Paper 501-2013

Finding the Gold in Your Data: An Overview of Data Mining

David A. Dickey, NC State University, Raleigh NC

1. INTRODUCTION

We hear a lot about data mining these days but what exactly is it? On the analysis side, it consists of tools for making decisions under uncertainty, thus sharing a lot with the field of statistics. The techniques, however, must be applicable to very large data sets and certainly the computer science side of data mining is critical. Furthermore it is often the case that we are looking for interesting, unusual, and/or informative subsets of these very large data sets, "nuggets" so to speak. For example we might want to identify loan applicants who are likely to default, groups of subjects who are likely to have heart problems, or groups of alumni that are likely to donate substantial amounts to their alma maters. The data sets for which the methods are targeted often result from observational studies or even data collection efforts with no particular goal in mind. Personally, I define data mining as a collection of tools for accomplishing these purposes so that the tools themselves provide the definition. My intent is to present some of the most common of these as an introduction to the topic.

2. DECISION TREES

One of the most used tools in data mining is recursive splitting, more commonly known as the construction of decision trees or as "CART," an acronym for Classification And Regression Trees. Suppose, as an example (Figure 1 A), we start with a rectangle of data in which the vertical axis represents debt to income ratio and the horizontal represents age. Within the rectangle are red and green points showing the age and debt to income ratio for defaulters and nondefaulters respectively. A horizontal line divides the rectangle into two smaller rectangles, the top having a higher proportion of defaulters than the lower. To its right, these two rectangles have again been split, this time based on age. The final split, at the bottom, consists of 5 rectangles such that the proportions of defaulters are as different as possible from rectangle to rectangle within certain restrictions.

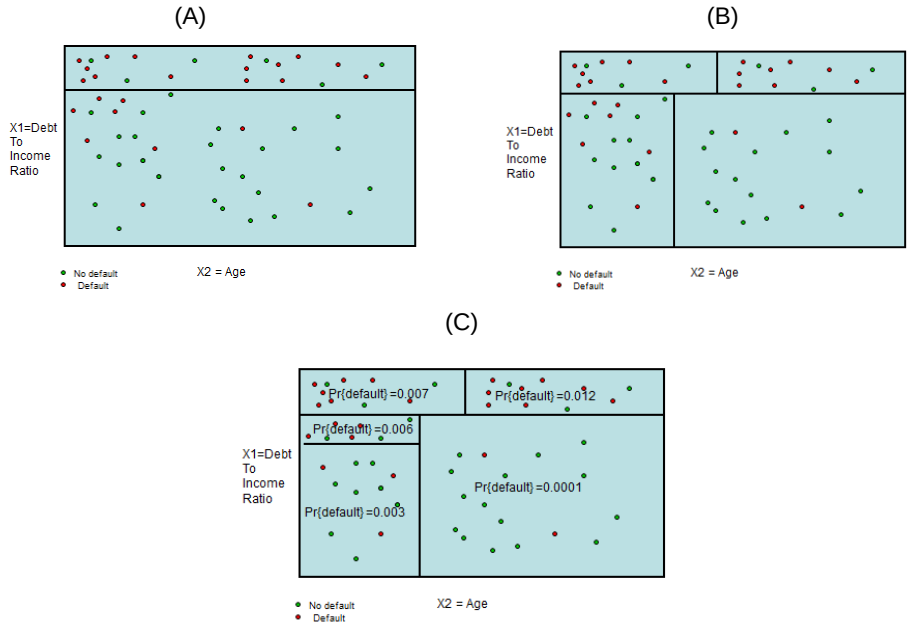


Figure 1: A, B, and C: Tree construction.

Figure 1 and its description beg several questions. For one thing, how does one pick the variable on which to split and at what value is that split made? Secondly, with enough tiny rectangles we could have each with only defaulters or only non-defaulters. This is a perfectly fitting but useless tree. What are the "certain restrictions" that stop us from going to that extreme? Note that each of the 5 rectangles is labeled with probabilities of default. Clearly that is not the proportion of defaulters within the rectangle so how is it computed?

2.a SPLITTING RULES

One common method of choosing a splitting variable for a binary response and split point uses a 2x2 Chi-square contingency table. The two cells in the top row have counts of defaulters (left cell) and non-defaulters (right cell) for high debt to income cases and the bottom two cells have corresponding counts for low debt to income ratios.

	Defaulters	Non -Defaulters
High ratio		
Low ratio		

A table such as this is constructed for every possible division point between high and low ratios, that is, there is a table for every possible definition of high debt to income ratio. The split point with the lowest p-value for the Chi-square test is the chosen division point for debt to income ratio. A similar sequence of Chi-square tables and tests is done for the age variable and a winning split value is declared. We now need to choose which of the two variables to use for splitting. Using just p-values is unfair. For example suppose our second variable had been gender rather than age. Because there are only 2 genders, there is only 1 possible split point and one Chi-square test for gender while there might be hundreds of possible split points for debt to income ratio in an example data set. In this sense, debt to income ratio has more chances to be chosen as the splitting variable. This is reminiscent of the multiple testing problems that arise in the analysis of variance where the Bonferroni correction, multiplication of the p-value by the number of possible split points, is often used to compensate for the consequences of multiple testing. A p-value 0.002 on gender is smaller after adjustment than a p-value 0.0001 on debt to income ratio after adjustment if there are more than 20 possible split points for debt to income ratio. Multiplying 0.0001 by anything more than 20 produces a number larger than the 0.002 p-value for gender thus leading to gender as the variable of choice for splitting.

Another splitting option is the Gini method. Suppose you have k categories. A population would be diverse if, when 2 elements are selected, they are likely not to be from the same category. The probability of this happening is 1 minus the sum of squared probabilities from each category. For three categories containing 40%, 50% and 10% of the population respectively, the Gini index is $1-.16-.25-.01=0.58$ whereas with 30% 30% and 40% it is $1-.09-.09-.16=0.66$ so the first population is less diverse and more uniform. We want uniformity within leaves so we choose to split where the sum of Gini indices in the child nodes is as much smaller than that in the parent node as possible. There is no p-value to adjust for different numbers of split value choices so Gini tends to grow larger trees than the default Chi-Square method.

To prevent too many tiny rectangles from arising, a limit on how few points a rectangle can have and a critical limit for the Bonferroni adjusted Chi-square p-value are used. Finally, no lender could survive with the default rate suggested by the ratio of red to total dots in the rectangles of Figure 1. Most likely the analyst oversampled defaulters and undersampled nondefaulters. This is typical in analyzing relatively rare events. Adjustment for the oversampling must have been done in the process of attaching probabilities to the rectangles. Costs can also be taken into account if the cost of calling a nondefaulter a defaulter differs from the cost of calling a defaulter a nondefaulter.

Several other ways of growing out trees have been suggested in the literature and are available in SAS® Enterprise Miner® using other criteria than Chi-Square and we are not limited in SAS to having a bivariate response, or even a categorical response.

Why is this method of splitting a data set into subsets referred to as building a tree? The answer lies in the way the results are displayed. We turn to a famous real data set, the Framingham Heart study, to illustrate. A decision tree built on this data, without the use of a validation data set is shown in Figure 2. Changes from default setting were: Gini split criterion, N=4 leaves, assessment = average square error.

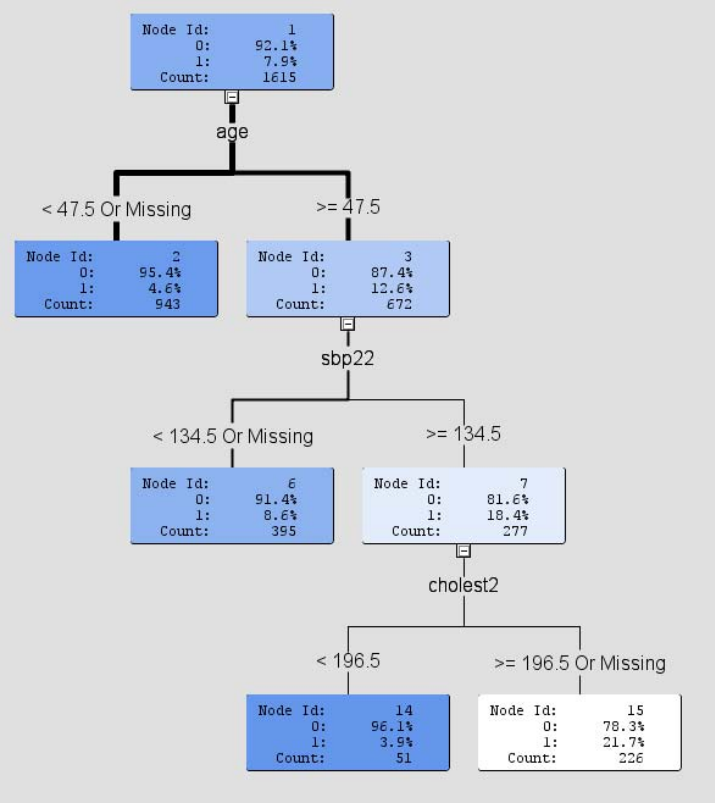


Figure 2. A tree built with SAS Enterprise Miner

The goal here is to discover and document factors (features as data miners call them) that predict incidence of heart problems, specifically first stage coronary heart disease. The data set has many of these features. The tree in Figure 2 suggests that youth (age 47 or younger) is a protective factor. For this group the incidence rate of first stage coronary heart disease is 4.6%, much lower than the overall 7.9% incidence rate and no further splits are done. For older people, keeping Systolic Blood Pressure under 135 seems to have a protective effect, bringing the incidence rate back closer to the overall rate. Older people with high blood pressure, however, had a quite high incidence of first stage coronary heart disease, 18.4% versus an overall 7.9% rate in the full data. The most interesting result is the last split for older high blood pressure people. It appears that even in this group, the 18.4% incident rate will drop to 3.9%, even lower than the overall population rate, if cholesterol can be controlled. Note that this advantageous node has only 51 of the original 1615 participants so the 3.9% rate should be taken with a grain of salt.

The tree algorithm starts with a splitting method to build a large tree using a split criterion. It then prunes it back using an assessment measure based on the validation data set (discussed later) if available or if not, on the training data. The assessment used to determine whether a split should be kept or pruned is important. Our example used Gini for splitting and average squared error for pruning. A response 1 (heart problem) in a leaf with proportion 0.08 has error 1-0.08 = 0.92 and a 0 (no problem) has error 0-0.08= -0.08. If instead one uses decisions (the default) as a criterion, so that only nodes with incidence greater than 50% are classified as heart problem nodes, then under the default settings the root node would not be split at all. No sequence of splits results in a node with more than half of the subjects exhibiting heart problems. Because of this phenomenon, in building decision trees for rare events, it is often the case that all observations exhibiting the event are chosen and a similar number of cases without events are

included to get the overall proportion of events nearer 0.50. This is called oversampling. Adjustments for the unequal sampling can be applied within the tree building process as long as the original proportions are known.

A major advantage of truly large data sets is the ability to set aside a subset called the validation data set as mentioned earlier in this section. The data used for fitting is then called the training data set. It is often the case that the training data tree has some branches which, when evaluated under the assessment criterion and on the validation data, are not helping or are actually harmful and should be deleted. This deletion step is called pruning in the literature on decision trees. Pruning choices can be based on making decisions (defaulter vs. non-defaulter), on estimating probabilities for the two classes, or getting the leaves properly ordered from least to most likely to default. These different goals can result, not surprisingly, in different final tree structures. For example, a subset representing people with probability 0.40 of defaulting might be split into two with probabilities 0.32 and 0.46 if estimates of probabilities are desired but if it is simply a matter of deciding if those people are more likely to default or not, then that last split results in two subsets with the same decision (will not default) and the split does not help. No validation data set was used in the Framingham example. A large tree was grown based on the Gini criterion then pruned back using average squared error on the training data itself.

2.b TREES FOR A CONTINUOUS RESPONSE

We have seen that trees recursively split data into smaller and smaller subsets, resulting in a partition of the data into disjoint subsets in a way that is fairly easy for an observer to understand and such that within the subsets, the observations are as alike as possible and thus between subsets they differ maximally, subject to conditions that prevent an unreasonable number of miniscule leaves. We have thus far looked at binary responses and the process goes quite similarly for categorical target variables with more than two levels. A (typically unachievable) ideal within these constraints is a partition in which all observations within a given subset fall into just one of the categories.

What if the response is continuous, as for example the cost associated with an automobile accident? We again wish to partition our data with respect to a collection of inputs (a.k.a. features, predictors, or Xs) by splitting recursively on the inputs. What might be an ideal partition here? One (again typically unachievable) goal with realistically sized subsets would be to have each subset having the same response for all of its members. Close to that would be a tree in which each leaf has all of its responses not identical, but tightly clustered around the subset average. This would give leaves with little variation within and relatively large variation between them. If, for any such data partition, we compare the variation in the subset means to the variation within the subsets we are performing computations similar to those in an analysis of variance (ANOVA), the critical difference being that the pre-specified treatments of ANOVA are now replaced by data derived groupings. We must compensate for using data derived classes. Just as we used multiple Chi square tables and their p-values to find optimal splits in binary data, so now do we use the analysis of variance F tests and their associated p-values to compare multiple splits at each step of the tree building process. Once we compare p-values as a selection criterion we can again compensate for the unequal number of possible splits when comparing, for example, speed on impact with its many levels as a possible split variable versus number of engine cylinders with few levels. We do this with a Bonferroni correction as before.

Here in Figure 3 is such a tree created from some data on the “cost to society” for traffic accidents in Portugal (personal communication from Guilhermina Torrao, NCSU Institute for Transportation Research and Education). The units are multiples of the overall cost of a fatality as estimated by a previous researcher. The first split separates accidents involving alcohol (mean cost 0.50) from those not involving alcohol or with that information missing (mean cost 0.16). For those involving alcohol or drugs, no further features were deemed helpful in predicting the cost of the accident. For the others, the engine size of vehicle 1, the vehicle deemed to have caused the crash, provides the next spilt with larger sizes implying higher costs. For the smaller size engines a split on whether or not it was a rear end collision is next. If a small engine vehicle rear ends another, the average cost is smaller than if it hits another vehicle in the side or head on or if it hits another object like a tree. The rear end collision group then splits again on vehicle 1’s engine size with, perhaps surprisingly, the smaller of these already small engine sizes giving a higher cost 0.15 than the larger engine sizes at 0.07. Perhaps within the accident causing cars with small engines, there are more injuries inside the truly small cars than inside those with intermediate size engines. Also note that one of the subsets has only 6 data points in it.

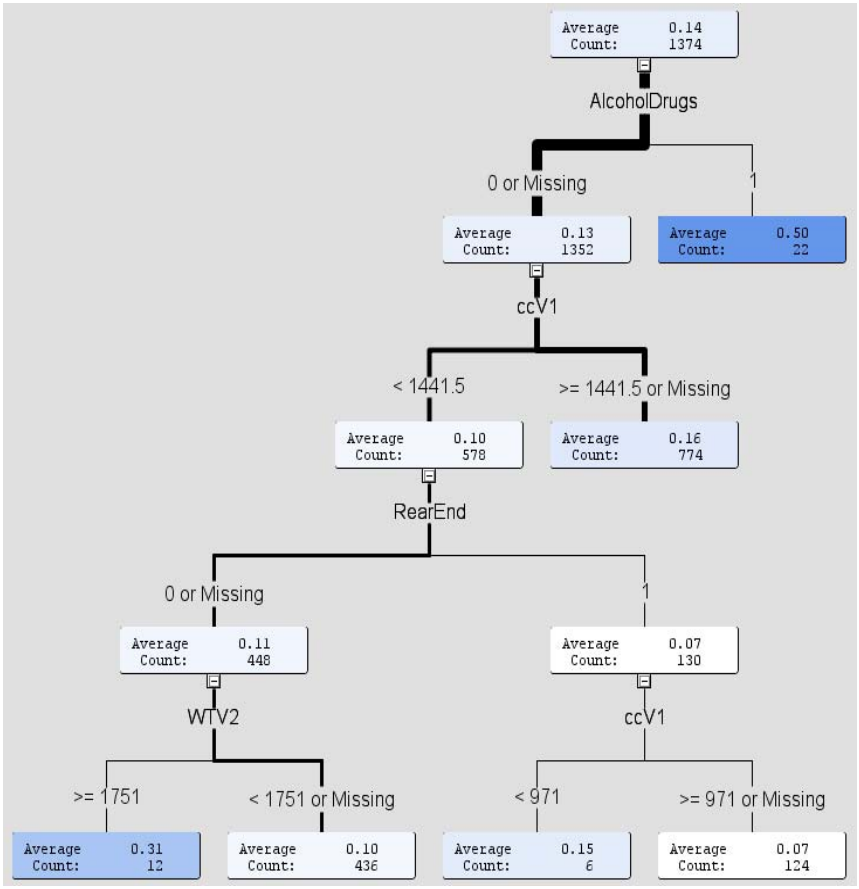


Figure 3: Cost to society of automobile accidents (courtesy of Guilhermina Torrao).

We have not yet discussed the left two subsets (leaves) at the bottom of Figure 3. These split off from the subset of the accidents in which no alcohol was involved and vehicle 1 had a small engine but did not cause a rear end accident. Of these cars, those running into a heavy vehicle 2 (WTV2) sustain the highest average cost 0.31 of all subsets not involving alcohol. This might be, for example, a small vehicle crossing a barrier and running headlong into an oncoming heavy limousine. The other branch consists of accidents where the driver hit a light vehicle 2 head on or on the side, hit an animal, or ran off the road into a boulder or tree for example.

3. REGRESSION MODELS

Included in a good set of data mining tools are some older, standard techniques. For the Framingham and accident data sets, logistic regression and ordinary regression respectively provide alternative standard approaches. Note however that the kinds of interactions discovered by the tree approach might be very hard to approximate with any kind of regression. Ordinary regression is such a common tool that I defer to standard textbooks for explanation. A logistic regression is appropriate when a binary response is to be modeled. What are the characteristics of a binary response? Suppose I shoot at a basket 200 times from the foul line and write down a 1 when I make the basket, 0 when I miss. The average of these 200 resultant 0 and 1 values, say 120/200 = 0.60 would be my sample proportion of made baskets. It is an estimate my probability of making a basket from the foul line on any future shot. Now if I made these 200 shots from all different distances from the basket, I would expect my probability of making a basket to decrease as distance increased. I want my probability, which must lie between 0 and 1, to change with distance. I do not want to use regression of my 0-1 responses on distance as that would give me a linear prediction of the

probability which in turn would have to rise above 1 and fall below 0 for extreme distances in the positive and negative directions.

Figure 4 shows a logistic regression which might result from a hypothetical situation in which 5 packets of food are stored at different temperatures (degrees centigrade) and a response variable Y, Y=1 if the packet spoiled and Y=0 otherwise, is recorded for each package. The points (X,Y) are seen along with the logistic regression plot which represents the probability of spoilage as a function of temperature (it could also represent 5 basketball shots with X being the shooter's distance in feet from the free throw line where positive X is a move towards the basket).

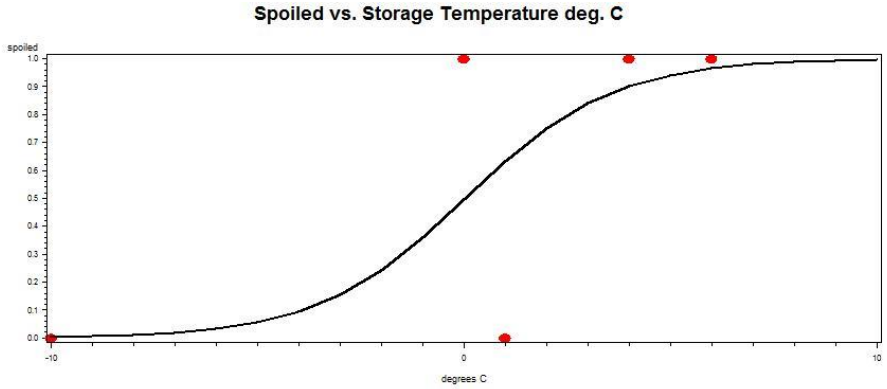


Figure 4. Logistic regression for Pr{spoilage} versus temperature.

The likelihood for the plotted points going from left to right is $(1-p_1)p_2(1-p_3)p_4p_5$ where p_i is the probability of spoiling for the i^{th} observed temperature (-10, 0, 1, 4, 6 ^oC). How can we relate p_i to X_i while ensuring $0<p_i<1$? Let $L_i = \beta_0 + \beta_1 X_i$ and $p_i = \exp(L_i)/(1+\exp(L_i))$. Notice that no matter what number you have for L, the associated exp(L) will be positive and the fraction $p=\exp(L)/(1+\exp(L))$ will thus also be positive. Clearly p will be less than 1 as well no matter how large or small is L. We find, for temperatures X in {-10, 0, 1, 4, 6 ^oC},

$(1-p_1) = 1/(1+ \exp(\beta_0 - 10\,\beta_1))$ for X=-10
$(1-p_3) = 1/(1+ \exp(\beta_0 + 1\,\beta_1))$ for X=1

$p_2 = \exp(\beta_0 + 0\,\beta_1)/(1+ \exp(\beta_0 + 0\,\beta_1))$ for X=0
$p_4 = \exp(\beta_0 + 4\,\beta_1)/(1+ \exp(\beta_0 + 4\,\beta_1))$ for X=4
$p_5 = \exp(\beta_0 + 6\,\beta_1)/(1+ \exp(\beta_0 + 6\,\beta_1))$ for X=6.

The likelihood is the product of these 5 factors, which will be maximized if we minimize -2 ln(likelihood). Notice that the product of the 5 factors involves only 2 unknown quantities, β_0 and β_1 so we will minimize -2 ln(likelihood) by searching over β_0 and β_1. Figure 5 shows -2 ln(likelihood) as a function of β_0 and β_1 and we see labeled the values that minimize -2 ln(likelihood) and thus maximize the likelihood. There is a very famous large sample statistical test, the likelihood ratio test, which compares a full model to a reduced model by taking the difference in the -2 ln (likelihood) values and comparing it to a Chi square critical value whose degrees of freedom are the number of parameters omitted in reducing the model. In Figure 5, that 5% critical value has been added to the minimum point to form a ceiling. Whenever a (β_0, β_1) pair has -2 ln(likelihood) exceeding this ceiling that ceiling replaces -2 ln(likelihood). The egg shaped hole in the ceiling thus represents a type of 95% confidence region for the (β_0, β_1) pair.

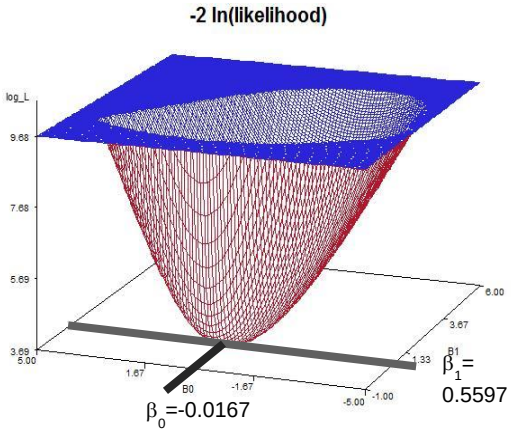


Figure 5. The -2 ln(likelihood) plot for the food spoilage example.

Had a normal rather than a binomial distribution governed the response variable, the confidence region would have been an ellipse rather than this egg shaped region.

3.a A REAL DATA EXAMPLE

The U.S. space shuttle missions have for the most part been successful however the 24th mission on January 28, 1986, involving the shuttle Challenger exploded upon takeoff with all aboard losing their lives. The cause of the explosion is thought to be failure of an O-ring which eventually caused an explosion of a fuel tank. Each mission uses six O-rings. O-rings are recovered and inspected. When hot gasses pass by an O-ring, which can happen prior to the O-ring sealing completely, this is called “blowby” whereas a more permanent hole in the O-ring is referred to as “erosion.” We will classify an O-ring as a failure if it experienced erosion or blowby. Data and further discussion can be found in Dalal, Fowlkes, and Hoadley (1989).

Initially, a random mission effect was included in a logistic regression of failure on launch sequence and temperature which was fit in PROC GLIMMIX. There was no evidence of a random mission effect so we proceed with a logistic regression using launch number and launch temperature as inputs (features). Having checked for a mission effect variance component, we now are cheking for a possible linear trend in missions over time using the launch number. Because there is no random mission effect, we treat all 23x6 = 138 O-rings as independent. Table 1 shows the output.

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	4.0577	3.0314	1.7917	0.1807
temp	1	-0.1109	0.0445	6.2122	0.0127
launch	1	0.0571	0.0563	1.0311	0.3099

(table continues next page)

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
temp	0.895	0.820	0.977
launch	1.059	0.948	1.182

Table 1: Logistic regression for shuttle data.

Here we see that the estimate of the logit L is 4.0577 – 0.1109(temperature)+0.0571(launch) and we recall that for any temperature and launch number, the probability of failure on a particular single O-ring is p=exp(L)/(1+exp(L)). If your favorite team has probability 2/3 of losing its next game, the odds against your team are 2/3 to 1/3, that is, 2 to 1. In general, p/(1-p) = [exp(L)/(1+exp(L)] / [1/(1+exp(L))] = exp(L) is the so-called odds (of O-ring failure here). We see that an increase of 1 unit in temperature decreases the logit L by 0.1109 and the odds go from exp(L) to exp(L-0.01109)= exp(L)exp(-0.01109). The ratio of the second odds to the first is seen to be exp(-0.1109) = 0.895 which is called the "odds ratio" for this reason. For the shuttle data each unit increase in temperature reduces the odds of failure to about 90% of what it was previously. Launching at higher temperatures is better. It is important to note that it is the odds, not the probability, upon which the temperature has a constant effect. An odds ratio 0.9 like this reduces a probability of 70% down to 63%, a reduction of 0.07 in probability whereas it reduces a probability .20 by .06, not .07. The change in probability is not constant.

To finish this example, a plot of the probability of failure of a prespecified single O-ring (left) and, using the binomial probability function for each p, the probability of 4 or more of the mission's O-rings failing (right) are shown in Figure 6. The leftmost temperature in both plots is 31 degrees, the Challenger launch temperature. The highest tickmark is at p=0.875 on the left and at p=0.970 on the right. Moving along the launch number axis, front to back in the plot, we see a slight but statistically insignificant increase as expected from Table 1.

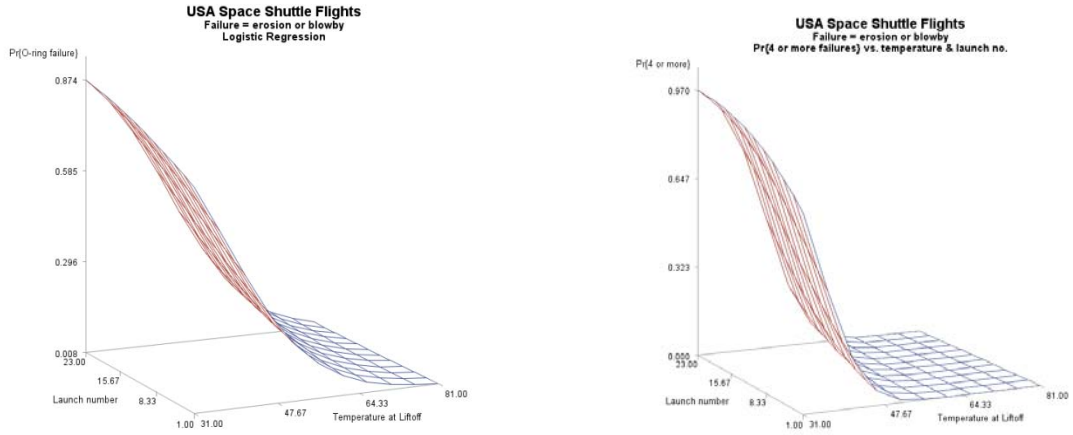


Figure 6: Probability for one prespecified (left) and four or more (right) O-ring failures.

4. NEURAL NETWORKS

These models are, at least in the basic cases, simply compositions of functions. Recall that a composition of functions Y=f(p(L)) computes a function of L, p(L) as input to function f( ) from which Y is found. In Neural Networks these functions are linear or logistic-like by default although other functions are also available as options. Here as before, L is a linear combination of inputs X_1, X_2, … X_n. Previously we computed the probability p(L) = exp(L)/(1+exp(L)) to estimate the probability p. The function p(L) expresses p as a function of L and is referred to as

a logistic function. Its inverse L=ln (p/(1-p)) gives us what is called the logit link function to use in logistic regression. Whenever L exceeds 0, p exceeds ½ suggesting an event is more likely than a non-event. The range of the logistic function is between 0 and 1, the allowable range for probabilities.

The logistic-like function used in the neural network node of SAS Enterprise Miner is called a hyperbolic tangent. Again thinking of a linear combination L of our features as input to the function, the hyperbolic tangent function is H(L) = (exp(2L)-1)/(exp(2L)+1)= 2[exp(2L)/(1+exp(2L))] - 1 = 2p-1 where the last expression shows that if we use a logistic function evaluated at 2L, p=[exp(2L)/(1+exp(2L))], then H(L) = 2p-1 and thus H(L) ranges between -1 and 1 in a sigmoidal fashion just as the logistic function ranges between 0 and 1. In essence then, H(L) is simply a rescaled logistic function. Because the coefficients relating L to the Xs are estimated and linear combinations of the hyperbolic tangent functions are used in neural networks, there is no difference in results for a neural network using logistic functions and one using hyperbolic tangents.

Figure 8 shows the predicted probability of an event versus 2 predictors X_1 and X_2 using a neural network type of structure with three "hidden units" H, each a hyperbolic tangent. A flow chart representation of the network is helpful for understanding the general structure.

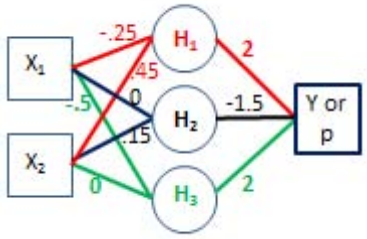


Figure 7. Neural Network Diagram for Figure 8.

From each of the hidden units H we see 2 lines extending to the left, one for each X. Labels are the coefficients. Along with the constant, these coefficients provide the linear combination of the Xs that is the argument of the hyperbolic tangent function. From the three H circles we also see three lines extending rightward which again are labeled with weights to represent the linear combination of H values (the hyperbolic tangent values) that, along with the constant term, give the continuous response variable Y. Of course Y might also be a logit that is then transformed to a probability p. A plot of p versus X1 and X2 appears in Figure 8.

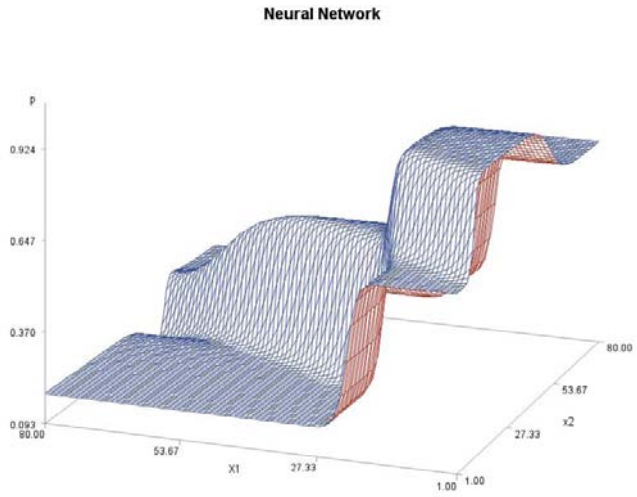


Figure 8. Neural network predictions: probability as a function of 2 predictors.

Once the coefficients are obtained, the response in Figure 8 is computed in steps:

(step 1) Connect the inputs X1 and X2 to the three "hidden units" using hyperbolic tangent functions of linear combinations of the Xs.

L_1=-4.5 -.25 X_1 + 0.45X_2 , H_1= (exp(2L_1)-1)/(exp(2L_1)+1)
L_2=-7.5 +0 X_1 + 0.15 X_2 , H_2= (exp(2L_2)-1)/(exp(2L_2)+1)
L_3=11 – 0.5 X_1 + 0X_2 , H_3= (exp(2L_3)-1)/(exp(2L_3)+1)

(step 2) Connect the hidden units to the logit Y of the probability using Y = -1.5 + 2H_1 -1.5H_2+ 2H_3

(step 3) Compute the probability from the logit Y p=exp(Y)/(1+exp(Y)

In summary, p is a logistic function of Y with in turn is a function of H_1, H_2, and H_3 each of which is a (hyperbolic tangent) function of X_1 and X_2. That is, we are looking at p as a highly nonlinear function of X1 and X2. In practice the coefficients given above would be estimated from data but here they are just selected to show the flexibility of a network graphically.

The flexibility of neural networks has its good and bad features. It is flexible enough to fit a complex surface, like that in Figure 8 where probabilities range from 0.004 to 0.982 as an exotic function of X_1 and X_2. Without control on the number of neurons or hidden units as the H functions are called, many of that surface's twists and turns could be modeling noise in the training data. It becomes very important to have a validation data set. Another down side, especially compared with trees, is difficulty in interpretation. Without Figure 8, it would be almost impossible to look at the equations in steps 1 to 3 and have any intuition on how the probabilities change with X_1 and X_2. Because there is no way to graphically render a surface in more than 3 dimensions, the results lack interpretability for a model with more than 2 inputs.

Various clever ways of estimating the coefficients have been developed. One way to think of this is that Y is a function of functions of the X's that we could write down then search over the parameter space for values that minimize the error mean square or optimize some other statistic. In other words, this is a nonlinear regression type of problem and hence involves an iterative search.

5. ASSOCIATION ANALYSIS

Basic association analysis is simply a matter of accounting in a large data set. It deals with "rules" such as B→A which means, in a marketing example, that purchasers of item B will also purchase A. In Figure 9 below, the rectangle represents all items sold in a store.

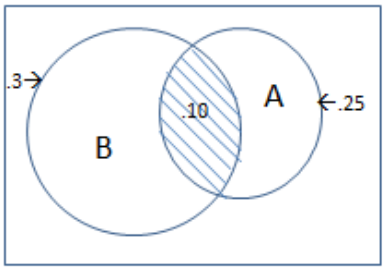


Figure 9. Market basket proportions for items A and B (not to scale).

The circle labeled B shows that 30% of market baskets have item B in them and the other circle shows that 25% of all baskets have item A. Now if the proportion of item B baskets that have item A in them is the same as the overall proportion of baskets with item A then there is no particular association of A with B – you get the same proportion of

A in baskets with item B as in those without item B. In such a case, with purchase of A independent of whether or not B was purchased, you would expect 25% of 30%, or 7.5%, of baskets to have both. We see that 10% of all baskets have both rather than the 7.5% we'd expect under independence. We conclude it is more likely that B purchasers will also purchase A than will non B purchasers. In fact it is 0.10/0.075 = 1.333 times as likely that a B purchaser will purchase A (probability .1/.3 = 0.3333 from the B circle) than the probability that a randomly chosen shopper will purchase A (probability 0.25). This ratio, or probability multiplier, 0.10/0.075 = 1.333 = 0.3333/0.25 is referred to as the *lift* of the rule B→A.

Association analysis lists rules, like B→A, along with the lift and two other quantities, *support* and *confidence*, that again are just new names for standard probability expressions. In practice, of course, the probabilities in Figure 9 are derived from the data in practice.

Lift: We have seen that lift is the probability of purchasing both A and B, 0.10, divided by the same probability under the assumption of independence (no relationship) which is 0.3(0.25) = 0.075. In general probability terms it is written Pr{A and B}/(Pr{A}Pr{B}) so by interchanging A and B we see that the lift for the rule A→B is the same as that for the rule we have been discussing, B→A.

Support: The support is simply the proportion of all market baskets containing both A and B or in probability terms Pr{A and B} which is 0.10 in this example. We see that the lift is the support divided by the expected support Pr{A}Pr{B} under the assumption of no relationship. Support is the same for the rule A→B as for B→A. The reason that the support is of interest is that in a typical association analysis, there is little interest in a rule that has good lift but the event almost never occurs. Generally a lower limit on confidence is used to prevent the reporting of rules for such rarely occurring events.

Confidence: This is simply the probability of purchasing A among shoppers purchasing B for the rule B→A. This is written in probability terms as Pr{A|B}, the probability of A given that B occurred. From basic probability or from just considering Figure 9, we know that Pr{A|B} = Pr{A and B}/Pr{B} which is .10/.30 = 0.3333 in our example whereas the rule A→B has confidence .1/.25 = 0.4000. Confidence, unlike lift and support, is not symmetric in its two arguments A and B. The probability of purchasing A is Pr{A}=0.25 in our example. If A and B were independent (no relationship) then 0.25 would also be the conditional probability of A given B, Pr{A|B} which we have seen is Pr{A and B}/Pr{B} so now it is clear that lift for the rule B→A is the confidence Pr{A and B}/Pr{B} divided by the expected confidence under the assumption of no relationship, Pr{A}. In summary we have seen that Lift = Pr{A|B}/Pr{A} = (Pr{A and B}/Pr{B})/Pr{A} = Pr[A and B}/(Pr{A}Pr{B}).

SAS Enterprise Miner computes support, confidence, and lift for all rules for which support exceeds a lower bound that can be adjusted by the user. It lists them along with some informative plots that allow a quick assessment of which rules seem useful. Rule like A&B→C are considered as well. For time stamped data, like information on what day different market baskets were purchased, a sequence analysis option is available that accounts for which item was purchased first. We look, for example, at the probability that item B will be purchased after item A.

6. ASSESSMENT

The useful idea of lift can be applied to any model thus far discussed and SAS Enterprise Miner allows a comparison of models in terms of lift. Suppose a logistic regression, neural network, and tree model have been fit to data with a binary response. Using the predicted probabilities from any of these models, the data can be lined up from most to least likely to produce an event according to the model, that is, from the highest estimated probability to the lowest. From this we can subset the data into the most likely 5%, the next most likely 5% and so on. For example suppose that the 5% subset deemed by the model to be most likely to produce an event has a 7% rate of event occurrence whereas overall there is only a 2% rate. The lift associated with that model at the best 5% subset would be 7/2=3.5. In a large mailing list of alumni with an estimated 2% overall rate of donating to the university, this lift figure would be stating that we are 3.5 times more likely to get a donation from someone who the model puts in the top 5% than just soliciting a random 5% of alumni. There would be a cumulative lift (based on best 5%, best 10%, best 15%, etc.) for each model and these can be overlaid on a plot as is done in SAS Enterprise Miner.

Cumulative lift can be used to select a model if some pre-specified percentile like the best 25% is used. If oversampling of events was used to get the data, this should be accounted for in the computation of lift. Models can also be compared based on profit/loss when the cost of incorrectly predicting an event differs from the cost of incorrectly predicting a non-event. Furthermore, there are several ways that predictions can be combined to get what is known as an ensemble model. SAS Enterprise Miner performs all of these tasks.

Sometimes it is of most interest to sort the data, based on a model, from most to least likely to produce an event. For example we might want to arrange mortage applicants from most to least likely to default based on debt to income ratio, age, and years of education. An underlying concept here is that of concordant and discordant pairs. Take two observations from historic data, one of which resulted in an event (e.g. default) and the other in a non-event. If the model gave a higher probability of event for that first point (the event one) than the second (the non-event one), the pair is called a concordant pair. If the model gave both the same probability, the pair is a tie. Finally if the model gave a higher probability of event to the second (non-event) observation than it gave to the first then the pair is deemed discordant. It is seen that this concept is related to the model's ability to correctly sort observations. Two other related terms are specificity (the probability of calling a non-event a non-event, i.e. calling a 0 a 0) and sensitivity (the probability of calling an event an event, i.e. calling a 1 a 1)

Now consider a decision tree with 3 leaves sorted in order from most to least likely to produce an event, that is, to deliver a response Y=1.

	Leaf 1	Leaf 2	Leaf 3	(Totals)
Number of 1's	20	20	10	(50)
Number of 0's	5	10	25	(40)

Table 2: Three ordered leaves from a decision tree with 50 1's and 40 0's.

We cannot distinguish items within leaves so our only reasonable decisions are

(1) Predict all 0s (sensitivity 0, specificity 1)
(2) Predict 1 in leaf 1, 0 otherwise (sensitivity 20/50 =.4, specificity 35/40=0.875)
(3) Predict 1 for leaves 1 and 2, 0 otherwise (sensitivity 40/50=0.8, specificity 25/40=0.625)
(4) Predict all 1s (sensitivity 1, specificity 0)

From these 4 decisions, we compute 4 points on the so called Receiver Operating Characteristic curve or ROC curve. Coordinates are (X,Y)= (1-specificity, sensitivity) giving points (0,0), (0.125,0.4), (0.375,0.8), (1,1). Figure 11 displays the ROC curve and some reference lines splitting the graph into rectangles and triangles.

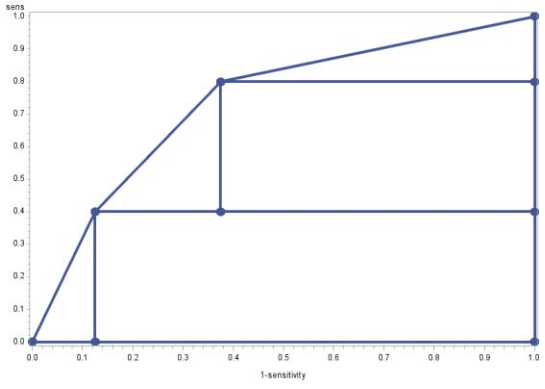


Figure 10. ROC curve for 3 leaf tree.

Consider decision 2, the bottom rectangle in Figure 10, and it's upper left corner at the point (0.125,0.4). The height of that rectangle 0.400 is the proportion of the 1s that fall in leaf 1 while the width 0.875=1-0.125 is the proportion of 0s falling in the other leaves. Since leaf 1 is the most likely to deliver 1s, the 1s and 0s just described form concordant pairs and the number of such pairs would be the product of their counts which, converted to proportions becomes the height times the width of the rectangle which is its area. This contributes to the proportion of concordant pairs. In the leftmost triangle the width is the proportion of 0s that fall in leaf 1 while the height is the proportion of 1s falling in leaf 1. These points are tied as they are (0,1) pairs that fall in the same leaf and hence have the same probability of being 1 based on our model. The product of these gives the proportion of ties so the area of the triangle is half the proportion of ties in leaf 1. Likewise if we move to decision 3, another rectangle and triangle appear above the first rectangle. Using the same reasoning, these are areas contributing more concordant and tied pairs. The remaining top triangle similarly has area half the proportion of tied pairs in leaf 3. Adding up to get the area under the ROC curve, we see that this area is the proportion of concordant pairs plus half the number of ties. The closer this area is to 1, the better the model. This area is sometimes called AUC for area under the curve and sometimes just the "C statistic."

The receiver operating characteristic curve's name derives from radar operators in wartime, some of whom were good at distinguishing enemy planes from friendly planes and some not based on the received radar signals – these receivers of signals had different operating characteristics.

As a further example, we see in Figure 11 in the left plot two normal distributions where the X axis represents income. The two curves represent incomes for home purchasers in the right hand curve and for non-purchasers in the left hand curve. Using vertical lines like the one shown, we can divide the income range at various points, deciding that people with incomes to the right of that point will buy and to the left they won't. As we sweep that vertical decision line across the plot, each time computing the area to its right on the rightmost curve, the sensitivity, and the area to its left on the left most curve, the specificity, we get a point on the ROC curve shown in the right panel. The point shown therein is the point associated with the displayed decision boundary in the left plot. The tangent line with slope exceeding 1 indicates that a small decrease in specificity (small move to the right) produces a larger increase in sensitivity.

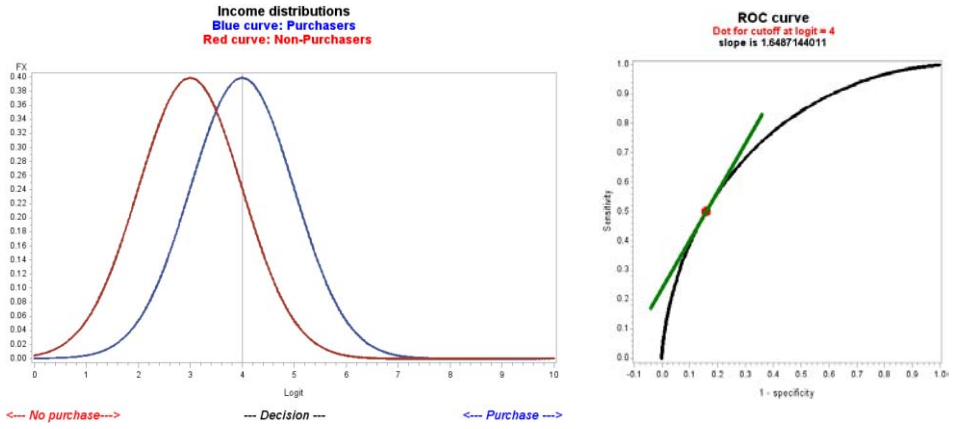


Figure 11. Computation of ROC curve

Using the point of intersection of the two probability density functions as a decision boundary results in a point on the ROC curve where the slope is 1, that is, where a small increase in sensitivity produces an almost equal drop in specificity.

7. UNSUPERVISED LEARNING

We have thus far been talking about supervised learning, so called because the data has a target variable, often binary, that gives feedback on the correctness of model predictions. Having a target allows the tuning of model

parameters to optimize decisions. The predictor variables, or features, are used to compute the predictions and make decisions. Some data sets have only the features, perhaps a vector of 5 measurements (X1, X2, X3, X4, X5) on each customer, and no target variable. For example, one might have age, education level, income, average age of children, and number of children for customers and might want to group them into sets with similar features, then look at the clusters, perhaps focusing on the "centroids," the 5 dimensional vectors of mean values for each cluster, to characterize the clusters and perhaps develop different marketing strategies for the different clusters. We might market differently to a cluster of young couples having less education and income but lots of small children than to some other cluster even if we do not have a target like purchase history available.

The distance between 2 points, like (30, 15, 45, 7, 3) and (32, 14, 40, 9, 4) is the square root of the sum of squared coordinate differences, e.g. the square root of 4+1+25+4+1=35 which is slightly less than 6. Clearly a variable like age that might range from 30 to 70 will have a larger impact on this calculation than, say, number of children that might range from 0 to 5 in a data set so it is usually advisable to center and scale the data before clustering as the clustering idea is to compute distances and place observations that are close together into the same cluster. There are several methods of clustering divided into direct clustering and hierarchical clustering. In hierarchical clustering of the agglomerative kind, you start out with each point in its own cluster then combine the two closest points and continue combining points with points, points with clusters, and clusters with clusters until you get to a desired number of clusters. Once two points are joined, they remain together in the same cluster. Divisive clustering starts with all points in one big cluster then recursively partitions the existing clusters until the desired number of clusters is reached. A well-known agglomerative method is Ward's method in which points and clusters are combined in such a way that the resulting increase in sum of squared distances from points to their cluster centers (summed over clusters) is minimal.

Direct clustering picks a chosen number of points as seeds or initial centroids, clusters each other point to the nearest seed, computes the new centroids, re-clusters, re-computes, re-clusters, etc. until there is no change in the clusters. SAS Enterprise Miner uses a combined strategy in which a large number of clusters are formed by direct clustering then agglomerated by Ward's method. A method for deciding on how many clusters to use, the Cubic Clustering Criterion, is described in SAS Institute (1983). Because there are no assumptions put on the distributions of the features, there has been no rigorous statistical theory developed for making this choice as far as I know.

8. COMBINED EXAMPLE

The goal here is to distinguish black circles from green dots based on (X_1,X_2). Figure 12 is a generated plot of green dots and black circles plotted by coordinates X1 and X2. Three models, a default neural network, a default tree, and a logistic regression with up to degree 3 polynomial terms were fit in SAS Enterprise Miner. For example, you could think of the coordinates as points of origin of cell phone texts with black indicating phones that move more than 50 yards during the first 2 minutes of texting.

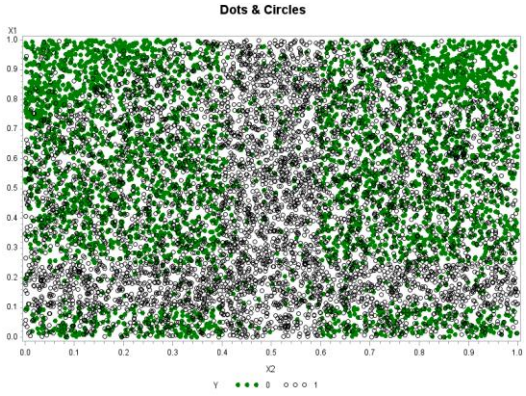
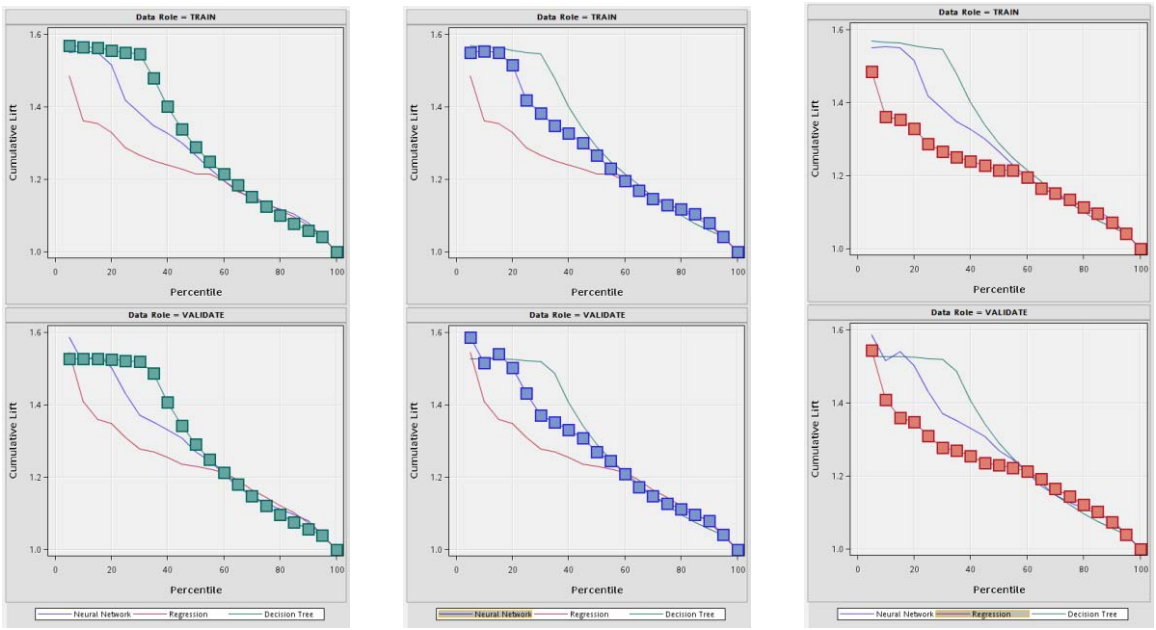


Figure 12: Distinguish black circles from green dots based on (X_1,X_2)

A model comparison node gave overlaid lift charts for the training (6000 points) and validation (4000 points) data as shown in Figure A, B, and C. On the left, the decision tree's lift is highlighted, in the middle the neural network and on the right the logistic regression's lift. Top panels are training and bottom validation data sets.

Tree Neural Net Logistic Regression



For most but not all percentiles, the decision tree has the best lift. Finally, in Figure 13 plots of the fitted surfaces are shown.

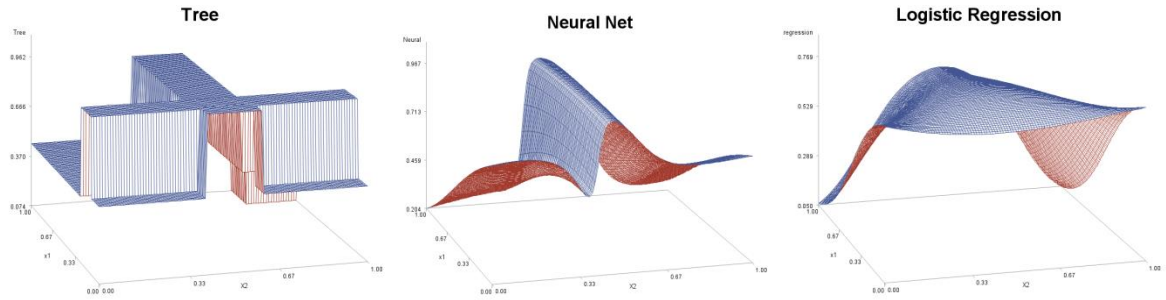


Figure 13: Fitted surfaces from 3 models.

The plots were obtained using the SAS Enterprise Miner generated code over a grid of (X_1,X_2) values. Using the texting idea, the tree result might signal the crossing of two roads, most of the black dots then indicating texting while driving.

9. CONCLUSION

Data mining on the analysis side is a collection of tools. Some of the basic tools, trees, logistic regression, neural networks, clustering, and association analysis have been reviewed here. SAS Enterprise Miner provides these and many other specialized tools for the user.

REFERENCES

Dalal, Fowlkes, and Hoadley (1989). "Risk Analysis of the Space Shuttle: Pre-Challenger Prediction of Failure," *Journal of the American Statistical Association, Vol. 84, #408, pp. 945-957*

SAS Institute Inc. (1983) *SAS*™ *Technical Report A-108 Cubic Clustering Criterion,* SAS Institute, Cary NC.

CONTACT INFORMATION

Name: Professor David A. Dickey
Enterprise: Department of Statistics
Address: Box 8203, North Carolina State University
City, State ZIP: Raleigh, NC 27695-8203
E-mail: dickey@stat.ncsu.edu
Web: http://www4.stat.ncsu.edu/~dickey/