Chapter 12: Using Prepared Data

Overview

So what's the benefit of using prepared data to build models? You get more effective models faster. Most of the book so far has described the role of data preparation and how to properly prepare data for modeling. This chapter takes a brief look at the effects and benefits of using prepared data for modeling.

Actually, to examine the preparation results a little more closely, quite often the prepared data models are both better and produced faster. However, sometimes the models are of better quality, but produced in the same time as with unprepared data. Sometimes they are no better but produced a lot faster. For a given data set, both the quality of the model and the speed of modeling depend on the algorithm used and the particular implementation of that algorithm. However, it is almost invariably the case that when data is prepared in the manner described in this book, using the prepared data results in either a better model, a faster model, or a better model faster than when using unprepared data—or than when using data inadequately or improperly prepared. Can this statement be justified?

One credit card company spent more than three months building a predictive model for an acquisition program that generated a response rate of 0.9% over a 2,000,000-piece mailing. Shortly thereafter, using essentially the same modeling tools and data, the company launched another campaign using a model constructed from prepared data. Response rate in this model was 1.23%—a more than 36% response improvement. Was this particularly significant to the credit card company? That 36% improvement comes straight off the cost of acquisition. In this case, data preparation translated into 6,522 more customers. Another way of looking at it is that in the first program, the company paid about $138 to acquire a new customer. Data preparation reduced this cost to about $101 per acquisition.

A large commercial baker servicing nearly 100 stores sought to increase profitability by focusing attention on under- or oversupply of products, that is, not having enough, or having too many, bagels, croissants, and other “morning goods” on the shelves. (Shelf life is the day of manufacture only.) Insufficient products mean lost sales; too many products means waste since they have to be discarded. The baker used both statistical models and data mining to estimate demand and appropriate inventory levels. Rebuilding existing models with prepared data produced models that were better than 8% more accurate than the previous models. This immediately translated into saving about $25 per store per day in shortage and overage losses. What seems like a tiny saving per store comes to nearly $1 million saved over all the stores in a year—and that was before improved models were built.

A financial institution modeled trading data over many markets and combinations of markets, all of which produced different types of data. Each required from two to four weeks to build an optimized model of each market. Each successful model contributed to trading success during the life of the model, which could be quite short (from days to weeks after deployment). By using automated data preparation techniques they produced models of equivalent quality in hours or days, thus getting more models of more markets deployed more quickly.

An industrial controls manufacturer developed a printing press color process optimizer. When a multicolor press starts its print run, and until the press is at operating temperature and speed (which can take a lot of time), the press is producing such inferior-quality print that it is waste. Paper and ink are very expensive, and reducing the waste is a high priority (and gives a high payback). The manufacturer's control measured performance characteristics and implemented a model that controlled the ink flow rates, paper feed rate, and manipulated other press controls. The automated model, when built conventionally, cut the run-up time to half of what it was without the control. Using prepared data to build the model produced an improvement that reduced the waste run time by an additional 10%.

Data preparation techniques as described in this book have been used to improve modeling results in traffic pattern analysis in a major U.S. city, molecular structure analysis in the research department of a major pharmaceutical company, and in feature detection for medical image processing. They have been used to enhance severe weather detection in meteorological data and to reduce consumable product returns in the confectionery industry. In every case, data preparation alone—that is, just the application of the techniques described in this book—has yielded from small to moderate but in all cases a real and significant improvement. In one stunning case (see Chapter 10), a brokerage data set that was essentially useless before preparation was very profitably modeled after preparation.

Preparing data for mining uses techniques that produce better models faster. But the bottom line—the whole reason for using the best data preparation available—is that data preparation adds value to the business objective that the miner is addressing.

12.1 Modeling Data

Before examining the effect the data preparation techniques discussed in this book have on modeling, a quick look at how modeling tools operate to extract a model from data will provide a frame of reference.

12.1.1 Assumptions

Modeling of any data set is based on five key assumptions. They are worth reviewing since if any of them do not hold, no model will reflect the real world, except by luck! The key assumptions are

1. Measurements of features of the world represent something real about the world.

2. Some persistent relationship exists between the features measured and the measurements taken.

3. Relationships between real-world features are reflected as relationships between measurements.

4. Understanding relationships between measurements can be applied to understanding relationships between real-world features.

5. Understanding relationships between real-world features can be used to influence events.

In other words, data reflects and connects to the world so that understanding data and its relationships contributes to an understanding of the world. When building a model, the modeler must ask if, in this particular case, these assumptions hold. (In several places throughout the book, there are examples of failed models. In every case, if the modeling itself was valid, the failure was that the modeler failed to check one or all of these five assumptions.)

12.1.2 Models

It's fine to say that a modeler builds a model, but what actually is a model? A *model*, in a general sense, is a replica of some other object that duplicates selected features of that larger object, but in a more convenient form. A plastic model World War II battleship, for instance, models the external appearance of the original to some reduced scale and is far more convenient for displaying in a living room than the original! Small-scale aircraft, made from a material that is much too heavy to allow them to fly, are useful for studying airflow around the aircraft in a wind tunnel. In hydrographic research, model ships sail model seas, through model waves propelled by model winds.

In some way, all models replicate some useful features of the original so that those features themselves, singled out from all other features, can be studied and manipulated. The nonphysical models that the data miner deals with are still models in the sense that they reflect useful features of the original objects in some more-convenient-to-manipulate way. These are symbolic models in which the various features are represented by symbols. Each symbol has a specific set of rules that indicate how the symbol can be manipulated with reference to other symbols. The symbolic manipulators used today are usually digital computers. The symbols consist of a mixture of mathematical and procedural structures that describe the relationships between, and operations permitted on, the symbols that comprise the model itself.

Typically, data miners (and engineers, mathematicians, economists, and statisticians, too) construct models by using symbols and rules for manipulating those symbols. These models are active creations in the sense that they can be computationally manipulated to answer questions posed about the model's behavior—and thus, by extension, about the behavior of the real world.

But data miners and statisticians differ from economists, engineers, and scientists in the way that they construct their models. And indeed, statisticians and data miners differ from each other, too. Engineers, scientists, and economists tend to form theories about the behavior of objects in the world, and then use the language of symbols to express their appreciation of the interrelationships that they propose. Manipulating the model of the proposed behavior allows them to determine how well (or badly) the proposed explanation "works." So far as modeling goes, data miners and statisticians tend to start with fewer preconceived notions of how the world works, but to start instead with data and ask what phenomenon or phenomena the data might describe. However, even then, statisticians and data miners still have different philosophical approaches to modeling from data.

12.1.3 Data Mining vs. Exploratory Data Analysis

Exploratory data analysis (EDA) is a statistical practice that involves using a wide variety of single-variable and multivariable analysis techniques to search for the underlying systemic relationships between variables. Data mining shares much of the methodology and structure of EDA, including some EDA techniques. But EDA focuses on discovering the basic nature of the underlying phenomena; data mining focuses tightly on the practical application of results. This difference, while it may seem narrow, is actually very broad.

Data miners are perfectly happy to accept a proven working "black box" that is known to work under a specified range of conditions and circumstances. The essential nature of the underlying phenomenon is not of particular interest to a miner, so long as the results are reliable, robust, and applicable in the real world to solve an identified problem. Statisticians, using EDA, want to be able to "explain" the data (and the world). How does this difference in approach turn out in practice?

A major credit card issuer had a large staff of statistical modelers. Their job was to investigate data and build models to discover "interesting" interactions in the data. Their approach was to take statistically meaningful samples from the credit card issuer's voluminous data set and look for proposed interesting interactions between the variables in the data set. Some of the proposed interactions were found to be statistically justified, and were further proposed for possible marketing action. Other interactions were found to be statistically insignificant, and no further action was taken on these.

The models produced by the statisticians were working well, but the vice president of marketing wanted to try data mining, a new discipline to her, but one that had received good reviews in the business press. How did the data mining project differ from the statistical approach?

Data mining did not start with a sample of the data. It also did not start with proposed interactions that might or might not be supported by the data. Instead miners started with the business problem by identifying areas that the marketing VP defined as strategically important for the company.

Next, miners selected data from the credit card company's resources that seemed most likely to address issues of strategic importance, guided in this by the marketers as domain experts. The mining tools surveyed all of the selected data available, which amounted to many gigabytes, including many hundreds of variables. One data set was built from credit card transaction records that were fully assayed, reverse pivoted, and transformed into consumer activity by account. A data survey revealed many possibly interesting areas for exploration.

Instead of approaching the data with ideas to test, the miners used the data survey to extract from the data a moderately comprehensive set of all the interactions that the data could possibly support. Many of these interactions were, naturally, noise-level interactions, only fit to be discarded. However, a key point here is that the miners asked the data for a comprehensive list of interactions that could possibly be supported by that data. The statisticians asked the data what level of support was available for one, or a few, predefined ideas that they had developed.

With a list of candidate interactions generated, the miners reviewed the list with the domain experts (marketers) to determine which might, if they proved successful, be the best and most effective possibilities.

Given the reverse pivot, both the statistical staff and the data miners attacked the data set with their respective tools. The statisticians cut off their samples and identified broad market segments that differentiated credit card usage. The data miners set their tools to mining all the data, extracting both broad and narrow fluctuations. The main search criteria for the data miners was to find the "drivers" for particularly profitable groups and subgroups in the data (inferential modeling); they then had to design models that would predict who those people were (predictive modeling).

The statistical analysts built regression models mainly, carefully examining the "beta weights" for linear and nonlinear regressions; analyzing residuals, confidence intervals, and many other items in the models; and translating them back into English to explain what was happening in the data. Because they were working with samples, they were able to build only high-level models of general trends.

The data miners used several techniques, the three principal ones being rule induction, decision trees, and neural networks. The decision tree proved to be the most useful on this occasion. It has the ability, with or without guidance, to automatically find the gross structure in the data and extract increasingly finer structure as more and more data is examined. In this case, several very valuable insights were waiting in the fine structure.

As is so often the case, one insight was fairly obvious with hindsight. It turned out that 0.1% of the accounts showed a pattern of exceptionally high profitability. This subsegment was identified as about 30% of all people buying ski equipment valued at $3000 in a 30-day period. These people also went on to buy travel packages, presumably to ski resorts, well in excess of an additional $3000. High profitability indeed.

The insight was used to build a marketing offer rolling ski equipment purchases, travel packages, and other benefits and services into a discounted package that was triggered by any purchase of ski equipment of appropriate value. A separate program was also built on the same insight: a “lifestyle” credit card offering the cardholder permanent packages and point accumulation toward ski vacations.

When appropriately targeted, this structured offer produced spectacular returns. The figures look like this: the identified segment was 0.1% of 2.5 million people’s accounts generated from the reverse pivot, or about 2500 people. These 2500 people were about 30% of the subpopulation, so the whole subpopulation was about 8300 people. Roughly 40% of the remaining subpopulation responded to the marketing offer. Forty percent of 8300 people—that is, 3300 additional people—purchased travel packages worth $3000 or more. These 3300 customers, by increasing their indebtedness by $3000 each, produced an increase in loan inventory of about $10 million!

A significant point: in the example, a fluctuation of about 0.1% would almost certainly be missed by statistical sampling techniques. This figure is technically “statistically insignificant” when considered as a fluctuation in a population. But it is far from insignificant in a commercial sense. Thus it is that the philosophical approach taken by EDA and data miners is quite different.

12.2 Characterizing Data

Before looking at the effect of data preparation on modeling, we need to look at what modeling itself actually does with and to data. So far, modeling was described as finding relationships in data, and making predictions about unknown outcomes from data describing the situation at hand. The fundamental methods used by data miners are easily outlined, although the actual implementation of these simple outlines may require powerful, sophisticated, and complex algorithms to make them work effectively in practice.

The general idea of most data mining algorithms is to separate the instance values in state space into areas that share features in common. What does this look like? Figure

12.1 shows a sample two-dimensional state space containing a few instance values. Simply looking at the illustrated state space seems to reveal patterns. This is hardly surprising, as one of the most formidably powerful pattern recognition devices known is the human brain. So powerful is the brain that it tends to find patterns even where objectively none are known to exist. Thus, just looking at the illustration of state space seems to reveal groups of points representing the instance values in the state space. A modeling tool has the task of separating and grouping these points in meaningful ways. Each modeling algorithm takes a slightly different approach. With a pencil, or in some other way, you could quite easily mark what seem to be “natural” groups, clumps, or clusters. But how do data mining algorithms, which cannot look at state space, go about finding the associations and related points?

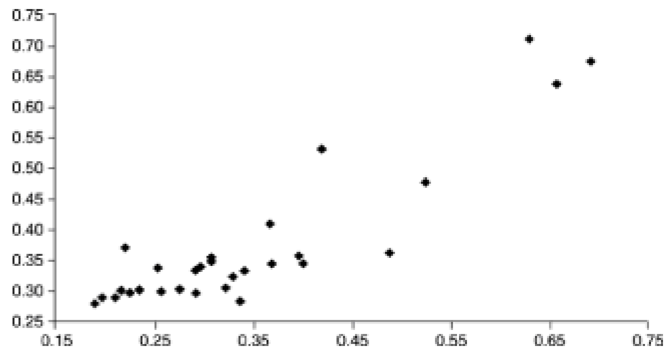


Figure 12.1 Instance values filling part of state space. Some patterns may be intuitively evident in the way the instance values fill the space. The axes show the range of values for each input variable.

12.2.1 Decision Trees

Decision trees partition state space such that the points in each separated partition have the maximum difference in some selected feature. Partitioning proceeds by selecting a partitioning point in the values of a single variable. Each of the partitions is then separately partitioned on the most appropriate partitioning variable and point for that partition. It continues until some stopping criterion is reached, or until each partition contains only a single point, or when it is no longer possible to separate the points.

The partition criteria can be expressed in the form of rules. In Figure 12.2, the partition shown in the upper-right partition is covered (explained) by a rule like “IF X > 0.5 AND Y > 0.51 THEN . . . .” The figure shows that the result of the partitioning is to create boundaries that are parallel to the dimensions of the space. Each axis is treated separately, resulting in a mosaic of boxes, each including some part of state space. Each partition covers some part of the whole space, and the whole of the space is covered by some partition. This comprehensive coverage is an important point.

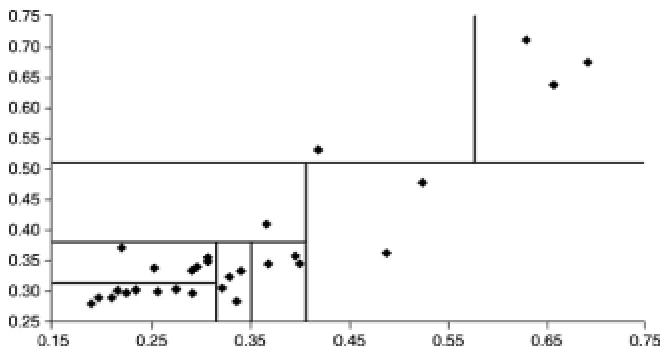


Figure 12.2 State space divided by a decision tree.

12.2.2 Clusters

Clustering also partitions state space, separating areas that have points sharing a common feature. This sounds similar to decision trees; however, there are marked differences. There are many different methods of clustering, but the end result of a clustering algorithm can be thought of as producing “clouds” in state space. The clouds, as shown in Figure 12.3, have marked differences from decision trees. One major difference is that they don’t have linear boundaries parallel to the axes! The nonlinearity makes it very difficult to extract seemingly simple rules from clusters. However, expressing what clusters have in common—their ranges on various axes—can produce meaningful statements about the clusters.

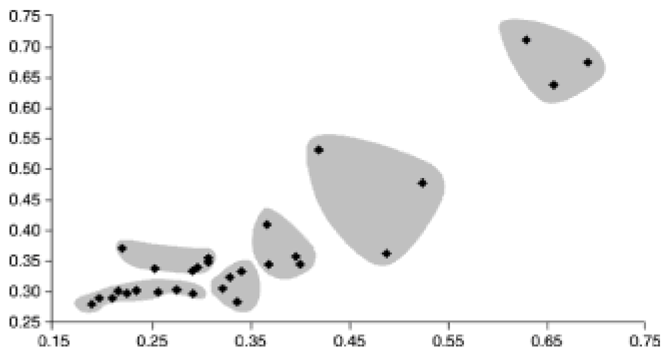


Figure 12.3 A clustering algorithm identifies groups, or clusters, of points that are associated in some way. The cluster areas cover all of the points in space, but do not necessarily cover the whole of the state space.

Another difference is that quite clearly the clustering algorithm used to produce the clusters shown does not cover the whole of state space. If the model is applied to a prediction data set that happens to have instance values defining a point outside a cluster boundary, the best that can be done is to estimate the nearest cluster. With such methods of clustering, the area outside the defined clusters is not itself a cluster, but more like undefined space. Other clustering methods, such as sequential natural clustering (briefly mentioned in Chapter 11), do produce clusters that have some cluster membership for every point in state space. In all cases, however, the clusters have irregular boundaries.

12.2.3 Nearest Neighbor

The way that nearest neighbors are used to describe interactions in state space is described in detail in Chapter 6. Very briefly, nearest-neighbor methods select some specific number of neighbors, and for every point use that number of nearest neighbors. Some averaging of the state of the neighbors for that point represents the point whose features are to be inferred or predicted. Figure 12.4 illustrates how neighbors might be selected. The figure illustrates the use of four nearest neighbors in any direction. For each of the selected points L, M, and N, the four closest points are used to estimate the characteristics of the selected point. Points L and M are characterized by four asymmetrically distributed points, which may not actually be the most representative points to choose.

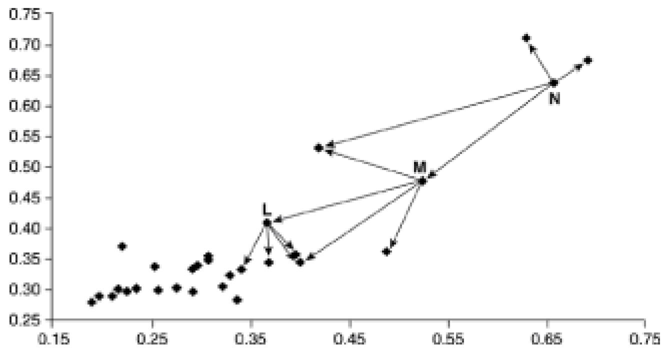


Figure 12.4 Principles of nearest-neighbor methods.

Nearest-neighbor methods have strengths and weaknesses. In the figure, some of the points use relatively distant nearest neighbors while others use closer neighbors. Another possible problem occurs when the nearest neighbors are not representatively distributed—points L and M in the figure show neighbors asymmetrically distributed. There are various methods for tuning nearest-neighbor algorithms to optimize for particular requirements, some of which are mentioned in Chapter 6.

12.2.4 Neural Networks and Regression

Figure 12.5 shows how manifold fitting methods characterize data. Neural networks, and how they characterize data, are discussed in Chapter 10. Linear regression (a basic statistical technique) fits a stiff, flat manifold to the data in some “best fit” way. In two dimensions a flat, stiff manifold is a straight line.

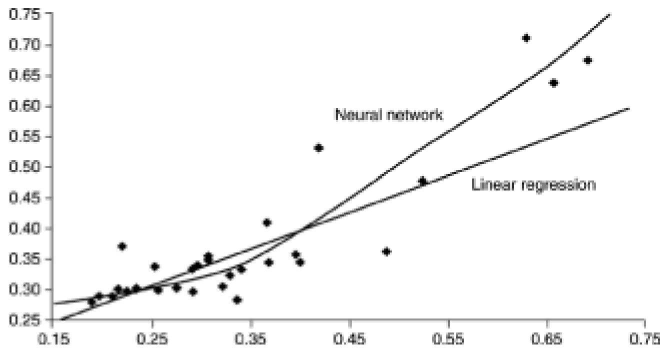


Figure 12.5 Fitting manifolds—either inflexible (linear regression) or flexible (neural network)—to the sample data results in a manifold that in some sense "best fits" the data.

These methods work by creating a mathematical expression that characterizes the state of the fitted line at any point along the line. Studying the nature of the manifold leads to inferences about the data. When predicting values for some particular point, linear regression uses the closest point on the manifold to the particular point to be predicted. The characteristics (value of the feature to predict) of the nearby point on the manifold are used as the desired prediction.

12.3 Prepared Data and Modeling Algorithms

These capsule descriptions review how some of the main modeling algorithms deal with data. The exact problems that working with unprepared data presents for modeling tools will not be reiterated here as they are covered extensively in almost every chapter in this book. The small, example data set has no missing values—if it had, they could not have been plotted. But how does data preparation change the nature of the data?

The whole idea, of course, is to give the modeling tools as easy a time as possible when working with the data. When the data is easy to model, better models come out faster, which is the technical purpose of data preparation. How does data preparation make the data easier to work with? Essentially, data preparation removes many of the problems. This brief look is not intended to catalog all of the features and benefits of correct data preparation, but to give a feel for how it affects modeling.

Consider the neural network—for example, as shown in Figure 12.5—fitting a flexible manifold to data. One of the problems is that the data points are closer together (higher density) in the lower-left part of illustrated state space, and far less dense in the upper right. Not only must a curve be fitted, but the flexibility of the manifold needs to be different in each part of the space. Or again, clustering has to fit cluster boundaries through the higher density, possibly being forced by proximity and the stiffness of the boundary, to include points that should otherwise be excluded. Or again, in the nearest-neighbor methods, neighborhoods were unbalanced.

How does preparation help? Figure 12.6 shows the data range normalized in state space on the left. The data with both range and distribution normalized is shown on the right. The range-normalized and redistributed space is a "toy" representation of what full data preparation accomplishes. This data is much easier to characterize—manifolds are more easily fitted, cluster boundaries are more easily found, neighbors are more neighborly. The data is simply easier to access and work with. But what real difference does it make?

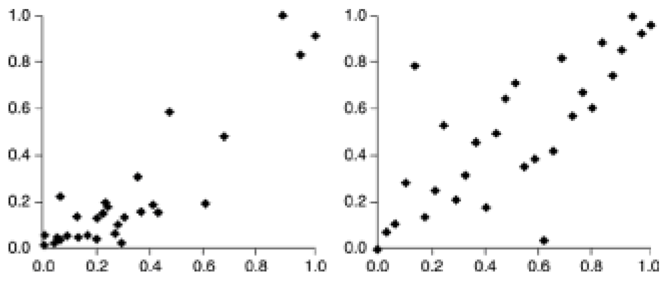


Figure 12.6 Some of the effects of data preparation: normalization of data range (left), and normalization and redistribution of data set (right).

12.3.1 Neural Networks and the CREDIT Data Set

The CREDIT data set is a derived extract from a real-world data set. Full data preparation and surveying enable the miner to build reasonable models—reasonable in terms of addressing the business objective. But what does data preparation alone achieve in this data set? In order to demonstrate that, we will look at two models of the data—one on prepared data, and the other on unprepared data.

Any difficulty in showing the effect of preparation alone is due to the fact that with ingenuity, much better models can be built with the prepared data in many circumstances than with the data unprepared. All this demonstrates, however, is the ingenuity of the miner! To try to "level the playing field," as it were, for this example the neural network models will use all of the inputs, have the same number of nodes in the hidden layer, and will use no extracted features. There is no change in network architecture for the prepared and unprepared data sets. Thus, this uses no knowledge gleaned from the either the data assay or the data survey. Much, if not most, of the useful information discovered about the data set, and how to build better models, is simply discarded so that the effect of the automated techniques is most easily seen. The difference between the "unprepared" and "prepared" data sets is, as nearly as can be, only that provided by the automated preparation—accomplished by the demonstration code.

Now, it is true that a neural network cannot take the data from the CREDIT data set in its raw form, so some preparation must be done. Strictly speaking, then, there is no such thing—for a neural network—as modeling unprepared data. What then is a fair preparation method to compare with the method outlined in this book?

StatSoft is a leading maker of statistical analysis software. Their tools reflect statistical state-of-the art techniques. In addition to a statistical analysis package, StatSoft makes a neural network tool that uses statistical techniques to prepare data for the neural network. Their data preparation is automated and invisible to the modeler using their neural network package. So the "unprepared" data in this comparison is actually prepared by the statistical preparation techniques implemented by StatSoft. The "prepared" data set is prepared using the techniques discussed in this book. Naturally, a miner using all of the knowledge and insights gleaned from the data using the techniques described in the preceding chapters should—using either preparation technique—be able to make a far better model than that produced by this na‹ve approach. The object is to attempt a direct fair comparison to see the value of the automated data preparation techniques described here, if any.

As shown in Figure 12.7, the neural network architecture selected takes all of the inputs, passes them to six nodes in the hidden layer, and has one output to predict—BUYER. Both networks were trained for 250 epochs. Because this is a neural network, the data set was balanced to be a 50/50 mix of buyers and nonbuyers.

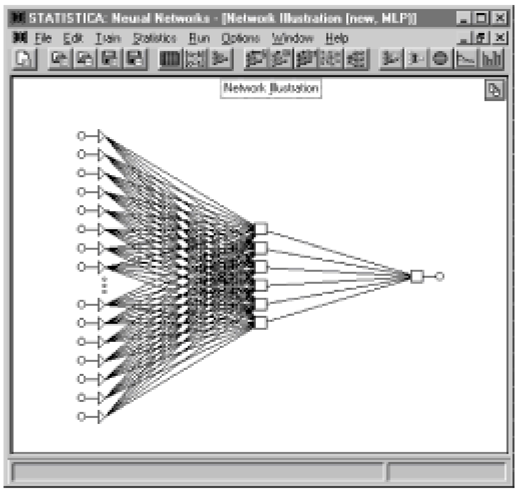
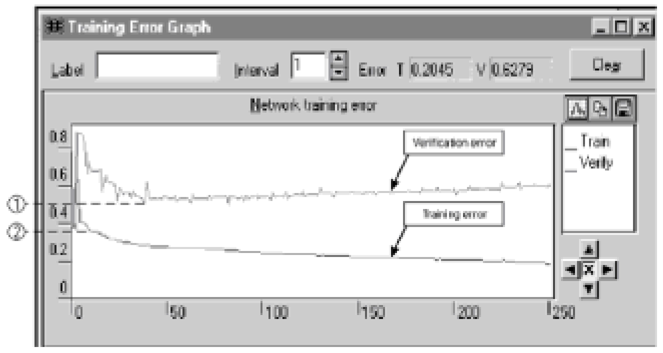

STATISTICA: Neural Networks - [Network Illustration (new, MLP)]
File Edit Train Statistics Run Options Window Help
Network Illustration

Figure 12.7 Architecture of the neural network used in modeling both the prepared and unprepared versions of the CREDIT data set predicting BUYER. It is an all-input, six-hidden-node, one-output, standard back-propagation neural network.

Figure 12.8 shows the result of training on the unprepared data. The figure shows a number of interesting features. To facilitate training, the instances were separated into training and verification (test) data sets. The network was trained on the training data set, and errors in both the training and verification data sets are shown in the “Training Error Graph” window. This graph shows the prediction errors made in the training set on which the network learned, and also shows the prediction errors made in the verification data set, which the network was not looking at, except to make this prediction. The lower, fairly smooth line is the training set error, while the upper jagged line shows the verification set error.



Interval 1 Error T 0.2045 V 0.6279

Figure 12.8 Errors in the training and verification data sets for 250 epochs of training on the unprepared CREDIT data set predicting BUYER. Before the network has learned anything, the error in the verification set is near its lowest at 2, while the error in the training set is at its highest. After about 45 epochs of training, the error in the training set is low and the error in the verification set is at its lowest—about 50% error—at 1.

As the training set was better learned, so the error rate in the training set declined. At first, the underlying relationship was truly being learned, so the error rate in the verification data set declined too. At some point, overtraining began, and the error in the training data set continued to decline but the error in the verification data set increased. At that point, the network was learning noise.

In this particular example, in the very early epochs—long before the network actually learned anything—the lowest error rate in the verification data set was discovered! This is happenstance due to the random nature of the network weights. At the same time, the error rate in the training set was at its highest, so nothing of value was learned by then. Looking at the graph shows that as learning continued, after some initial jumping about, the relationship in the verification data set was at its lowest after about 45 epochs. The error rate at that point was about 0.5. This is really a very poor performance, since 50% is exactly the same as random guessing! Recall that the balanced data set has 50% of buyers and nonbuyers, so flipping a fair coin provides a 50% accuracy rate. It is also notable that the error rate in the training data set continued to fall so that the network continued to learn noise. So much then for training on the “unprepared” data set.

The story shown for the prepared data set in Figure 12.9 is very different! Notice that the highest error level shown on the error graph here is about 0.55, or 55%. In the previous figure, the highest error shown was about 90%. (The StatSoft window scales automatically to accommodate the range of the graph.) In this graph, three things are very notable. First, the training and verification errors declined together at first, and are by no means as far separated as they were before. Second, error in the verification declined for more epochs than before, so learning of the underlying relationship continued longer. Third, the prediction error in the verification data set fell much lower than in the unprepared data set. After about 95 epochs, the verification error fell to 0.38, or a 38% error rate. In other words, with a 38% error rate, the network made a correct prediction 62% of the time, far better than random guessing!

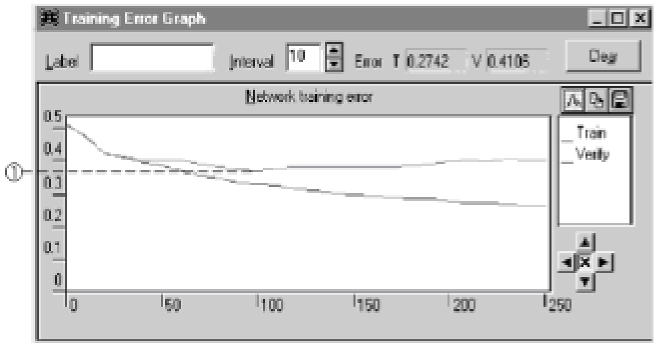


Figure 12.9 Training errors in the prepared data set for identical conditions as before. Minimum error is shown at 1.

Using the same network, on the same data set, and training under the same conditions, data prepared using the techniques described here performed 25% better than either random guessing or a network trained on data prepared using the StatSoft-provided, statistically based preparation techniques. A very considerable improvement!

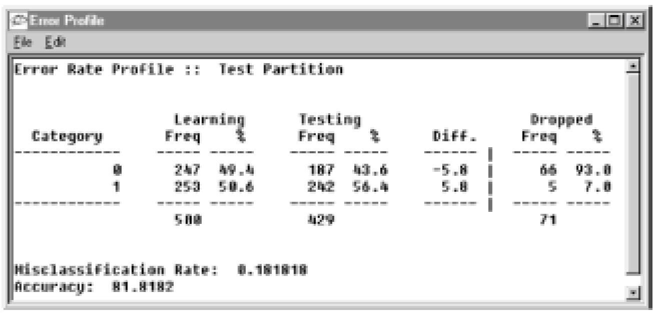
Also of note in comparing the performance of the two data sets is that the training set error in the prepared data did not fall as low as in the unprepared data. In fact, from the slope and level of the training set error graphs, it is easy to see that the network training in the prepared data resisted learning noise to a greater degree than in the unprepared data set.

12.3.2 Decision Trees and the CREDIT Data Set

Exposing the information content seems to be effective for a neural network. But a decision tree uses a very different algorithm. It not only slices state space, rather than fitting a function, but it also handles the data in a very different way. A tree can digest unprepared data, and also is not as sensitive to balancing of the data set as a network. Does data preparation help improve performance for a decision tree? Once again, rather than extracting features or using any insights gleaned from the data survey, and taking the CREDIT data set as it comes, how does a decision tree perform?

Two trees were built on the CREDIT data set, one on prepared data, and one on unprepared data. The tree used was KnowledgeSEEKER from Angoss Software. All of the defaults were used in both trees, and no attempt was made to optimize either the model or the data. In both cases the trees were constructed automatically. Results?

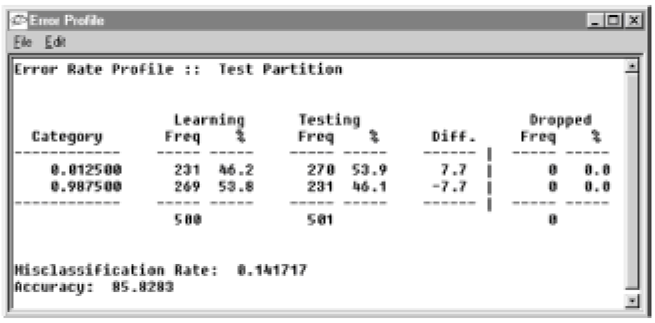
The data was again divided into training and test partitions, and again BUYER was the prediction variable. The trees were built on the training partitions and tested on the test partitions. Figure 12.10 shows the results. The upper image shows the Error Profile window from KnowledgeSEEKER for the unprepared data set. In this case the accuracy of the model built on unprepared data is 81.8182%. With prepared data the accuracy rises to 85.8283%. This represents approximately a 5% improvement in accuracy. However, the misclassification rate improves from 0.181818 to 0.141717, which is an improvement of better than 20%. For decision trees, at least in this case, the quality of the model produced improves simply by preparing the data so that the information content is best exposed.



```
Error Rate Profile ::  Test Partition


                    Learning        Testing                   Dropped
  Category         Freq     %      Freq     %        Diff.     Freq     %
-------------     ------ ------   ------ ------    ------- |  ------ ------
            0        247   49.4      187   43.6       -5.8 |      66   93.0
            1        253   50.6      242   56.4        5.8 |       5    7.0
-------------     ------ ------   ------ ------    ------- |  ------ ------
                     500             429                          71


Misclassification Rate:  0.181818
Accuracy:  81.8182
```



```
Error Rate Profile ::  Test Partition


                    Learning        Testing                   Dropped
  Category         Freq     %      Freq     %        Diff.     Freq     %
-------------     ------ ------   ------ ------    ------- |  ------ ------
     0.012500        231   46.2      270   53.9        7.7 |       0    0.0
     0.987500        269   53.8      231   46.1       -7.7 |       0    0.0
-------------     ------ ------   ------ ------    ------- |  ------ ------
                     500             501                           0


Misclassification Rate:  0.141717
Accuracy:  85.8283
```

Figure 12.10 Training a tree with Angoss KnowledgeSEEKER on unprepared data shows an 81.8182% accuracy on the test data set (top) and an 85.8283% accuracy in the test data for the prepared data set (bottom).

12.4 Practical Use of Data Preparation and Prepared Data

How does a miner use data preparation in practice? There are three separate issues to address. The first part of data preparation is the assay, described in Chapter 4. Assaying