Identifying and Overcoming Common Data Mining Mistakes

Doug Wielenga, SAS Institute Inc., Cary, NC

ABSTRACT

Due to the large amount of data typically involved, data mining analyses can exacerbate some common modeling problems and create a number of new ones. These problems can greatly increase the time that it takes to develop useful models and can hamper the development of potentially superior models.

This paper discusses how to identify and overcome several common modeling mistakes. The presentation begins by providing insights into common mistakes in data preparation; it then follows the data flow of a typical predictive modeling analysis through setting variable roles, creating and using data partitions, performing variable selection, replacing missing values, building different types of models, comparing resulting models, and scoring those models using SAS® Enterprise Miner™. The paper concludes with a discussion of common issues with cluster analysis and association/sequence analysis. Applying these techniques can greatly decrease the time it takes to build useful models and improve the quality of the models that are created.

INTRODUCTION

We seek to identify several common data mining mistakes and provide suggestions for how to overcome the problems associated with these mistakes. The collection of mistakes and corrective measures discussed here should not be considered complete, because a full discussion could fill several volumes. However, the collection does include those mistakes that have been frequently observed and can almost always be overcome. Please note that the choice of the best approach is highly subjective, and it is possible that certain suggestions recommended in this paper are not well suited for a particular situation. In the end, it is the responsibility of the analyst to choose the most appropriate method for a given analysis. The discussion that follows seeks to raise awareness of certain situations that can lead to undesirable results and describes ways in which those situations can be addressed.

PREPARING THE DATA

It often requires more time to prepare the data than to analyze the data. Unfortunately, deadlines can force shortcuts to be made since a good answer today is often more desirable than a better answer tomorrow. These shortcuts often minimize the time spent in data preparation, but failing to prepare the data adequately can greatly increase the analysis time, thereby minimizing the value of the shortcut. Additionally, the resulting models often perform more poorly than a model developed with adequate data preparation. The mistakes involved with data preparation often appear in the form of failing to consider enough variables, improperly preparing (or failing to prepare) categorical predictors, and improperly preparing (or failing to prepare) continuous predictors.

FAILING TO CONSIDER ENOUGH VARIABLES

Faced with the potentially daunting task of investigating all of their data, users often want to know which variables to use for a given model. This type of thinking has at least one inherent problem: it relies on the existence of some common subset of traits that can be used to satisfactorily model the problem in question.

Consider the following points:

- Not every company has the same “variables.” Every company has variables that are similar, but they often have a reasonably large number of variables that they collect which their competitors do not capture. The variables that are collected by multiple companies can differ in how they are defined or in how often they are measured.
- Reducing a modeling exercise to the subset of common variables ignores the richness and the uniqueness of a company’s data. If all companies are using the same subset of variables, then each company loses a great opportunity to identify patterns that might be unique to that company. Customizing your model based on your own data enables you to choose the approach that works best in your situation.
- Being satisfied with doing what everyone else is doing leaves you at a disadvantage compared to those who take advantage of the richness of their data. This isn’t merely a question of growth in many markets; it is a question of survival. If your competitor is able to increase retention or cross-sell/up-sell more than your company, your competitor is likely to gain market share over time.

Adopting this one-size-fits-all approach doesn’t make sense in most business situations, and it certainly doesn’t make sense when doing modeling. Having modeled similar problems in many large- and medium-sized companies, I have found information is often gleaned from unexpected sources. The process of analyzing the data provides great

insights into a company's customer base. In some cases, these findings have confirmed some commonly held beliefs, but in other cases those beliefs have been refuted in part or in whole. The process can also lead to the discovery of errors in the database, which provides additional benefits.

To overcome this problem, try to use all of the data that is reasonably accessible within the time allowed for doing the work. It is not uncommon to leave potentially rich data stores untapped due to the limited amount of time available to develop the model. Time spent investigating the larger set of variables benefits all future modeling efforts. The experience gained from previous modeling efforts enables an analyst to identify key variables that are important for modeling the business problem as well as those that are largely useless. As a result, the time it takes to refit a model in the future is typically far less than the time needed to fit it initially. Inevitably, many variables might be useful and might be considered when the model is refit as time allows. Every so often, an analyst needs to revisit some of those variables that were discounted from earlier consideration since internal and external conditions can greatly change the nature of the relationships. It is therefore important to fit and then monitor a model's performance. When the performance starts to wane, it is time to consider refitting the model.

INCORRECTLY PREPARING OR FAILING TO PREPARE CATEGORICAL PREDICTORS

Categorical predictors, and the failure to prepare them properly, are the source of much heartache in modeling. The availability of point-and-click automated methods of handling categorical variables can lead to some unexpected results. Problems with categorical variables take at least three common forms, including having too many overall levels, having levels that rarely occur, or having one level that almost always occurs. These problems manifest themselves in predictable ways.

TOO MANY OVERALL LEVELS

Using a categorical variable with too many levels often results in performance problems. This is a common reason why model processing slows to a crawl or even stops in some cases. This has nothing to do with the software and everything to do with the way in which categorical variables must be estimated. In general, a categorical variable with k levels requires a minimum of k–1 parameters in the model. Since a typical continuous variable requires only one parameter barring interactions or higher-order terms, a single categorical variable with k levels requires the same amount of estimation as k–1 continuous variables. Additionally, data requirements are proportional to the number of parameters in the model. Increasing the amount of data to allow estimation of excessive numbers of parameters can further slow down processing and often generate very little performance improvement.

To overcome this problem, it is critical to evaluate the reason for the large number of levels and answer questions like: Can this variable be represented by a group of variables with far fewer levels? and Is there a higher level hierarchy that makes sense when modeling this variable?" For example, consider the situation where you want to use zip codes in the model. Zip code is a classic example of a variable with far too many levels (in most data sets). Rather than looking at the zip code as the variable of interest, observe that the goal of incorporating zip codes is likely to account for the differences in the geographic, demographic, and/or economic status of the clients. Rather than using zip codes directly, obtain some of these metrics for the zip codes and use this set of variables in place of zip codes. Alternatively, consider using MSA and/or STATE to group the zip codes at a less granular level with far fewer distinct values. Any of these approaches would help you to identify some additional dimensions in the data without creating a model that cannot be well fit due to the excessive number of levels in one predictor.

LEVELS THAT RARELY OCCUR

This problem won't necessarily slow down processing greatly, but it is a potential source of great inefficiency. Many variables encountered in business create great modeling inefficiencies because they have a few dominant levels that account for the majority of the data as well as a large number of levels that are extremely small in comparison. Many of these levels have too few observations to have any real impact on the model fit.

Consider a stockholder meeting where each person can vote in proportion to his or her number of shares. Five of the shareholders own 96% of the stock, and the other 1,000 shareholders own the remaining 4%. Paying careful attention to every person holding shares wastes a lot of time because the outcome is largely decided by the five shareholders who hold virtually all of the stock. The smaller shareholders can have an impact collectively, but they have virtually no impact individually. This problem creeps into predictive modeling when you include grouping variables with a large number of levels without a sufficient number of observations to have any real impact on the outcome.

To overcome this problem, consider grouping some levels of your categorical variable together. In some situations, you can merely group the infrequently occurring categories into an "other" category. In other situations, it might make sense to group the infrequently occurring levels with a more frequently occurring level that seems to make the most sense. This can often be accomplished by choosing a less granular level of a hierarchy. The proliferation of levels can occur due to excessive classification. It is generally far more efficient to generate a set of levels with a nontrivial number of observations, which should make any resulting model more stable and easier to manage.

ONE LEVEL THAT ALMOST ALWAYS OCCURS

In this problem, only one level accounts for virtually all of the observations. This problem is closely related to the preceding one except that this variable might in fact be nearly useless. Modeling success is dependent on being able to differentiate among the observations. As described earlier, categories that have too few observations have a trivial amount of influence on the overall prediction. If the only nontrivial level has virtually all of the observations, there is almost no variability, and therefore no information to help the model differentiate between the possible outcomes.

To overcome this problem, investigate whether the variable can be represented at a level where there is more than one nontrivial level. If only one dominant level appears, the variable is highly likely to be useless in any model since a large portion of the observations cannot be differentiated with respect to this variable. However, in the case of modeling rare events, it is still possible that an infrequently occurring level of the predictor is very useful in predicting the outcome. In this situation, the predictor variable should be prepared so that each level has a nontrivial number of observations from which to estimate the model.

This problem should not be confused with the situation where there are a substantial number of missing values. It is common for programmers to code an event as a "1" when it occurs but to leave the variable missing when it does not occur. These variables appear in SAS® Enterprise Miner™ as UNARY variables because they have only one non-missing level. If the proportion of observations that have a missing value for a certain variable is substantial, this level can be recoded to add the needed variability. Those observations with a missing value for the variable then constitute a second nontrivial level, and the newly recoded variable has the potential to provide useful information about the outcome.

INCORRECTLY PREPARING OR FAILING TO PREPARE CONTINUOUS PREDICTORS

Continuous variables can also be a great source of heartache in modeling. The ability to create rapid transformations of variables can sometimes cause the value of the transformation to be overlooked. Additionally, there are times when it might be useful to consider both the transformed and non-transformed version of a variable in the variable selection process. Problems with continuous predictors take at least four common forms, including being extremely skewed, having a spike at one level and a distribution at other levels, having one level that almost always occurs, or having their time components ignored. These problems manifest themselves in predictable ways as well.

EXTREMELY SKEWED PREDICTORS

Extremely skewed predictors can be problematic because the number of observations available to predict the target varies greatly across the range of the input values. The points in the tails of their distributions can have a great impact on the fitted model. Because most predictions aren't being made at these extreme values, the resulting model fit can be suboptimal for many predictor values of interest. This problem can also result in making the predictor appear far more (or less) important than it actually is.

To overcome this problem, there are at least two strategies available:

1. Find a transformation of the original predictor that stabilizes the variance and generates more consistent support across the range of values.

2. Choose an appropriate binning transformation that does not create too many (or too few) bins in order to enable each portion of the predictors' ranges to be weighted appropriately.

In many cases, it might be desirable to do both of these transformations so that both continuous and categorical versions of the predictor are available for variable selection. The categorical transformation of the variable allows for nonlinearity in the response in regression models, while the continuous transformation should help to stabilize the model fit across the range of the values of the predictor.

It should be noted that nonlinear transformations (that is, log transformations, square root transformations, power transformations, and so on) can introduce some difficulty in interpreting the results because it is often not intuitive to think in terms of the transformed units. For example, log (dollars) is less intuitive than dollars when interpreting results because people don't tend to think on a log scale. However, if the goal is prediction, transforming might achieve a more stable model that deploys better on holdout samples. Also note that if you transform the target variable, you might also be transforming the error term leading to different assumptions on the structure of the model. As a result, the optimal model fit in the transformed space might not be the same as the transformation of the optimal solution in the original space.

A SPIKE AND A DISTRIBUTION

This problem occurs when a predictor equals one value quite frequently but follows a certain distribution elsewhere. Ignoring the duality of this predictor can lead to understating or overstating the importance of the predictor. For example, a variable can be very useful for predicting the target, but this relationship is masked by those observations appearing in the spike. The spike might even occur at a value far removed from the distribution, again making the

overall fit seem stronger (or weaker) than it actually is. Fitting a model to both pieces simultaneously misrepresents the relationship between the predictor and the target.

To overcome this problem, create two new variables from the one in question. Create a flag that indicates whether the value is in the spike, and create another variable from the values of the predictor in the distribution. For the latter variable, set the observations that have the value at the spike to missing. You can later impute this value for regression and neural network models. Choose an imputation strategy that minimizes the impact on the relationship between the predictor and the response. The flag contains information about whether the original value was in the spike, and the second variable contains information to assess the relationship between the response and values in the distribution outside the spike. Alternatively, you might consider fitting a separate model to those variables found in the spike and those found in the distribution. In many cases, fitting a model to each group yields better performance than fitting one model to both. This approach must be used sparingly because it is not practical to take this approach for too many variables.

You might also consider creating a three-level missing value indicator for this variable in order to differentiate between the case where the predictor was part of the spike and was changed to missing; the case where the predictor was present but not modified; and the case where the predictor value was missing in the original data. This three-level indicator could be useful if there is a nontrivial amount of missing data for the predictor in question. Finally, should the continuous portion of the predictor be of limited usefulness, consider transforming the entire variable into an ordinal predictor by binning the predictor values to optimally predict the target.

ONE LEVEL THAT ALMOST ALWAYS OCCURS

This problem is an extreme version of the spike and distribution problem and occurs most frequently as a spike to the extreme left or right of the remaining values. In this example, the distribution is virtually flat except for the spike and accounts for a relatively small portion of the data. Any relationship that is identified is driven almost entirely by the most extreme values. Because so many points are accounted for at the spike, and the largest values—being most affected by changes to the line of fit—have greater leverage, the predictor can appear to be strongly correlated to the target when it actually has limited value.

To overcome this problem, create a new variable that is a binned version of the original continuous variable. In many cases, you might be able to create only a binary predictor due to the limited proportion of observations outside of the spike. Keeping in mind that it is not generally useful to have categorical predictors that have levels with virtually no data, you might be able to create more bins depending on the proportion of the data outside the spike. It is also possible that the common level can represent a group of levels so that using this more granular level of the data actually overcomes the problem by creating an adequate number of nontrivial levels.

IGNORING OR MISUSING TIME-DEPENDENT INFORMATION

This problem can occur when time-stamped or regularly occurring (for example, monthly) data is available. Incorporating historical information into a predictive model can have a dramatic impact, but it must be used appropriately. While most transactional data cannot be modeled in its native form by predictive modeling tools, it can be processed into a form that retains a lot of the information about these transactions and/or periods and is useful for modeling at the same time.

In general, date variables can often be converted to new variables that measure the amount of time that has passed since some event (for example, account opened) or the amount of time before an event (for example, end of contract). When periodic summary statistics are available, it is often best to use rolling windows. Using fixed date summary values such as year-to-date (YTD) statistics can be misleading when scoring future observations. In many cases, people scored early in the year will be low with respect to YTD values, while people scored near the end of the year will be high. Rolling windows consider behavior relative to the current time period. In many cases, it is useful to look at recent behavior as well as less recent behavior to determine whether a change has occurred. For example, you could create lag variables for each of the three to six most recent monthly summaries in addition to looking at the overall average of the three- to six-month periods prior to those monthly periods. It is important to account for seasonality when taking this approach so that seasonal changes are not interpreted as individual changes.

Suppose transactional data is available that can be summarized into periodic data where each row corresponds to an observation, and each column corresponds to a particular summary statistic for a given time. It is often useful to have baseline information and recent information to determine whether changes in baseline behavior are predictive of the target variable of interest. When identified, these changes can then be used as early warning signs allowing for potential intervention to stop an undesirable outcome. Consider evaluating the last several time periods of interest as well as a pooled estimate of baseline behavior based on more distant time periods. Observe that this might greatly increase the total number of variables in the model, so judicious selection of time periods is required. It might be useful to investigate if there is any seasonality in the data. Adjusting for seasonality improves the overall fit and should lead to a better understanding of the overall relationships.

DEFINING ROLES, PERFORMING SAMPLING, AND DEFINING TARGET PROFILES

After the available data has been evaluated and the analyst has determined how to prepare the data, the analyst should consider how much of the data to use for analysis. Historically it was necessary to analyze every observation because the amount of data was so limited, but data mining is typically performed when there is a large amount of data available. It might seem desirable to build models on all of the data, but the cost of time spent analyzing all the data often outweighs the benefit when compared with modeling against a well-chosen sample. The challenge is to identify an appropriate sample so that the analysis of the sample provides valuable insights into what is happening in the larger data set or population. The remaining data can then be used to validate the models that are built. Any sampling strategy needs to take into account the nature of the target variable as well as the number and nature of the predictors. After the variables are chosen and the sample is selected, the target variable must be evaluated to ensure that the modeling strategy is appropriate. In the case of a categorical target, it might be necessary to create a target profile to obtain useful models, particularly when the level of interest is relatively rare in the population or larger data set.

INAPPROPRIATE METADATA

Establishing the correct metadata is critical to the modeling process. The metadata determines how each variable should be used. SAS Enterprise Miner automatically assigns the modeling type and role of each variable based on the name or values of the variables. Unfortunately, this process cannot prevent inappropriate variables (for example, a numeric ID variable) from being seen as a continuous input because numeric data often has a large number of levels. Identification information, date information, and many other categorical variables often appear as numbers and are stored in a numeric variable. Using variables inappropriately in the analysis can easily lead to misleading results.

To overcome this problem, explore each of your variables before running them through a modeling node. You will often find many issues described in earlier sections that must be addressed to make the data as useful as possible for modeling. This can be extremely time-consuming because data mining often involves many hundreds or even many thousands of variables.

INADEQUATE OR EXCESSIVE INPUT DATA

Some analysts believe that sampling their data leads to inferior results, and therefore they seek to analyze the entire population. While this might be true in some situations, it is not typically true when the sample is selected appropriately. After selecting a sample to build candidate models, the remaining data can then be used to compare competing models and to evaluate the final model.

When modeling extremely rare events, sampling is almost certainly necessary in order to obtain a model that can outperform the null model. In the null model, every observation is assigned to the most frequently occurring group for a categorical target or to the mean for a continuous target. In the case of a rare event, this null model can be extremely accurate. For example, consider a binary target where the level of interest occurs only 1% of the time. In this case, the null model would be correct 99% of the time by concluding that none of the observations would be classified as the target event.

In other situations, the existence of large amounts of data provides an opportunity to perform empirical validation of any model that is fit. By fitting the model to a portion of the data (known as the *training* data), resulting models can be compared using the holdout sample (or *validation* data). This holdout sample can be used to evaluate the fitted models to determine how each model performs. Choosing a model that performs well jointly on both data sets provides protection against finding chance associations in the data. Additionally, should enough data be present to split the data set into a third set for final testing of the model, the user has an opportunity to obtain a final unbiased estimate of model performance by using this data set known as the *test* data set.

While it is easy to understand the ramifications of sampling too few observations, sampling an excessive number of observations might increase computation time without greatly affecting the resulting prediction. This additional processing time can often be substantial so that there is far less time to evaluate and improve intermediate models. Using too much data for the training and/or validation data sets also leaves little or no data for use in the test data set. This makes it difficult to obtain an unbiased estimate of model performance.

It is equally dangerous to undersample the data. Undersampling occurs frequently when the analyst plans on modeling a binary categorical target variable with a relatively infrequent target level. In the case of two possible outcomes and a fixed sample size, the maximum power occurs when you sample the same number of observations from each group (assuming equal variances and holding all the other parameters constant). However, eliminating data to obtain these equal-sized samples reduces power more than the original imbalance in group sizes. As a result, you should not eliminate additional data in order to create equal sample sizes (Muller and Benignus 1992).

When modeling a binary target variable where the levels are not balanced, an analyst often samples the data to get the sample proportions to 50% for each outcome. Unless adjustments are made to resulting probabilities, the

resulting probabilities are then unrealistic because they reflect the sample and not the population. In the case of a rare event, more problems surface because there might be a relatively small number of outcomes of interest. A sample that contains all of the rare events and the same number of cases with alternative outcomes is often a very small data set. Additionally, the small number of cases sampled for the alternative outcome is likely to inadequately represent this group, which represents a large proportion of the population.

In many cases, it would be far more appropriate to sample proportionally and to handle the lack of balance by adjusting the target profile. In situations where the event is very rare or where the relationship between the predictors and the target is very weak, sampling proportionally might not be feasible. In this case, it is reasonable to oversample, but taking an equal number of observations from each group yields a sample that is not representative of the group without the rare event. It is important to create a target profile that generates the desired decision rule when oversampling the data. The next section suggests some strategies for creating a target profile.

No simple way of determining a minimum number of observations exists. Most strategies for calculating sample size focus on the number of observations that would be needed to attain a particular significance level or to achieve a certain level of accuracy given a set of assumptions. In practice, hardware limitations often dictate the size of a sample that can be reasonably used in a given model. In many cases, a threshold exists beyond which the cost of processing time or disk space requirements increases far more rapidly than the benefits of an increased sample size.

In cases where an extremely large number of variables are available, it is often beneficial to do some variable selection using a smaller initial sample to identify those variables that appear to have little usefulness in predicting the outcome of interest. Unimportant categorical variables can be removed, and continuous variables can be removed or reduced via a data reduction technique such as principal components. By reducing the original set of variables, a larger number of observations can be analyzed in a timely fashion.

INAPPROPRIATE OR MISSING TARGET PROFILE FOR CATEGORICAL TARGET

It is essential to understand how models are evaluated in order to understand the impact of choices you make regarding the target profile. Failure to specify a target profile is equivalent to choosing the default target profile. Using this default profile can lead to suboptimal results in cases where the target classes are unbalanced and/or when there are greatly different costs for misclassification.

For a categorical target, the default profile assigns equal profit to correctly predicting each outcome successfully. The decision rule is very simple in this case—the outcome with the highest probability is chosen. This is well and good except for situations where the sample proportions are very unbalanced, or when the outcomes have very different misclassification costs.

TARGET VARIABLE EVENT LEVELS OCCURRING IN DIFFERENT PROPORTIONS

In this situation, one event occurs less frequently than the other. Modeling rare events is very common in predictive modeling, so this happens more often than not. Even if the sample is balanced, adjusting the priors to reflect an unbalanced population proportion adjusts the posterior probabilities to take into account the oversampling. The default decision rules are unlikely to select outcome levels that occur less frequently in the population.

Suppose you have a binary target where the event of interest occurs in 10% of the population (that is, Pr(event)=0.1), and the non-event occurs in 90% of the population (that is, Pr(nonevent) = 1–Pr(event) = 1–0.1 = 0.9). After building a predictive model, suppose an observation is predicted by the model to be 40% likely to have the event (that is, Pr(event)=0.4). For this observation, Pr(nonevent)=1–0.4=0.6. Note that although this observation is four times as likely to have the target event as an observation taken at random from the population, the observation is more likely to have the non-event. If you weight a correct prediction of either outcome equally, you will end up predicting the non-event for this observation because this outcome is more likely. In fact, the observation must be more than five times as likely to occur (Pr(event)>0.5) to have the predicted probability of the target event be greater than the predicted probability of the non-event. Correspondingly, an observation predicted to be less than five times as likely to have the target event is predicted into the non-event group using the default decision rule. This imbalance can cause variable selection to drop all of the predictors because no model can be built using the available predictors, which can identify observations that are more than five times as likely to have the target event.

To overcome this problem, specify a different profit to predicting the outcome events correctly. In most situations, the target is binary, and the target event occurs more rarely than the alternative. In situations where the target is not binary, the modeler can apply the logic described below to multiple models, with each model treating one event as the target event and the remaining events as non-events. Alternatively, the logic can be extended to *k*–level target variables, but this is not addressed in this paper.

Suppose the response variable Y takes on values of 1 or 0 where 1 is the event of interest. Suppose further that X_1, X_2, X_3, ... , X_n represent the input variables of interest.

Assume

P is the unadjusted predicted probability of the target event based on the model

P_{adj} is the adjusted predicted probability of the target event based on the model

p_1 is the proportion of target events in the sample

$p_0 = 1 - p_1$ is the proportion of non-events in the sample

τ_1 is the proportion of target events in the population

$\tau_0 = 1 - \tau_1$ is the proportion of non-events in the population

then the adjusted probability for a particular observation is

$$P_{\text{adj}} = \frac{(P * \tau_1 * \text{p}_0)}{[(P * \tau_1 * \text{p}_0) + ((1 - P) * \tau_0 * \text{p}_1)]}$$

Further assume that the decision matrix of predicted versus actual is

		Predicted	
		1	***0***
Actual	***1***	D_{tp}	D_{fn}
	0	D_{fp}	D_{tn}

where

D_{tp} = the profit of correctly predicting the event of interest (tp = True Positive)
D_{fp} = the cost of incorrectly predicting the event of interest (fp = False Positive)
D_{fn} = the cost of incorrectly predicting the non-event (fn = False Negative)
D_{tn} = the profit of correctly predicting the non-event (tn = True Negative)

SAS Enterprise Miner classifies each observation based on an extension of a classifier known as *Bayes' rule,* which minimizes the expected loss. For a binary target taking on levels 1 (positive) and 0 (negative), Bayes' rule classifies an observation as 1 (positive) if

$$posterior\ \ probability > \frac{1}{1 + \left(\frac{\text{cost of false negative}}{\text{cost of false positive}} \right)} = \frac{1}{1 + \left(\frac{\text{D}_{\text{fn}}}{\text{D}_{\text{fp}}} \right)}$$

In this situation, SAS Enterprise Miner classifies an observation as 1 (positive) if

$$P_{\text{adj}} > \frac{1}{1 + \left(\frac{\text{D}_{\text{tp}} - \text{D}_{\text{fn}}}{\text{D}_{\text{tn}} - \text{D}_{\text{fp}}} \right)}$$

This equation is identical to Bayes' rule when $D_{tp} = D_{tn} = 0$. The generalization enables the user to specify the decision rule in terms of profit and cost associated with correct and incorrect predictions of either type. For example, a decision rule can seek to maximize profit by assigning the profit associated with a true positive and the profit associated with a true negative. By default, SAS Enterprise Miner uses $D_{tp} = D_{tn} = 1$, and $D_{fn} = D_{fp} = 0$. As a result, the rule simplifies to

$$P_{\text{adj}} > \frac{1}{1 + \left(\frac{1 - 0}{1 - 0} \right)} = \frac{1}{1 + \left(\frac{1}{1} \right)} = \frac{1}{2} = 0.5$$

when no target profile is created. The equal value assigned to predicting either outcome correctly leads to the decision rule that chooses the outcome with an adjusted probability greater than 0.5. In this example dealing with a

binary target, the probability of one outcome can be calculated by subtracting the probability of the other outcome from one, so this rule is equivalent to choosing the outcome with the highest adjusted posterior probability. If no prior is specified (that is, no oversampling has been done), then random sampling is assumed. In this case,

$$E(p_0) = \tau_0$$
$$E(p_1) = \tau_1$$
$$P_{\text{adj}} = \frac{P * \text{p}_1 * \text{p}_0}{(P * \text{p}_1 * \text{p}_0) - ((1-P) * \text{p}_0 * \text{p}_1)} = \frac{P * \text{p}_1 * \text{p}_0}{\text{p}_1 * \text{p}_0 * (P + (1-P))} = \frac{P * \text{p}_1 * \text{p}_0}{\text{p}_1 * \text{p}_0 * 1} = \frac{P * \text{p}_1 * \text{p}_0}{\text{p}_1 * \text{p}_0} = P$$

Let P^* represent the probability above which the model predicts the outcome to be the target event. It follows that

$$P^* = \frac{1}{1 + \left(\frac{\text{D}_{\text{tp}} - \text{D}_{\text{fn}}}{\text{D}_{\text{tn}} - \text{D}_{\text{fp}}} \right)}$$

In this simple but common case of a binary target with a relatively infrequent target level of interest, we can control the threshold for the cutoff by judiciously choosing D_{tp}, D_{fn}, D_{fp}, and D_{tn}. In practice, there are many combinations of these values which will yield the same decision because it is the ratio of differences which controls the value in the denominator.

In practice, there is often only a hard cost of action associated with predicting the target event. In a direct marketing scenario, predicting someone wouldn't buy means that the person will likely not be sent a mailing. Because the cutoff probability only depends on the ratio of the differences, assume that $D_{fp}=D_{tn}=0$ yielding

		Predicted	
		1	***0***
Actual	***1***	D_{tp}	0
	0	D_{fp}	0

In this example, the equation then reduces to

$$P^* = \frac{1}{1 + \left(\frac{\text{D}_{\text{tp}} - 0}{0 - \text{D}_{\text{fp}}} \right)} = \frac{1}{1 - \left(\frac{\text{D}_{\text{tp}}}{\text{D}_{\text{fp}}} \right)}$$

so choosing D_{tp} and D_{fp} appropriately yields the desired threshold. For example, suppose anyone who was predicted to have a value of 1 would be targeted, for which the marketer incurs a cost of \$1.00. Assume responders spend \$10.00 on average, and non-responders spend \$0.00. The value of D_{tp} is then \$10.00 – \$1.00 = \$9.00 and the value of D_{fp} is then \$0.00 – \$1.00 = –\$1.00. Because we are creating a ratio, the units cancel and we end up with

$$P^* = \frac{1}{1 - \left(\frac{9}{-1} \right)} = \frac{1}{1 - (-9)} = \frac{1}{1 + 9} = \frac{1}{10} = 0.1$$

Similarly, if the value of a responder was only \$4.00 (still assuming a relative cost of \$1.00), D_{tp} = 4 – 1 = 3, and D_{fp} = 0 – 1 = –1, and therefore

$$P^* = \frac{1}{1 - \left(\frac{3}{-1} \right)} = \frac{1}{1 - (-3)} = \frac{1}{1 + 3} = \frac{1}{4} = 0.25$$

You can see a pattern emerge here. If you assume a fixed cost of one unit and therefore assign the value of –1 to the cell associated with D_{fp}, then choose the value of D_{tp} so that the desired probability for the cutoff can be computed by

$$P^* = \frac{1}{1 + \mathrm{D}_{\mathrm{tp}}}$$

Assuming that $D_{fp} = -1$, choosing D_{tp} equal to

- 1, implies $P^* = \frac{1}{1+1} = \frac{1}{2} = 0.5$
- 4, implies $P^* = \frac{1}{1+4} = \frac{1}{5} = 0.2$
- 9, implies $P^* = \frac{1}{1+9} = \frac{1}{10} = 0.1$

Observe that this approach incorporates the cost into the "profit" with a particular action. If you set up the target profiler with a fixed cost of 1 for the event of interest, SAS Enterprise Miner takes this cost and subtracts it from the values set up in the decision matrix.

As a result, to obtain a cutoff of 0.1, you should specify

		Predicted	
		1	***0***
Actual	***1***	9	0
	0	–1	0

if no cost is specified, and

		Predicted	
		1	***0***
Actual	***1***	10	0
	0	0	0

if you have specified a cost of 1.0 to be used for those predicted to have the target event (in this example, Y=1). In some situations, the target event is so rare that using the prior probability might still generate the null (intercept only) model even when extremely large weights are put into the profit matrix, unless variable selection settings are modified. In order to deal with this situation, it is often better to oversample and adjust the profit matrix based on the oversampled data, not on the original priors. In doing so, you must make a posterior adjustment to the probabilities unless you are interested only in the sort order of the observations. In this case the adjustment to the probabilities is likely to affect how several observations are classified, but the sort order of the observations does not change.

DIFFERENCES IN MISCLASSIFICATION COSTS

This problem occurs when you are using a sample where the decision rule is not in alignment with the actual misclassification costs. Model selection seeks to find the model that optimizes a particular decision rule. When the decision rule does not reflect the true cost of misclassification, the model selected might perform suboptimally. To overcome this problem, create the target profile to be as close to the actual decision rule as possible. In doing so, variable selection and model assessment return the best possible model.

PARTITIONING THE DATA

After determining how to prepare and sample the data, consider how to partition the data for analysis. Modern data mining methods allow for extremely flexible modeling strategies to be put in place in a relatively short amount of time. The flexibility of these modeling strategies enables the data to be fit much more closely. Unfortunately, flexible methods can lead to overfitting even with large amounts of data. When sufficient data is present as it usually is in data mining, it is important to choose appropriate holdout samples. Mistakes are commonly made in misunderstanding the roles of the partitioned data sets and in using inappropriate amounts of data for one or both holdout samples.

MISUNDERSTANDING THE ROLES OF THE PARTITIONED DATA SETS

In SAS Enterprise Miner, three key data sets are available from the Data Partition node—the training, validation, and test data sets. The training data set is used to build competing models; the validation data set can be used to compare competing models within a node or across nodes; and the test data set is intended to provide an unbiased estimate of how well the final model performs in practice.

The most common mistake is to misunderstand how SAS Enterprise Miner uses these data sets by default. When fitting a neural network, a tree, or a stepwise regression in a modeling node, SAS Enterprise Miner uses the validation data set by default—if it is available—to select among the models fit within any specific modeling node. The test data set is not used for model selection in the modeling nodes, but predicted values and fit statistics are computed for these observations as well. However, when comparing models fit by different modeling nodes in the Model Comparison node, SAS Enterprise Miner selects the model that performs the best on the test data set when it is available, by default. In this situation, the performance on the test data set does not provide an unbiased estimate of model performance because it is now being used in model selection and has effectively become a secondary validation data set. Of course, the default can be changed to use the performance on the validation or training data sets in the Model Comparison node, but the test data set is used by default to choose the best model when it is present. If no test data set is available, it selects the best model based on the validation data set. If neither a test nor a validation data set is available, it selects the best model based on the training data set.

Using the test data set as a secondary validation data set might have one advantage. While the validation data set helps reduce the bias that is introduced using the training data set for building candidate models, the test data set helps minimize the bias that is introduced using the validation data set for selecting the best model within a given modeling node. In practice, the performance across the training, validation, and test data sets should not be markedly different. If the performance differs greatly across these data sets, it might point to an overfit model, a nonrepresentative sampling strategy, or an inadequate sample size. Regardless of how you use the test data set, it is important to understand the conditions under which the results will be biased. In general, the test data set can provide an unbiased estimate of model performance only if it is used after a single final model has been selected.

FAILING TO CONSIDER CHANGING THE DEFAULT PARTITION

By default, the Data Partition node partitions raw data into training (40%), validation (30%), and test (30%) data sets. The node stratifies these samples on the target variable by default when the target is a class variable. Unfortunately, there is not a unique way to correctly divide the observations into training, validation, and/or test data sets. As a result, the user must exercise caution to ensure that this allocation is appropriate for the problem at hand. Recalling the earlier discussion about sample size, the key to allocating data sets is to ensure that you have a sufficient number of observations in the training and validation data sets. If there are not enough observations to split out a test data set, it is better to use the available data for the training and validation data sets.

On occasion, the number of observations with the target event might be so low that the data should not be partitioned at all. In these situations, proceed with extreme caution because the modeling nodes have no way to evaluate potential overfitting problems. Regression models and decision tree models can be pruned judiciously by an expert familiar with the underlying relationships. However, no such protection is available from perhaps the most flexible modeling method, neural networks. It would be best to avoid using neural networks when insufficient validation data is present. In practice, there might often be fewer than the desired number of observations that have a target event of interest. It is up to the individual analyst to determine how this splitting, if any, should be done.

CHOOSING THE VARIABLES

After the raw sample has been taken and (typically) the partitioning has been done, variable selection is required to identify a useful subset of predictors from a potentially large set of candidate variables. Because regression models (linear and nonlinear regression models such as neural networks) operate only on complete observations with no missing values for independent or dependent variables, it is important to replace any missing values for cases that you want to consider. Because decision trees handle missing values automatically, this is not an issue for this type of model.

In many cases, imputation should be done both before and after variable selection, because it can be instructive to compare the variables selected on the imputed data to those selected on the raw data. Missing values can be present due to coding efficiency (that is, only people meeting certain criteria have a non-missing value or flag) or due to incomplete data. The former scenario is data preparation rather than imputation, which involves guessing at unknown values. Performing imputation in the latter scenario might affect which variables are selected. If different variables are selected when imputation is performed, it might point to problems in the imputation process. Regardless of whether variable selection is done before or after imputation, it is important that any required data preparation is already done to address potential problems among the continuous or categorical inputs as described earlier. Mistakes in variable selection include failing to evaluate the variables before performing variable selection, using only one type of variable selection, and misunderstanding or ignoring variable selection options.

FAILING TO EVALUATE THE VARIABLES BEFORE SELECTION

Data mining analyses routinely consider a large number of variables. The addition of time-dependent variables and/or summary statistics, transformed variables, and missing value indicators can create an unusually large number of predictors. Running these potential input variables through a variable selection method before preparing the data as described earlier can lead to the selection of variables that appear to be important but that do not generalize well to the population. In many cases, it would be better to drop certain variables rather than to include them due to the proportion of missing values, the proportion of values in the spike (if one exists), and/or the excessive number of levels. Several modeling methods have the ability to group some of these levels and can address some of the problems associated with extremely skewed predictors (that is, binning transformations). It is far better to transform problematic variables or remove them from consideration.

For those with a large number of variables to begin with, the process of evaluating all of the variables beforehand might sound particularly distasteful. While automated methods can provide a great deal of protection against unwanted modeling problems, they are not perfect. Things that look fine numerically might be inappropriate due to other reasons. For example, a particular variable might be routinely used in a way that its name and/or label do not imply. The analyst is ultimately responsible for the quality of the fitted model, and failing to investigate the value of the original input variables can result in models that perform poorly in practice. In this situation, it is useful to remember the adage “garbage in, garbage out.”

The process of investigating the entire set of variables and identifying the appropriate way to use the information they contain adds value to all future models and not just the one in question. In some cases, this exploration can identify a subset of the variables for modeling in the future, thereby reducing the modeling time for other models. Additionally, most variable transformations are done to improve the usefulness of an observation, regardless of the target. Even if it is known that a variable should be binned to optimize the relationship to the target, this step can be taken without excessive investigation in the future, allowing additional reduction in the modeling time. Standard transformations that are desired can be incorporated into the ETL process to reduce the typical modeling time even further.

In general, the approach taken must adapt to the constraints of the situation, and it is not always feasible or practical to investigate every possible variable before going through an automated selection and/or transformation process. In these situations, it is critical to review the variables selected by the process to ensure that the resulting model is choosing variables that should generalize well to the population.

USING ONLY ONE SELECTION METHOD

SAS Enterprise Miner provides several different methods for performing variable selection. Several nodes have selection capabilities, including the Variable Selection node, the Tree node, and the Regression node. Limiting variable selection to one of these nodes and/or one of the methods available in one of these nodes can miss important predictors that could improve the overall model fit. Limiting selection can also include predictors that should not be included in the final model. In situations where there are a limited number of variables, the benefit of performing variable selection in a variety of ways might be limited. However, the benefit generally increases as the number of variables and/or the complexity of the relationships increases. Because it is impossible to know beforehand which method is best for a given situation, it is useful to consider the variables selected from a variety of variable selection methods.

For categorical targets, the Variable Selection node provides three different methods, including the R^2 method, the χ^2 method, and the combined method. Only the R^2 method is available when the target is continuous. In the combined method, variables are kept as inputs only if they are selected by both the R^2 method and the χ^2 method. In practice, the combined method provides the most protection against overfitting, but this is an incomplete selection strategy considering that the variable selection is performed purely on the training data set. As a result, predictors might be chosen that are not useful on holdout data.

Decision tree and stepwise regression models use the validation data set to obtain their final result by default, which provides some protection against overfitting. However, other issues make these methods undesirable as the only method for modeling. Decision tree models are excellent with respect to modeling interactions and nonlinear relationships, but they do not model simple linear relationships particularly well. Stepwise regression models cannot detect nonlinear relationships unless the relationship is added to the default model. For example, a quadratic relationship would not be evaluated unless a quadratic term was added to the model. Additionally, stepwise regression models have been shown to optimize chance associations in the data, potentially leading to misleading results. Finally, stepwise regression is typically the slowest variable selection method because it is trying to perform variable selection and model fitting simultaneously.

To overcome these problems, particularly when large numbers of variables are involved, consider using several different variable selection methods and create a pool of predictors based on all variables that are chosen by any of the methods. This might require some manual effort, but it is the safest way to ensure that all variables are fairly considered. After the initial cut is made, you could perform a secondary variable selection making sure to use a

method that protects from overfitting, such as a decision tree or a stepwise regression. While the perils of these methods were described earlier, we have chosen variables that were selected by at least one variable selection method as our initial candidates, which should limit the impact of any of these problems. Rather than deciding on a final set of modeling variables, consider evaluating models based on different sets of candidate inputs before determining the final model.

MISUNDERSTANDING OR IGNORING VARIABLE SELECTION OPTIONS

The Variable Selection node can operate using two main modes, the χ^2 mode and the R^2 mode. The node can also operate using the combined results of the χ^2 mode and the R^2 mode. This combination is accomplished by keeping only variables that are selected by both criteria. Regardless of the mode that is selected, it is critical to understand the default settings as well as how to change them in order to achieve the desired results.

CHOOSING SETTINGS IN THE χ^2 MODE

In the χ^2 mode, the user can specify three additional options: one option determines the number of equally spaced bins to create for the continuous input variables; one option controls the number of passes that are made through the input data when performing binary splits; and one option specifies a critical value to determine whether or not a predictor is going to be retained or not. By default, the critical value is 3.84, which corresponds to $\alpha=0.05$ for a χ^2 distribution with one degree of freedom. Rather than acting as a final judge about whether or not a variable is actually important or not, this critical value acts as a measuring stick. Any variable that fails to meet the minimum level of significance is excluded.

Unfortunately, by definition, most data mining applications have a large amount of data. As the number of observations increases, the more significant a particular test statistic becomes for a given effect size. In other words, a difference that is not significant at a given sample size often becomes significant at a much larger sample size. In the case of data mining, the traditional $\alpha=0.05$ threshold might allow far more variables into the model than is desirable in many situations. Trying different values of α enables the analyst to vary exactly how many variables are being retained. In situations where further variable selection is being done, there is no penalty for retaining some of the less useful variables until a later stage. However, if this node is being used for selecting the final model, it would be prudent to review the practical importance of some of the less important variables to assess whether or not they should be included in the final model.

Regarding the other χ^2 settings, increasing the number of passes might obtain a slightly better fit but will take additional processing time, while decreasing the number of passes might do somewhat more poorly but will run faster. In data mining, the number of observations and the number of variables can be extremely large, so it might be necessary to lower the number of passes and/or raise the critical value in order to speed up processing. Similarly, lowering the number of bins below the default 50 bins might speed up processing as well.

CHOOSING SETTINGS IN THE R^2 MODE

In the R^2 mode, the squared correlation coefficient (simple R^2) for each input variable is computed and compared to the default Minimum R-Square. Variables that have a value lower than this minimum are rejected. Following this initial pass, a forward stepwise selection is performed. Variables that have a stepwise R^2 improvement less than the cutoff criterion are rejected. Other options can be used to request interactions in addition to grouped versions of the continuous variables (AOV16 variables) and categorical variables (group variables). The AOV16 option creates up to 16 bins from each continuous input based on analysis of variance.

As with the discussion of the χ^2 mode, the user can set a minimum threshold for inclusion by adjusting the minimum R^2 value. Setting this value lower tends to include more variables while setting it higher excludes more variables for a given data set. The maximum number of variables option controls the maximum number of variables that will be included, so the number of variables retained might be less than the number that meets the minimum R^2 value. Again, allowing additional variables into the candidate set of variables for the model is appropriate if additional variable selection is being done, so changing these values is probably not necessary unless far more or far fewer variables are available than desired. The AOV16 variables allow for nonlinear relationships between each predictor and target, but this also increases the number of parameters to estimate. Similarly, creating interactions increases the numbers of parameters to estimate. As a result, use these methods sparingly. You should not need to group categorical data unless you have too many levels in the first place.

REPLACING MISSING DATA

By default in the Replacement node, SAS Enterprise Miner imputes the missing values for interval variables by using a sample mean, and imputes the missing values for categorical variables by using the sample mode. Unfortunately, there are several situations where these methods might be less than optimal. The size of this problem is related to the proportion of observations with missing values as well as to the relationship to the target variable in question. The impact of missing values can be surprising because the proportion of observations with complete data is often very small. As the number of variables increases, the chance that a given observation will have missing values increases. Regardless of the imputation method chosen, it is important to investigate missing values to evaluate

whether the observations with missing values are somehow related to the target value. Mistakes in imputation often arise from failing to evaluate the imputation method used or from overlooking missing value indicators.

FAILING TO EVALUATE IMPUTATION METHOD

In the case of interval variables, a missing value often implies that the variable equals a certain value (often zero). In situations where an appropriate value is known, it is far more meaningful to use the actual implied data rather than to use some method to guess what the value might be. Using the mean for all the missing interval values can take an observation that was unusual with respect to a given variable and make it look typical. When the proportion of missing values is relatively high, this can lead to one of the distributional problems described earlier, by creating a spike somewhere in the distribution. In other cases, the value of the variable might be critical to accurately predicting the outcome. In this situation, it would be better to use a tree to predict the value, thereby using all available information to obtain the best prediction of the missing value. Because using the tree imputation creates a separate model for each imputed variable, this method should not be used for all variables, particularly when a missing value implies an appropriate imputed value or when the variable is of little relative importance to the model.

In the case of categorical variables, the missing value can be naturally handled by treating the missing value as an additional class of the variable. In situations where the proportion of missing observations is extremely small, this would be a mistake because it is adding an additional parameter to the model for a trivial number of observations. In this situation, it would be more appropriate to use the default (mode) or to impute using a tree if the variable appears to be important in predicting the response. As with continuous variables, the tree method should not be used for all variables unless it is truly needed.

OVERLOOKING MISSING VALUE INDICATORS

Decision tree models have a distinct advantage over regression and neural network models, because tree models can use missing values directly without having to impute. In cases where a missing value is related to the target, the tree can respond appropriately because it knows which observations had missing values. However, in a regression or neural network model, this missing value has been replaced by the mean or some other value so that the observation can be considered in the analysis. Without taking additional steps, the regression and network models now have no idea which observations originally had missing values.

To overcome this problem, you can create missing value indicators for those variables that behave differently with respect to the outcome when they have a missing value. Rather than creating a missing value indicator for all variables, thereby doubling the number of variables under consideration, investigate the relationship of the missing values to the response to identify variables for which this would be helpful. Also consider the proportion of observations with a missing value for each variable so that you are not creating a new categorical variable where virtually all of the observations are in the same class.

FITTING LINEAR REGRESSION MODELS

Linear regression models are very popular for both their simplicity and their acceptance. While neural networks and decision trees to a lesser extent are sometimes viewed with skepticism, regression models are broadly accepted as being useful and interpretable. Unfortunately, the popularity of a method can sometimes decrease sensitivity to its potential drawbacks. Mistakes in fitting linear regression models often result from overusing stepwise regression or from inaccurately interpreting the results.

OVERUSING STEPWISE REGRESSION

Stepwise regression is popular for its ease of use and its ability to generate a model with limited involvement from the user. Two large drawbacks to using stepwise regression include its tendency to optimize chance tendencies in the data and its relatively slow speed. Unfortunately, these drawbacks can become even more substantial as the number of observations and variables increases.

The relatively slow speed at which stepwise regression tends to proceed is easy to demonstrate using any large data set with a reasonably large number of input variables. Often there is limited time available for modeling. Time spent waiting for the regression to proceed would typically be better spent evaluating the input data by using a variety of variable selection methods, so as to minimize the variables left to consider before performing such a regression. These investigative methods often provide insights into the data that can be used in generating a better final model.

The larger concern is highlighted by Derksen and Keselman (1992), who investigated a variety of stepwise selection methods and stated that subset models selected through stepwise algorithms contain both authentic and noise variables. The authors later state that the average number of authentic variables found in the final subset models was always less than half the number of available authentic predictor variables. This is particularly bad news for both the predictive modeler who is interested in model performance as well as the business analyst who is seeking to interpret the results because they are in danger of making business decisions based on patterns that happen randomly.

The response to such findings is to consider whether or not to include each variable in the model based on expert opinion. Derksen and Keselman (1992) conclude that "the initial set of predictors should be selected carefully, including for study only those variables that according to theory/previous research are known/expected to be related to the response variable." The difficulty with this approach is twofold:

1. The number of variables to investigate makes careful investigation too time-consuming to perform.

2. The goal of data mining is to find new patterns in the data that might or might not be consistent with traditional thinking and/or historical findings.

Additionally, the large number of variables available in a typical data mining problem are likely to include a large number of noise variables.

A more recent paper suggests that the concerns raised by Derksen and Keselman might be overstated. Ambler, Brady, and Royston (2002) studied the performance of stepwise regression and found that standard variable selection can work well, even if there are a large number of irrelevant variables in the data. These findings seem to disagree with those of Derksen and Keselman. In the face of contradictory suggestions, it seems prudent to consider models built using stepwise methods cautiously.

To overcome this problem, the analyst must choose a strategy such that the model can be obtained within the requisite amount of time and is protected as much as possible from the tendency to select noise variables. Any approach that is suggested might need to be modified when the number of variables increases dramatically. Additionally, the severity of making a type I or type II error must likewise be weighed in assessing the inclusion of a certain variable in the model. Any strategy that is chosen should be weighed carefully by the analyst so that he or she can make an appropriate choice for the given situation.

INACCURATELY INTERPRETING THE RESULTS

Regression is a familiar concept to most people who rely on data analysis to make a decision, while other modeling methods such as decision trees and neural networks might be somewhat less familiar. This familiarity difference often results in a bias toward using a regression model because it is believed to be more interpretable and relatively safe due to its long history of use. Both of these conclusions can be misleading and can lead to misinterpretation and/or misapplication of the results.

Suppose your regression line is given by the equation

$$\hat{Y}_i = 2 * X_i$$

where

$\hat{Y}_i$ = the predicted value of the response for the *i*th observation

X_i = the value of the predictor for the *i*th observation

This implies that the predicted value of $\hat{Y}_i$ increases by two units for each unit increase in X_i. Unfortunately, this becomes more complicated when even one more predictor is added. Suppose your regression line is given by the equation

$$\hat{Y}_i = 2 * X_{1i} + 7 * X_{2i}$$

where

$\hat{Y}_i$ = the predicted value of the response for the *i*th observation

X_{1i} = the value of the first predictor for the *i*th observation

X_{2i} = the value of the second predictor for the *i*th observation

Observe that this equation can be rewritten as

$$\hat{Y}_i - 2 * X_{1i} = 7 * X_{2i}$$

implying there is a relationship between X_2 and the variability in Y that is not explained by X_1. If two variables are highly correlated, it is unlikely that both will be included in the model. In this situation, the model is not saying that one variable is important and the other is not. Instead, the model is merely stating that the rejected variable is not explaining a significant amount of variability over and above what the included variable explains. Certain combinations of variables in the final model might contain much of the information stored in a variable that is rejected.

In many cases, the modeler needs to assess how much each variable is contributing to the final model. This is a reasonable thing to assess as long as it is understood that any conclusions drawn are relative to a given set of predictors for a given data set. As interactions are brought into the data, this task and the associated task of interpretation become more difficult, because most people want to interpret the main effects, but interactions make this interpretation conditional on the levels of the variables in question. Additionally, a regression model is fairly inflexible when compared to decision tree and neural network models that handle nonlinearity much more naturally. The regression model requires a carefully specified model that defines the relationship between the input variables and response. Should this model be too inflexible to adequately represent the relationship for which it is used, the apparent interpretability of the model becomes meaningless because it does not reflect the true relationship between the inputs and the target variable.

To overcome this problem, consider all the things that affect the importance of a particular variable in a particular regression model. The importance of any variable in the model can be interpreted only in relationship to the subset of variables actually in the model. Before investing extensive resources into conclusions that seem to follow from the interpretation, take some time to investigate whether the relationships appear valid on other holdout data such as the test data set. It is also useful to evaluate the performance of other flexible models such as decision trees and neural networks to assess whether anything might be missing from the regression model. The interpretation obtained from the model is useful only if the model provides a sufficiently good representation of the true relationship. Be careful to investigate findings that don't make sense. These findings might be the source of additional insights into the relationship being modeled or insights into problems with the modeling process.

FITTING DECISION TREE MODELS

Initially, decision trees appear to have the best of all possible worlds. They model interactions automatically; they are capable of fitting complex models without a known structure in advance; they handle missing values without imputation; and they lend themselves easily to interpretation. These great features of decision trees can overshadow some difficulties that they present, including their inherent instability as well as their difficulty with modeling simple linear relationships.

IGNORING TREE INSTABILITY

The interpretation difficulty with regression problems is present with decision tree models as well because decision trees are highly dependent on the training data. When the input data is modified even slightly, the tree might generate a different initial split from the root node or might even choose a different variable to use for the initial split. Any difference at any given node affects all of the subsequent nodes, so the final tree might look very different even though the samples are virtually identical. However, the overall performance of the tree remains stable (Breiman et al. 1984).

To overcome this difficulty, be cautious in applying too much weight to conclusions drawn from a single tree. It might be useful to evaluate different samples of the training data to see the variability inherent in the tree model. It is critical to use a validation data set whenever possible to keep from overfitting on the training data, which leads to poorer results when the model is deployed. In situations where the relationships are somewhat vague, consider creating several trees and generate predicted values by pooling the predictions from the individual trees. Pooling the predictions makes the interpretation of the resulting model far more difficult but should reduce the inherent instability.

IGNORING TREE LIMITATIONS

Trees work by recursively partitioning the data set, enabling them to detect complex nonlinear or discontinuous relationships. Unfortunately, trees have far more difficulty fitting simple linear or smoothly changing relationships. Additionally, the tree provides less insight into the nature of this type of relationship. Consider how a tree would model a sample from a population where $y = x$ for values of x between 0 and 1. The tree might make an initial split at x=0.5 and generate a prediction of y = 0.75 when $x > 0.5$ and y = 0.25 when $x < 0.25$. Continued recursive splitting would result in an unnecessarily complicated model that describes a relationship that would have been much more simply described by a linear regression model.

To overcome this problem, always consider alternative modeling strategies that might provide a better fit or a simpler representation of the relationships in the data set. Relying solely on a tree can lead to inferior models in situations where little nonlinearity is present or where a smooth relationship more succinctly summarizes the relationship.

FITTING NEURAL NETWORK MODELS

Neural networks provide the ability to fit smooth nonlinear models without knowing the precise model structure in advance. Theoretically, a multilayer perceptron (MLP) with one hidden layer and a sufficient number of hidden units is a universal approximator (Ripley 1996), which means that an MLP can approximate any given surface within a prescribed error range. The flexibility and power of this method make it desirable, but there are problems that arise from their misuse, resulting from failure to do variable selection and/or failure to consider neural networks. Recall also that it is important to have an adequate number of observations for training and validation to protect from overfitting.

FAILING TO DO VARIABLE SELECTION

Neural networks fit a sequence of models much like decision tree and stepwise regression methods. Unlike decision trees and stepwise regression methods, each model in the sequence uses the same variables. Instead of modifying the variables or combinations of variables used in a particular model, a neural network model is obtained by updating the model coefficients for several iterations until a stopping rule is satisfied. Because the neural network is not performing variable selection, all of the original variables are now required in the scoring data set even if they contribute little or nothing to the fit. As a result, the scoring data set must be larger than necessary, making this task more memory and time intensive than it should be.

Additionally, neural network models produce many more coefficients than are fitted by a corresponding linear regression model. A simple linear regression model with two continuous predictors requires two slope parameters, one for each continuous input. However, a simple MLP with one hidden layer and three hidden units actually requires 13 parameter estimates. As the number of variables increases, the number of parameters required increases dramatically as well, which increases computation time.

To overcome these problems, it is important to perform variable selection before fitting a neural network model. Variable selection removes unnecessary variables, thereby greatly reducing the number of parameters that need to be estimated and computation time. The scoring data set now requires a smaller number of variables as well making this processing more efficient. It can be challenging to choose the correct variables for the model because many of the methods for identifying important variables are linear in nature. At times, it might be necessary to include certain variables in additional models to determine whether they will have a marked improvement on the fit. As discussed earlier, performing variable selection in a variety of ways helps ensure that important variables are included.

FAILING TO CONSIDER NEURAL NETWORKS

Neural networks are routinely ignored as a modeling tool because they are largely uninterpretable overall and are generally less familiar to analysts and business people alike. Neural networks can provide great diagnostic insights into the potential shortcomings of other modeling methods, and comparing the results of different models can help identify what is needed to improve model performance.

For example, consider a situation where the best tree model fits poorly, but the best neural network model and the best regression model perform similarly well on the validation data. Had the analyst not considered using a neural network, little performance would be lost by investigating only the regression model. Consider a similar situation where the best tree fits poorly and the best regression fits somewhat better, but the best neural network shows marked improvement over the regression model. The poor tree fit might indicate that the relationship between the predictors and the response changes smoothly. The improvement of the neural network over the regression indicates that the regression model is not capturing the complexity of the relationship between the predictors and the response. Without the neural network results, the regression model would be chosen and much interpretation would go into interpreting a model that inadequately describes the relationship. Even if the neural network is not a candidate to present to the final client or management team, the neural network can be highly diagnostic for other modeling approaches.

In another situation, the best tree model and the best neural network model might be performing well, but the regression model is performing somewhat poorly. In this case, the relative interpretability of the tree might lead to its selection, but the neural network fit confirms that the tree model adequately summarizes the relationship. In yet another scenario, the tree is performing very well relative to both the neural network and regression models. This scenario might imply that there are certain variables that behave unusually with respect to the response when a missing value is present. Because trees can handle missing values directly, they are able to differentiate between a missing value and a value that has been imputed for use in a regression or neural network model. In this case, it might make more sense to investigate missing value indicators rather than to look at increasing the flexibility of the regression model because the neural network shows that this improved flexibility does not improve the fit.

To overcome this problem, select variables judiciously and fit a neural network while ensuring that there is an adequate amount of data in the validation data set. As discussed earlier, performing variable selection in a variety of ways ensures that important variables are included. Evaluate the models fit by decision tree, regression, and neural network methods to better understand the relationships in the data, and use this information to identify ways to improve the overall fit.

COMPARING FITTED MODELS

Assessing the candidate models correctly usually requires an understanding of how the partitioned data sets are being used as well as business knowledge about how the results are to be used. Many metrics focus on overall model performance or on model performance at certain points in the data. While the assessment of models with a continuous target is fairly well understood, the evaluation of models with a categorical target is sometimes more difficult. As in the earlier discussions about categorical targets, this section focuses on binary targets.

MISINTERPRETING LIFT

It is common to report lift for a particular binary target, but these numbers can sometimes seem much more (or less) dramatic when the background rate is not taken into consideration. Lift is computed by comparing the proportion in the group of interest to the proportion in the overall population. Specifically,

$$Lift = \frac{Percentage\ with\ event\ in\ group}{Percentage\ with\ event\ overall}$$

Consider a target scenario where the target occurs 50% of the time. A model is developed such that 75% of the people in the top demidecile have the event of interest. The lift can be computed by dividing the percentage with the event of interest in the demidecile (75%) by percentage of the overall data set with the event of interest (50%) yielding a lift of 1.5. However, a different target scenario might have a target that occurs 1% of the time. A model is developed such that 8% of the people in the top demidecile have the event of interest, which corresponds to a lift of 8. The lift of 8 sounds far more dramatic than a lift of 1.5. However, the lift of 1.5 corresponds to a 25% increase in the predicted probability while the lift of 8 corresponds to an increase of only 7%. Additionally, the resulting probabilities in the top demidecile are over nine times greater (75% versus 8%) in the case where the lift is 1.5.

Now take into account that if the overall percentage rate is 50%, the largest possible lift (resulting from a 100% target event rate) is only 2. However, if the overall percentage is 2%, the largest possible lift is 50. These examples are meant only to illustrate that lift can be deceiving when the overall incidence rate isn't taken into account. To overcome this problem, ensure that the actual improvement and resulting probabilities are considered in addition to the lift.

CHOOSING THE WRONG ASSESSMENT STATISTIC

It is common to report assessment statistics for the top few deciles or demideciles. However, a company targeting the top 3% of the people might be better served focusing on performance in the top 3% of the model rather than on the top deciles or demideciles. Focusing on the overall model fit or on a portion of the population, which will be ignored, will likely lead to poorer performing models. Additionally, this strategy can make a useful model appear to be otherwise because it might perform well in the top few percent but more poorly elsewhere, leading to a lower estimate for the model's usefulness. The performance of the model is inherently tied to its predictive ability, and the predictions are tied very closely to the target profile discussed earlier. Ensuring that the decision rule implied by the target profile is consistent with the business decision should lead to models that perform better.

SCORING NEW DATA

Scoring is the ultimate goal of most predictive models. In many cases, the amount of data that the model scores is far greater than the amount of data on which the model was built. In situations where large amounts of data must be scored, mistakes are often made in generating inefficient score code.

GENERATING INEFFICIENT SCORE CODE

It is not uncommon in data mining to encounter many hundreds or even thousands of variables. Even a relatively small set of input variables can easily expand into a large number when there are values available at different points in time. For example, looking at a six-month history of 400 monthly scores quickly turns the 400 scores into 400 * 6 = 2,400 scores. In most situations, a relatively small number of variables are actually meaningful in the model. However, the scoring code might contain unnecessary code for many unnecessary variables in the form of data imputation or data transformations. This problem is particularly exacerbated when trees have been used for imputation. If a variable is unimportant in the model but was used to perform tree imputation, the variable might be needed for scoring. Because each tree imputation fits its own model, there are far more models being scored in the final code than just the predictive model of interest. Even if tree imputation is not used, the amount of code being generated for unimportant variables can slow down processing, particularly when the number of unimportant variables is excessive.

To overcome this problem, use tree imputation only for important variables where a logical imputed value is not available. For example, a missing value for a certain individual's donation to a previous solicitation is much more likely to be zero than any other value. In this case and many like it, tree imputation would likely provide an inferior result while generating a great deal of unnecessary code. To make the code as efficient as possible, perform data processing and variable selection first, and then rebuild the scoring code by using a data set with only the necessary variables so that the code is as efficient as possible.

IGNORING THE MODEL PERFORMANCE

Validation data sets are useful for limiting the overfitting that results from fitting candidate models to training data. Unfortunately, because performance on the validation data sets often has a direct role in model selection, a bias is introduced so that performance on the validation data set might be superior to the performance on the population. This impact can be more dramatic when the training and validation data sets have been oversampled. Specification of alternative priors can be used to adjust the predicted probability of response for oversampled data. However, the best measure of model performance is its performance on actual data.

In many common situations, a surrogate target variable is used so that the target being modeled is not perfectly correlated with the outcome of interest. Consider the example where a company creates a promotion for a particular product. In order for the promotion to be most effective, the company needs to target customers who are most likely to respond to a promotion for the product. In this situation, the company can build a *propensity model* to predict whether the customer currently owns the product or a *response model* to predict whether the customer will respond to the promotion. In either situation, the target variable is different from the actual target of interest.

In the propensity model, the difference is easy to see because customers who currently own the product are not necessarily those who would respond to a promotion for the product. In the response model, the difference is more subtle because the customers who responded to past offers might or might not have responded to the current promotion because it is likely to be different from the previous promotion. Even if the offer was the same, as time goes by it becomes more likely that the subgroup who would respond to the promotion has changed. Because response data is generally not available, a propensity model often represents the best approach, but performance on the scored data is likely to be worse than the performance on the training or validation data set because the target that was modeled was not the target of interest.

In this situation as in most modeling situations, it is important to monitor actual model performance. The actual performance on the population provides a baseline for how the model will perform until a true response model can be developed. As products go through life cycles, additional groups of people become interested in the product, and these additional groups might look very different from the groups that have previously responded.

Even if the product is of interest to some groups, it is not clear that all of these groups will respond to an offer for the product. Additionally, it is important to understand how many of the people in the group would have responded anyway. The only way to estimate these types of metrics is to withhold certain candidates to your offer from each decile to enable you to monitor performance on the targeted group as well as the control group. The difference in the response between these two groups can identify those people who are most heavily affected by your offer.

CLUSTERING YOUR DATA

Clustering is often performed as one of the early steps in investigating new data. The hope is that groups will emerge that enable more specific actions to be taken on specific subgroups in order to optimize some desired response. While clustering was originally designed to deal with multivariate (numeric) data, categorical data can be incorporated via a variety of techniques. Mistakes are often made in trying to fit a single cluster solution or in including too many categorical variables.

BUILDING ONE CLUSTER SOLUTION

Clustering does not have a right or wrong answer. Using different subsets of variables creates different clusters. Identifying the best cluster solution is a function of the business purpose for which that cluster solution was created. Predictive modeling provides a more appropriate way to create groups that are maximally separated with respect to some target outcome. *Clustering* is simply trying to find natural groupings based on a set of input variables and definition of distance.

In many cases, an analyst attempts to use every variable in clustering hoping to let the algorithm sort out the important and unimportant variables. The problem with this approach is that only a small proportion of the variables have a large impact over how the clusters form. Several additional variables have a much smaller impact, but this impact only serves to muddle the interpretation of how the clusters are formed by the primary clustering variables. Additionally, because clusters are generated by attempting to maximize distance between groups, the variables that drive the clustering are typically those with the greatest variability. While this sounds promising, it becomes challenging when these clusters represent very different types of information. The resulting cluster solutions become very hard to interpret, making it difficult to assess their usefulness by the necessary business criteria.

To overcome this problem, consider building a set of cluster solutions on meaningful and related subsets of your data. Because clusters differ for different sets of variables, using subsets of related variables provides a set of meaningful and interpretable clusters. The clusters built on these different subsets then begin to build a profile of the individual. For example, one cluster might focus on buying patterns while others focus on demographic or geographic data. Other clusters might focus on customer loyalty metrics or purchase assortments. The resulting clusters are far more easily interpreted, and different cluster solution(s) can be chosen for different business purposes.

INCLUDING (MANY) CATEGORICAL VARIABLES

As discussed earlier, clustering was originally designed for multivariate continuous data. Methods are available to incorporate categorical data by coding to create the notion of distance between levels. The first problem with this coding is that it generates variables that provide maximal separation (for example, suppose a binary variable gender is coded using 0 and 1 so that only two values exist with nothing between those two values), making these variables very desirable for clustering. The second problem is that these variables already provide groupings of the data, and replacing categorical data with a numeric representation for the purposes of creating categories doesn't make sense. In many cases, the categorical variables drive a great deal of the cluster formation, but the presence of the continuous variables serves to muddle the clusters that would be formed just by looking at specific combinations of categorical variables.

To overcome this problem, consider clustering only your numeric variables. If you are interested in looking at certain subsets of your data based on the categorical data, consider creating profiles built on different combinations of the grouping variables of interest, and then perform the clustering on the groups with a nontrivial number of members. This approach maximizes interpretability and makes it easier to evaluate different cluster solutions based on a particular business goal.

PERFORMING ASSOCIATION AND SEQUENCE ANALYSIS

Association and sequence analysis are descriptive techniques that seek to identify what combinations of items tend to appear in the same transaction (associations) or in what order these items tend to appear (sequences). Because every possible combination of every item with every other item must be considered, this relatively simple descriptive technique can be very time-consuming. Mistakes associated with these techniques primarily result from inadequate data preparation.

FAILING TO SORT THE DATA SET

To perform association or sequence analysis, a transactional data set is required. If the data table is not sorted by ID, then a sort is performed. Transactional data sets can be huge so sorting can require a fairly large amount of space leading to excessive processing times. To overcome this problem, sort the data set before running the association or sequence analysis if the processing time is unacceptable. Additionally, ensure that system options allow the use of all available memory for processing.

FAILING TO MANAGE THE NUMBER OF OUTCOMES

Before SAS Enterprise Miner 5, it was important to manage the number of items of interest because processing time increases far more rapidly than the number of possible items. In the current version, you can perform association and sequence analysis on up to 100,000 items. Experience suggests that when a large number of items are considered, the proportion of those items occurring a nontrivial number of times is often relatively small. While it is possible to analyze a large number of items, it takes longer to process this data than it would if you were to consider preparing the items as described in the section "Incorrectly Preparing or Failing to Prepare Categorical Predictors". Focusing on the subset of relatively frequent items greatly speeds up processing time.

CONCLUSION

This paper contains a collection of strategies designed to assist the analyst in identifying and overcoming common data mining mistakes. As mentioned earlier, it is possible that a particular method suggested here is not appropriate for a given analytical situation. However, these methods have been used effectively in a broad set of business situations. Correct application of these techniques generally decreases process time and improves the usefulness of any resulting model.

REFERENCES

Ambler, G., A. R. Brady, and P. Royston. 2002. "Simplifying a Prognostic Model: A Simulation Study Based on Clinical Data." *Statistics in Medicine*. 21:3803–3822.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and Regression Trees.* London: Chapman and Hall.

Derksen, S. and H. J. Keselman. 1992. “Backward, Forward, and Stepwise Automated Subset Selection Algorithms: Frequency of Obtaining Authentic and Noise Variables.” *British Journal of Mathematics and Statistical Psychology* 45:265–282.

Muller, K. E. and V. A. Benignus. 1992. “Increasing Scientific Power with Statistical Power.” *Neurotoxicology and Teratology*. 14:211–219.

Ripley, B. D. 1996. *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.

ACKNOWLEDGMENTS

I would like to thank Ross Bettinger, Jay King, Bob Lucas, Craig DeVault, Jim Georges, and Annette Sanders for reviewing this material and providing valuable insights. I especially acknowledge the contributions of Ross Bettinger who reviewed several drafts of this document. Finally, I thank Will J. E. Potts who documented much of the decision theory in the target profile section.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Doug Wielenga
100 SAS Campus Drive
Cary, NC 27513
E-mail: Doug.Wielenga@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.