overseas). Suppose further that a credit card company is interested to investigate if the risk of fraud is dependent on the location of transaction. To do this, a matched sample of 1000 high-risk and low-risk transactions is reviewed. Assume that a matched sample is used here because high-risk transactions are rare and hence, a random sample will not yield a sufficient number of high-risk transactions for investigation. Therefore, a sample of 500 high-risk transactions is first identified/selected and then matched to a sample of 500 low-risk transactions based on transactional characteristics, except for location of transaction. A cross-tabulation of the sample data is given in Table 3.3.

Table 3.3 Cross-tabulation of Risk of Fraud and Location of Transaction

Location of Transaction	**Risk of Fraud**		**Total**
	High-risk	Low-risk	
Local	100 (20%)	300 (60%)	400
Overseas	400 (80%)	200 (40%)	600
Total	500 (100%)	500 (100%)	1000

The investigation of the relationship between risk of fraud and location of transaction can be expressed by the null hypothesis "H_0: Risk of fraud and Location of transaction are independent". A null hypothesis can be viewed as a statement for statistical testing. If the null hypothesis H_0 is rejected, then it can be concluded that risk of fraud and location of transaction are dependent. That is, the relative incidence of high-risk and low-risk transactions depends on whether the transactions are made locally or overseas.

Marginal (or prior) probabilities can be computed for the data in Table 3.3. These are shown in Table 3.4. As can be seen, if a transaction is randomly chosen from the sample of 1000 transactions, then the probability of it being a high-risk transaction is 0.50 and the probability of it being a local transaction is 0.40. To facilitate discussion, risk of fraud is abbreviated as "fraud" and location of transaction as "location". If fraud and location are independent of each other (i.e., if the null hypothesis H_0 is true), then probability theory requires that the joint probability of a particular category of fraud and a particular category of location is:

Prob(Fraud and Location) = Prob(Fraud)*Prob(Location)

For example,

Prob(High-risk and Local) = Prob(High-risk)*Prob(Local)

= 0.50*0.40 = 0.20

Table 3.4 Marginal Probabilities

Location of Transaction	**Risk of Fraud**		**Total**	**Probability**
	High-risk	Low-risk		
Local	100	300	400	0.40
Overseas	400	200	600	0.60
Total	500	500	1000	
Probability	0.50	0.50		1.00

Following from above, if fraud and location are independent, then out of the 1000 transactions in the sample, 200 (i.e., 0.20*1000) transactions are expected to fall into that particular (i.e., high-risk and local) cell in the cross-tabulation. Similar computations can be made for all the other cells. These are called expected frequencies as they represent expected outcomes under the assumption that H_0 is true. Expected frequencies are different from the

numbers shown in Table 3.3, which are referred to as actual frequencies (i.e., the outcomes actually observed). The expected frequencies (with joint probabilities within brackets) are shown in Table 3.5.

Table 3.5 Cross-tabulation of Expected Frequencies

Location of Transaction	**Risk of Fraud**		**Total**
	High-risk	Low-risk	
Local	200 (0.20)	200 (0.20)	400
Overseas	300 (0.30)	300 (0.30)	600
Total	500	500	1000

Whether fraud and location are dependent or independent can be assessed by comparing the actual frequencies in Table 3.3 with the expected frequencies in Table 3.5. In particular, Table 3.5 shows the outcomes expected if fraud and location are indeed independent (i.e., if H_0 is true) while Table 3.3 shows the outcomes actually observed. Hence, if fraud and location are in fact independent, then the actual and expected frequencies should be quite similar. Conversely, if fraud and location are in fact dependent, then the actual and expected frequencies should be rather dissimilar. One way to measure the degree of similarity/dissimilarity in this context is the chi-square test statistic:

$\chi^2 = \sum[(\text{actual frequency} - \text{expected frequency})^2/(\text{expected frequency})]$

summed over all the cells in the cross-tabulation.

A large χ^2 suggests that fraud and location are probably dependent and hence, H_0 can be rejected. As with other statistic tests discussed earlier, the chi-square test statistic has a p-value too. The test statistic χ^2 and its associated p-value are inversely related (i.e., a large test statistic is associated with a small p-value, and vice versa). Roughly speaking, for the current example, the p-value can be viewed as the probability of observing the actual

frequencies in Table 3.3 if H_0 is true. Therefore, if the p-value is small, then it is likely that H_0 is false (since the probability of observing the actual frequencies if H_0 is true is rather small). Hence, H_0 can be rejected. This means that a smaller p-value indicates a stronger likelihood that fraud and location are dependent. However, by rejecting H_0, the probability of an incorrect rejection (i.e., H_0 is actually true and yet rejected) is equal to the p-value.

Whether H_0 should be rejected depends on the risk of incorrect rejection that one is willing to bear. This is known as the level of statistical significance or α. Traditionally, the required level of statistical significance (α) is usually set at 0.01, 0.05 or 0.10. A higher α signifies a willingness to bear a greater risk of an incorrect rejection of H_0 (i.e., concluding dependence between two non-metric variables when there is actually independence). Also, H_0 can be rejected as long as the p-value is less than or equal to α. In this case, the probability of an incorrect rejection of H_0 (i.e., p-value) is smaller than the risk of an incorrect rejection that one is willing to bear (i.e., α).

In the current example, if it is concluded that fraud and location are dependent, then it means that the relative incidence of fraud depends on whether the location of the transaction is local or overseas. Here, the chi-square test statistic is 166.67, with a p-value of 0.0001. Hence, it can be concluded that fraud and location are dependent at α of 0.05. In particular, looking at Tables 3.3 and 3.5, it can be seen that a relatively higher incidence of high-risk transactions occurs among overseas transactions. Similarly, a higher incidence of low-risk transactions occurs among local transactions. Viewed differently, the chi-square test result suggests that location significantly differentiate between the relative occurrence and non-occurrence of fraudulent transactions. (More detail on the chi-square test of independence can be found in most basic statistics textbooks).

The above discussion facilitates the understanding of how decision trees are constructed. In particular, from the decision tree perspective, the chi-square test of independence helps to explain how the most significant variable

to split a decision tree into branches is determined and how the splitting points (or thresholds) are identified. Generally, the more statistically significant a variable is, the more important it is for constructing the decision tree and the more accurate it is for prediction. This understanding can then be extended to cover other decision tree algorithms.

3.4.2 Chi-square Automatic Interaction Detection (CHAID)

The chi-square automatic interaction detection (CHAID) algorithm is a commonly used algorithm to construct decision trees for non-metric target variables. The following example explains the working of the algorithm.

Suppose that the objective of a data mining application is to predict the buying status (i.e., buyer versus non-buyer) of a particular product on the basis of the demographic variables gender (categorised as male and female), race (categorised as Chinese, Malay and Indian), age and income. For these input variables, gender and race are non-metric while age and income are metric. Assume that the sample comprises 600 buyers and 900 non-buyers.

At the first step of constructing a decision tree using CHAID, each input variable is evaluated on its potential to split the data into two or more subsets so that the target variable is as differentiated (in a statistical sense) between the subsets as possible. Since the target variable has only two categories (buyer and non-buyer) and gender also has only two categories (male and female), a 2-way split can be made as shown in Figure 3.11.

From Figure 3.11, it can be seen that Nodes 1 and 2 together form a 2x2 contingency table defined by the variables buying status and gender. Hence, a chi-square test of independence can be performed to assess the statistical significance of gender in differentiating between buyers and non-buyers. In particular, if the null hypothesis (that buying status and gender are independent) is rejected, then it means that the incidence of buyers/non-buyers among males and females are significantly different. In other words,

gender can significantly differentiate between buyers and non-buyers (e.g., there are proportionately more female buyers than male buyers). In this example, the chi-square test statistic is 1.55 with a p-value of 0.213, which is not statistically significant for the traditional levels of α. Thus, buying status does not depend on gender. That is, gender does not help differentiate between buyers and non-buyers.

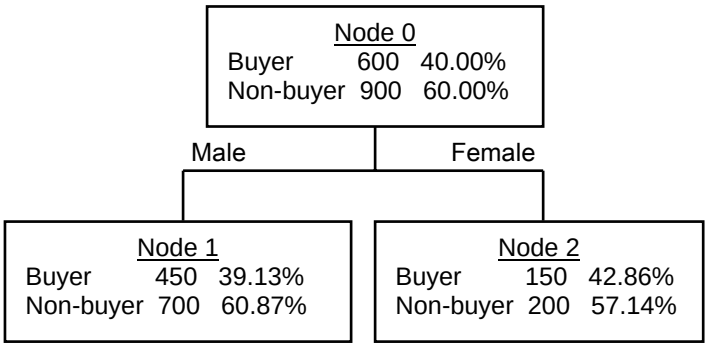


Figure 3.11 Decision Tree – Split by Gender

Similarly, the significance of race in differentiating between buyers and non-buyers can also be evaluated. Since there are three categories of race, one possible split is shown in Figure 3.12. As in the case of gender, a chi-square test of independence can be performed on race for Nodes 1, 2 and 3 (a 2x3 contingency table). The test assesses the statistical significance of race in differentiating between buyers and non-buyers. Here, the chi-square test statistic is 4.66, with a p-value of 0.097. Hence, at a significance level of 0.10, race is a significant variable in differentiating between buyers and non-buyers. From Figure 3.12, it appears that Chinese consumers are less likely to buy the product as compared to their Malay and Indian counterparts.

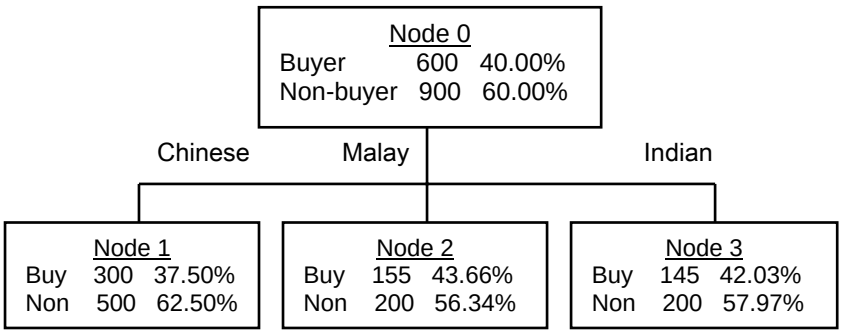


Figure 3.12 Decision Tree – Split by Race

It can also be noted from Figure 3.12 that Malay and Indian consumers have a very similar purchasing pattern. Hence, it may be possible to combine the two races into one single subset. This combination is shown in Figure 3.13. The chi-square test statistic is now 4.46, with a p-value of 0.035. Therefore, having Chinese as a node and Malay and Indian as another node differentiates between buyers and non-buyers better (in a statistical sense) than having the three races as separate nodes do. Generally, for a non-metric input variable with more than two categories, a decision tree algorithm would try different combinations of the categories to search for the split(s) that best differentiate (i.e., predict) the target variable. In CHAID, how well the target variable is differentiated or predicted is assessed by the chi-square test of independence. That is, a statistical criterion is used.

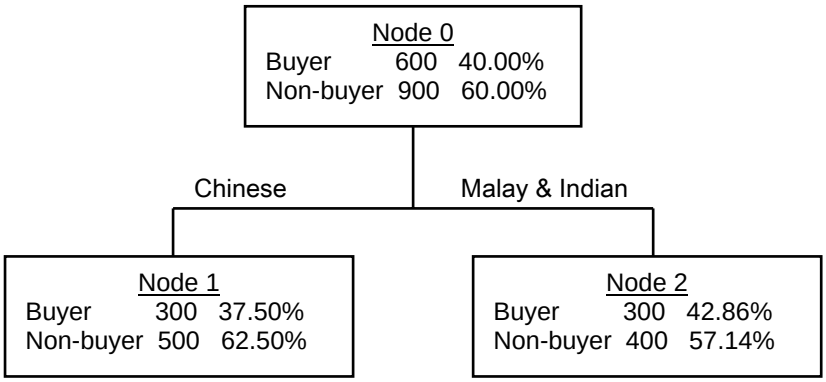


Figure 3.13 Decision Tree – Split by Race (Binary Split)

Determining the best split for metric input variables is more difficult. To illustrate, assume that a metric input variable has the following five values in ascending order: A < B < C < D < E. Then, four possible splitting points (or thresholds) are the average of A and B, the average of B and C, the average of C and D, and the average of D and E. This means that for the purpose of splitting, the metric input variable can be considered as a variable with five categories, with their boundaries defined by the four averages mentioned above. Hence, the chi-square test of independence can be applied to the five categories separately or in any combination. However, given that values in a metric input variable are ordered, combinations can only be made for adjacent values (e.g., A, B and C can be combined into one category but not A, C and E). This is not the case for non-metric input variables where all combinations of the categories are possible and interpretable (e.g., Chinese with Malay, Chinese with Indian, and Malay with Indian). As metric input variables can take on many values (much more than the five values illustrated here), decision tree algorithms perform very intensive computations to evaluate metric input variables.

In the current example, suppose that the best splits for age and income are those shown in Figures 3.14 and 3.15. Then, the corresponding chi-square test statistics are 5.52 (p-value = 0.019) and 10.77 (p-value = 0.001), respectively. These results are statistically significant (i.e., age and income do differentiate between the buyers and non-buyers). Hence, age and income can contribute towards predicting buying status. As the figures show, increasing age and increasing income are associated with a higher probability of buying the product.

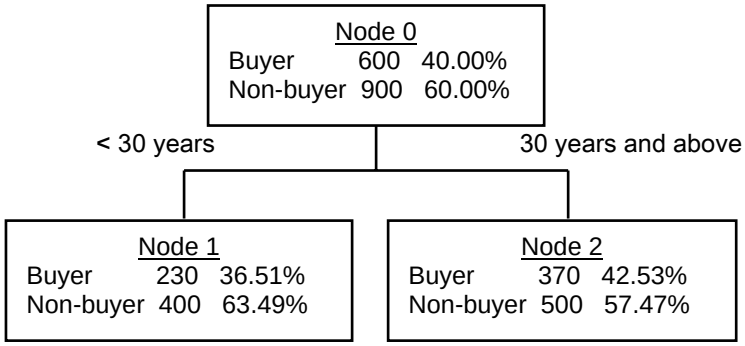


Figure 3.14 Decision Tree – Split by Age

At the second step of the decision tree algorithm, the statistical results (i.e., chi-square test statistics or p-values) of all the input variables are compared and the most significant input variable is selected to split the tree at the best threshold(s) identified for that variable. The most significant variable can be deemed to be the variable that best differentiate between the categories in the (non-metric) target variable, and hence the input variable that can predict the target variable best.

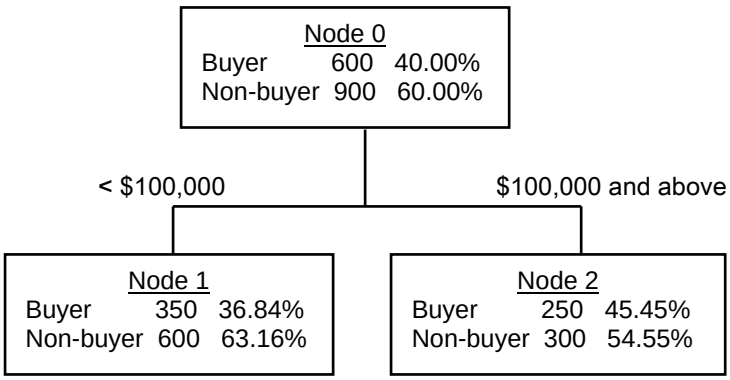


Figure 3.15 Decision Tree – Split by Income

For the input variables gender, race, age and income, the most statistically significant variable is income (with the smallest p-value of 0.001). When the decision tree is split by income as per Figure 3.15, there are two child nodes: Node 1 and Node 2. Node 0 is called the parent node. Child nodes can themselves be parent nodes as the decision tree grows to more levels (i.e., greater depth).

At the third step, the first and second steps are repeated for each child node in the decision tree. With reference to Figure 3.15, there are two child nodes: (1) income < $100,000, and (2) income of $100,000 or more. Hence, each input variable is again evaluated on its potential to split the data in each child node into two or more subsets so that the target variable (in each child node) is as differentiated (in a statistical sense) between the subsets as possible. At this level (or depth) of the decision tree the earlier child nodes are now parent nodes. Also, although Node 1 and Node 2 are the result of splitting Node 0 by income, the input variable income can still be used to further split the nodes into finer income ranges (e.g., income of $100,000 to less than $150,000 and income of $150,000 and above).

Using the same input variable again for splitting may not be possible in certain instances. For example, when a parent node is split by gender into male and female consumers, the node of male consumers cannot be further split by gender as the node comprises only male consumers.

Decision tree algorithms perform steps one to three repeatedly–hence the name recursive partitioning algorithm. The iterative process stops when a stopping rule in encountered. For example, tree growing may stop if: (1) the decision tree has grown to a pre-specified maximum depth or number of levels, (2) all the potential parent nodes fail to have a pre-specified minimum number of observations, (3) all the potential child nodes fail to have a pre-specified minimum number of observations, or (4) none of the input variables can reach a pre-specified level of statistical significance.

Sometimes to keep the decision tree parsimonious (i.e., small and yet accurate), tree pruning is performed. That is, the contribution of each split is evaluated by comparing the accuracy of the decision tree with and without the split. Splits that do not contribute significantly to the accuracy of the decision tree are then removed. In tree pruning, what constitutes a significant contribution has to be defined (e.g., a required minimum improvement in accuracy rates). Sometimes the number of splits at each node is limited to keep the tree parsimonious (i.e., small yet accurate). Stopping rules and tree pruning prevent the over-fitting of data, which produces decision trees that perform poorly on new data. However, there is always a trade-off between a more parsimonious (and therefore more interpretable) decision tree and a more accurate (and therefore bigger) decision tree.

To conclude the current example, assume that the final decision tree constructed is the one shown in Figure 3.16, where B represents buyer and N non-buyer. Also, the predicted classification in the terminal nodes is indicated in

bold type. Terminal nodes are final nodes that do not split any further. Their paths define the decision rules for classifying observations.

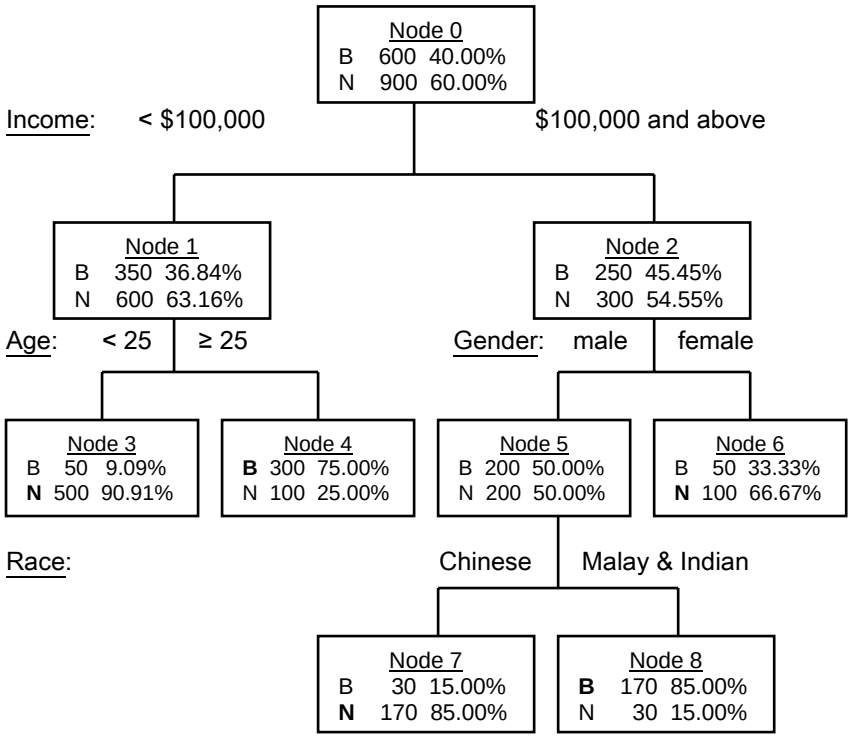


Figure 3.16 Final Decision Tree

The following comments can be made about the decision tree in Figure 3.16. Firstly, nodes 3, 4, 6, 7 and 8 are terminal nodes, which predict the classification of observations based on the mode (i.e., most commonly occurring category in the target variable) in each node. That is:

1) Node 3: a customer with income of less than $100,000 and age of less than 25 years is predicted as a non-buyer (with a confidence of 90.91%);
2) Node 4: a customer with income of less than $100,000 and age of at least 25 years is predicted as a buyer (with a confidence of 75.00%);
3) Node 6: a customer with income of at least $100,000 and who is female is predicted as a non-buyer (with a confidence of 66.67%);
4) Node 7: a customer with income of at least $100,000 and who is male and Chinese is predicted as a non-buyer (with a confidence of 85.00%); and
5) Node 8: a customer with income of at least $100,000 and who is male and Malay or Indian is predicted as a buyer (with a confidence of 85.00%).

Input variables that appear higher up in the decision tree can be deemed as more important variables in predicting the target variable. Hence, from Figure 3.16, income is the most important variable, followed by age and gender. The next most important input variable is race. It is noted that the contribution of age, gender and race to the prediction of buying status apply only to certain segments of the sample. For example, age is an important variable only for those with income of less than $100,000. Such segmentation allows the modelling of interaction relationships. Decision trees can model non-linear relationships too (e.g., when nodes split into more than two subgroups showing non-linear associations between the input and target variables). As can be observed, a decision tree is very easy to interpret as the results can be visually represented as a tree-like structure. The results can be summarised into explicit decision rules too.

To assess the adequacy of the decision tree, its accuracy rates can be computed from the terminal nodes. The classification table for Figure 3.16 is shown in Table 3.6. As shown, the accuracy rates for buyers and non-buyers are 78.33% and 85.56%, respectively. The overall accuracy rate is 82.67%.

Table 3.6 Accuracy and Error Rates of Decision Tree Model

Actual Status	Predicted Status		Total
	Buyer	Non-buyer	
Buyer	470 (78.33%)	130 (21.67%)	600 (100%)
Non-buyer	130 (14.44%)	770 (85.56%)	900 (100%)
Total	600	900	1500
Overall accuracy rate = 1240/1500 = 82.67%			

3.4.3 Other Decision Tree Algorithms

In the discussion of CHAID, a statistical criterion (i.e., the chi-square test statistic or its associated p-value) is used to determine the input variables for constructing the decision tree as well as the thresholds for splitting the parent nodes so as to predict the target variable as accurately as possible. Instead of a statistical criterion, however, some decision tree algorithms use non-statistical criteria. Whatever the criterion used, the primary objective is still to make the decision tree as accurate as possible in predicting the target variable. The term accuracy is used in a very general sense here.

One way to think about using non-statistical criteria to construct a decision tree is to imagine the tree as a filtering process. In particular, nodes that are higher up in the decision tree do not differentiate between the categories in the target variable as well as nodes that are lower down in the decision tree (see, for example, Figure 3.16). In other words, filtering gets better as the decision tree grows in that the nodes become purer, more orderly or more informative (i.e., the nodes become better in predicting or indicating a particular category of the target variable). Hence, one way to determine the best input variable and threshold (for splitting) is to identify that variable that

leads to the best improvement in purity, orderliness or information content. This is also the variable that leads to the most reduction in impurity, disorderliness or lack of information.

In some decision tree algorithms, impurity, disorderliness or lack of information is measured by the concept of entropy. In particular, a higher level of entropy implies a lower level of purity, orderliness or information content. The formula to compute entropy (H) can be written as follows:

$$H = - \sum P_i \log_2(P_i)$$

where P_i is the probability of the i^{th} category of the target variable occurring in a particular node. For example, in Node 8 of Figure 3.16, the probability of buyers is 0.85 and that of non-buyers is 0.15. Therefore, for this node:

$$H = - [0.85\log_2(0.85) + 0.15\log_2(0.15)]$$

$$= - [0.85(- 0.235) + 0.15(- 2.735)] = 0.61$$

Entropy can be computed for each node in the decision tree. Also, entropy for a decision tree is the total of the entropy of all the terminal nodes (i.e., not the intermediate nodes) in the tree. The best input variable and threshold to grow a tree are that variable and threshold that give the greatest reduction in entropy. As a decision tree becomes purer, more orderly and more informative, its entropy approaches 0. The reduction in entropy is also sometimes referred to as information gain. The C5.0 algorithm is a proprietary algorithm (see the web site http://www.rulequest.com) that is based on concepts similar to information gain.

In CART (classification and regression tree), instead of the entropy measure, the Gini measure is used. The formula for the Gini measure (G) is:

$$G = \sum P_i P_j$$

where P_i and P_j ($i \neq j$) are the probabilities of the different categories of a target variable in a node of the decision tree. The tree growing process is similar to that described above using entropy. However, CART performs only binary splits (i.e., two-way splits into only two subgroups).

The chi-square test of independence in CHAID is appropriate only for non-metric target variables. For metric target variables, if a statistical criterion is desired, then the analysis of variance (ANOVA) is a good alternative. Like the chi-square test, ANOVA generates test statistics and p-values that can be used to determine the best input variables and thresholds to grow the decision tree. The idea of differentiation is still applicable. However, differentiation for metric target variables refers to nodes that best differentiate the mean of the metric target variable.

Suppose that there are two nodes comprising male and female consumers and the target variable is expenditure. In this case, for gender to contribute significantly to the prediction of expenditure, it is desirable for male and female consumers to have mean expenditures that are as different (i.e., differentiated) as possible. How well gender can differentiate between the mean expenditure of male and female consumers can be indicated by the ANOVA results. The tree construction process as per the earlier discussion of CHAID can then be used. In SPSS AnswerTree, the CHAID algorithm can apply either the Chi-square test of independence or ANOVA (analysis of variance) to handle non-metric and metric target variables, respectively.

When the target variable is metric, purity, orderliness and information content can also be measured by the variance of the target variable. The greater the variance, the lesser the extent of purity, orderliness and information content. Therefore, measures involving variance can also be computed to guide the construction of decision trees. Finally, it is noted that CART can predict both non-metric and metric target variables.

3.4.4 Illustration of Decision Tree Model

To illustrate decision trees, suppose that MailPurchase wants to re-perform the illustration in section 3.2.3 using a decision tree model instead of a logistic regression model. In this application, MailPurchase is interested to classify

existing customers as likely purchasers or non-purchasers (i.e., purchase status is the target variable). The input variables are the purchasing patterns (i.e., expend and numpur) and demographic characteristics (i.e., age, gender, income, race, marital and member). Suppose that MailPurchase has decided to use SPSS Clementine (in particular, the C5.0 node) to construct the prediction model. The results are summarised in Figures 3.17 and 3.18.

As shown in Figure 3.17 (see bottom right table), the accuracy rates for purchasers and non-purchasers are 66.21% and 68.93%, respectively. The overall accuracy rate of the decision tree model is 67.50% (i.e., [488 + 457]/1400). This overall accuracy rate is better than those of the logistic regression and neural network models discussed earlier. (More detailed comparison of models will be discussed in the next chapter). The input variables included in the decision tree (in descending order of importance) are race, member, age and income (see middle right table of Figure 3.17). No statistical tests are associated with these results. Instead, the importance of the input variables is assessed on the basis of information gain.

The decision rules are also given in Figure 3.17 (see middle right table). For example, the last decision rule indicates that if an existing customer is an Indian, then he/she is predicted as a purchaser (with a confidence of 0.767 – to be explained later). The other rules are more complicated in that they involve more input variables. To illustrate, the first decision rule predict a purchaser if an existing customer is a non-Indian who is a member of the loyalty card programme and who is 42 years old or below.

For intermediate nodes, the number of observation is shown in brackets (e.g., the number of Chinese, Malay and Others is 1,117). For terminal nodes, in addition to the number of observations, the confidence of prediction (i.e., the probability of the predicted group in the node) is also given. For example, for the last decision rule, the rule is derived from 283 observations in the data set and of these, 76.7% are purchasers.

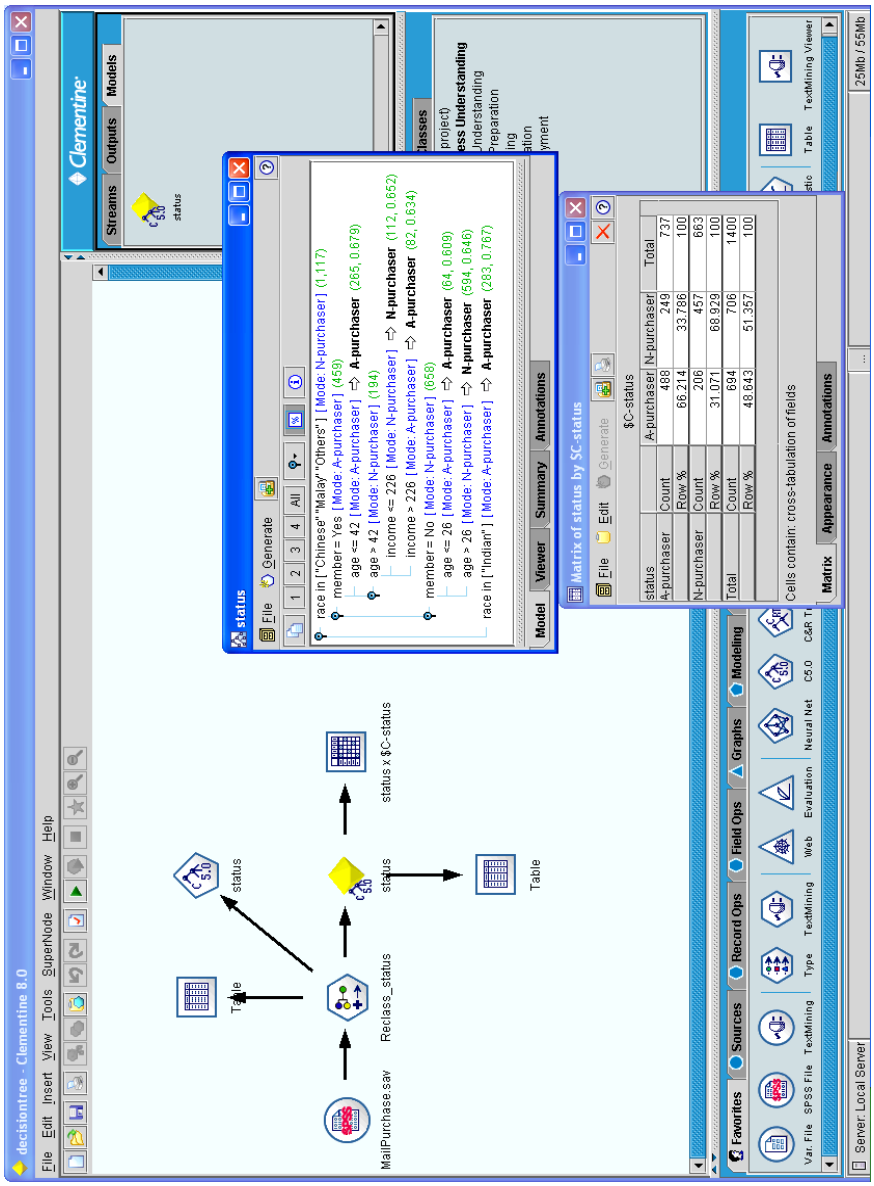


Figure 3.17 Decision Tree Results – Rules and Accuracy Rates

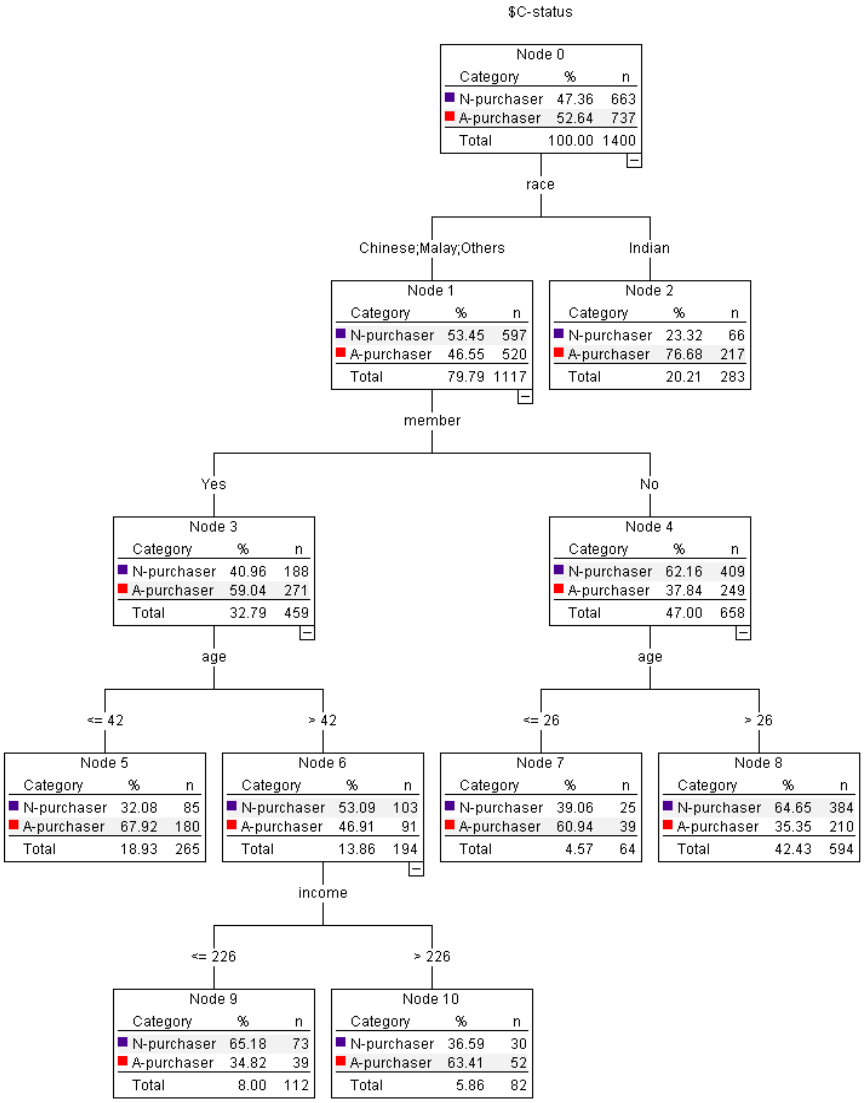


Figure 3.18 Decision Tree

The confidence of prediction is similar to the confidence of association rules in the previous chapter. It also serves as probability estimates of the decision tree model. However, they are not as sophisticated as the probability estimates generated by the logistic regression model.

Figure 3.18 visualises the decision tree and makes the decision tree results easier to understand and interpret. For example, node 8 indicates that an existing customer has a 64.65% probability of being a non-purchaser if he/she is a non-Indian who is not a member of the loyalty card programme and who is above 26 years of age. As in the cases of the logistic regression and neural network models, assuming that the accuracy rates are acceptable to MailPurchase, the decision tree model can be used for targeting the most likely customers for the next marketing campaign.

3.5 Summary

This chapter focuses on predictive modelling, which is one of the most common and important applications in data mining. In particular, this chapter discusses regression, neural networks and decision trees.

There is no one best data mining tool for predictive modelling as each of these models has its own pros and cons. For example, regression is easy to apply and use but it is cumbersome to include non-linear and interaction effects in regression models. To do so, additional terms are needed in the regression model (e.g., X^2 or X_1*X_2). In addition, the functional form of the non-linear and interaction effects has to be specified. On the other hand, neural networks are a very good universal approximator but their results are not very interpretable. A neural network is often referred to as a "black box" that reveals very little about the relationships captured by the model.

Decision trees have very interpretable results that can be visualised and that can also be converted into decision rules. Given this, it is not

surprising that in the KDD-Cup (a data mining competition) in the year 2000, the use of decision trees outnumbered other data mining tools more than two to one (see Kohavi, Rothleder and Simoudis, 2002). Decision trees can also handle missing values well (by assigning them as a separate category). However, decision trees are not truly multivariate in that at every split, only one input variable is considered at a time.

There is no one model that is superior under all circumstances. This is especially so because different models can lead to different results depending on the actual data being mined. Hence, in practice, it is common to construct all the regression, neural network and decision tree models and then assess the competing models to identify a champion (i.e., or so-called "best") model. This chapter focuses on the methods of predictive modelling. The assessment of prediction models and the comparison of these models comprise the content of the next chapter.

Finally, the two-category classification discussed in this chapter can be extended to target variables with more than two categories. For decision trees, this will be reflected in the decision tree nodes and decision rules. For neural networks, more output nodes will be added and predictions to all the categories will be made. For logistic regression, it is a little more complicated. Suppose that there are q number of categories in the target variable. Then, one of these categories will be used as a reference category and (q – 1) number of equations will be generated. These equations can be used to derive the probability of an observation belonging to each of the q categories. This observation is then classified as belonging to that category that has the highest probability. Generally, the greater the number of categories for classification, the lower the overall accuracy rate of the model will be because there is now more scope for "confusion" (i.e., misclassification).

Chapter References

Afifi, A. A. and Clark, V. (1996), *Computer-aided Multivariate Analysis*. Chapman & Hall, London.

Berry, M. J. A. and Linoff, G. S. (1997), *Data Mining Techniques: For Marketing, Sales, and Customer Support*. John Wiley & Sons, Inc., New York.

Berry, M. J. A. and Linoff, G. S. (2000), *Mastering Data Mining: The Art and Science of Customer Relationship Management*. John Wiley & Sons, Inc., New York.

Dhar, V. and Stein, R. (1997), *Seven Methods for Transforming Corporate Data into Business Intelligence*. Prentice Hall, New Jersey.

Greene, W. H. (2003), *Econometric Analysis*. Prentice Hall, New Jersey.

Kohavi, R., Rothleder, N. J. and Simoudis, E. (2002), "Emerging trends in business analytics", *Communications of the ACM*, Vol. 45 No. 8, pp. 45-48.

Smith, K. A. and Ng, A. (2003), "Web page clustering using a self-organizing map of user navigation patterns", *Decision Support Systems*, Vol. 35 No. 2, pp. 245-256.