Paper 079-31

Comparison of Data Preparation Methods for Use in Model Development with SAS Enterprise Miner®

By Charles Manahan, Ph. D.
Cingular Wireless, LLC.
Atlanta, GA

ABSTRACT:
SAS Enterprise Miner® is a powerful tool for model development. By providing a GUI interface, drag and drop model evaluation, automatic record keeping of multiple scenarios, and scoring code generation, it saves a great deal of time over the traditional method using SAS STAT® with the various regression PROCs. However, as most people who have developed predictive regression or other behavioral models are aware, the bulk of the time spent in developing models isn't in the final production of the regression coefficients, but in variable selection and preparing the input variables (e.g. clustering levels, imputing missing values, etc.).

Enterprise Miner has a module for variable selection and level condensation which is easy to use, but how well does this module compare with some traditional methods of variable selection? Several models were developed in Enterprise Miner in parallel using traditional methods for data preparation vs the Enterprise Miner variable selection module. The results were then compared in Enterprise Miner to evaluate various data preparation methods.

This is an intermediate level presentation and the audience should have knowledge of base SAS® and conceptual knowledge of model evaluation, regression, and variable selection techniques. The audience should also have some familiarity with SAS Enterprise Miner.

INTRODUCTION:

This was done using EM 4.3 and SAS 9.1.

The processes to build two very different models were compared using EM on the raw database and using some of the standard data reductions techniques in base SAS and then applying EM. The first of these models was a propensity to purchase (look alike) model for VAD (Voice Activated Dialing). This was done at the mobile level, a standard data mining task with large quantities of data.

The second was an analytical model to show correlation between signal strength in a market and churn in that market. The data were at the market level. Cingular divides its territory into geographic "Markets" with reporting values for most things being at the Market level. Unfortunately Markets are highly aggregated. For example, Virginia - West Virginia is one market. Arizona – New Mexico is another. This leads to the opposite problem from that which is usually found in data mining – namely this model had too little data.

When the task is to develop a model using a high dimensional opportunistic database of a couple of thousand possible independent predictor variables, the first task is to reduce the variables to "reasonable" number, and the second is to reduce the number of levels in the categorical variables. When the task is to develop a robust means of correlation with very few data elements, the path is somewhat different

This paper compares the results of two models developed in Enterprise Miner (see table I) using several variable preparation schemes. Namely, the schemes were variable clustering using principal components for numeric data reduction and using Greenacre's method of clustering to reduce the number of levels in categorical variables. The Voice Activated Dialing model is a propensity model for predicting possible customer behavior, and the Churn/NQ model is an analytical model designed to prove to the board of directors that it is worthwhile to spend capital funds on network.

Table I – Models

Abbr.	Title	Definition
VAD	Voice Activated Dialing	Product that allows customer to say the number rather than enter it on a keypad
Churn/NQ	Churn relation to Network Quality	Examination of those network measures that contribute significantly to customer loss

METHODS:

Cingular Wireless has a large database with many variables – some of which are appropriate for predictive models and some not. The first pass was a manual selection of those variables that were deemed to be possible predictors based on experience. For the VAD model, 50 variables – a mixture of about a dozen categorical variable with the remainder numeric was chosen. This dataset was then given a traditional modeling data reduction, and also fed directly into EM.

The data for the VAD (propensity) model was run through the varclus method = ward (greenacre – Macro example Appendix 1 – This is not original, but I've included it as a convenience) to do an initial screening of of categorical variables. An excellent discussion of this method is found in the SAS course notes Predictive Modeling Using Logistic Regression (1.)

The VAD model had a binary outcome. The Churn/NQ model was a continuous model. Since the Churn/NQ was assumed initially to be a linear regression, and there were a feasible number of variables to try all possible combinations, PROC RSQUARE (PROC RSQUARE is now a part of PROC REG.) with the Mallow's Cp option was used as a first screen. This is an old PROC, but still somewhat useful as can be seen by some sample output in Table II. RSQUARE computes the RSQUARE statistic for all possible combinations of variables as well as Mallow's Cp. It then prints the top few in each category with top few being defined as those combinations in each predictor group with the highest RSQUARE

RESULTS:

Two very different models were run through the process, and the effect of prior data preparation on the final outcome from EM was compared. The first was a predictive model built from a great deal of data and the second an analytical model built from very little data.

First we'll example the results from the predictive model.

The categorical variables for the VAD model were run through the Greenacre procedure. This procedure sets up a table with the frequency of each level and the proportion of the target value in each level. It then collapses the table level by level looking at the change in chi-square as the table is collapsed. Figures 1 and 2 give two visual examples of results of this procedure running against the VAD model.
Figure 1 is the result of looking at the tech type code and reduction of chi-square in a class variable with relatively few levels. You'll note that you can collapse seven levels into two with relatively little loss of information.

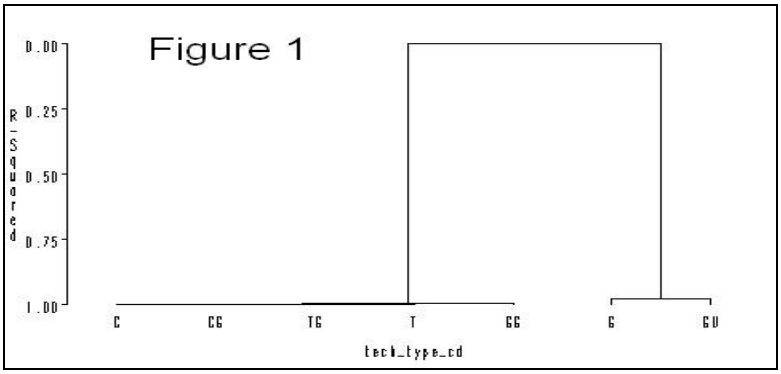
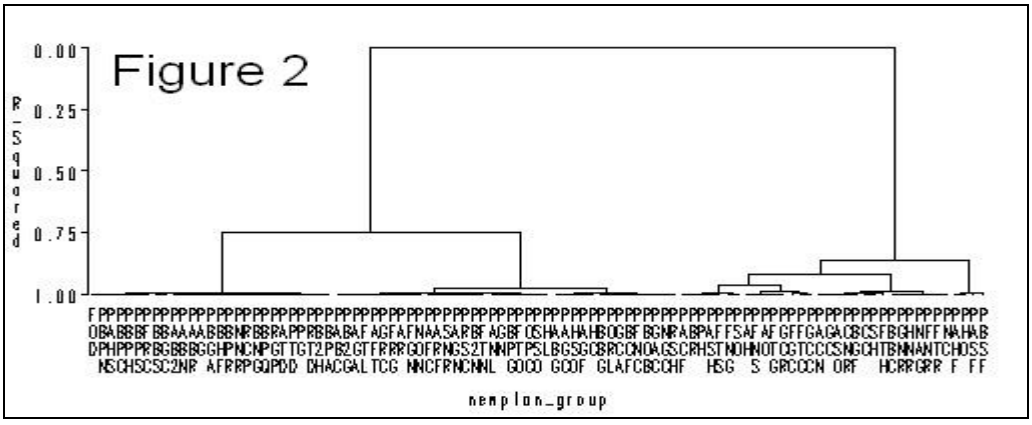


Figure 2 shows the reduction of class variable with roughly 99 levels



As you can see this can be collapsed to seven (or fewer) levels without much information loss. This procedure was applied to all of the class variables in the VAD model pool.

Next the numeric variables were run through an oblique principal component analysis using PROC Varclus to determine which were redundant. Varclus keeps splitting the variables until the split criterion (maxeigen) is reached. The SAS code to do this is in Appendix 2. The tabular result is in Table 1.

TABLE 1

Total Number of Clusters	Proportion Variation of Explained by Clusters	Minimum Variation Explained by Clusters	Maximum Proportion Explained by a Cluster	Maximum Second Eigenvalue in a Cluster
1	6.848511	0.2209	0.2209	3.132294
2	9.789094	0.3158	0.2377	2.469202
3	12.161461	0.3923	0.1906	2.056179
4	14.158030	0.4567	0.2178	2.056093
5	16.140185	0.5207	0.2760	1.924608
6	18.064780	0.5827	0.2760	1.900415
7	19.901773	0.6420	0.3593	1.198517
8	20.950904	0.6758	0.4552	1.036570
9	21.852217	0.7049	0.5265	1.010493
10	22.842035	0.7368	0.5265	0.999909
11	23.841944	0.7691	0.5265	0.993235
12	24.534667	0.7914	0.5814	0.964803
13	25.443033	0.8207	0.5959	0.929809
14	26.355279	0.8502	0.5959	0.894741
15	27.249729	0.8790	0.5959	0.808148
16	28.057877	0.9051	0.7446	0.603301

A look at the actual clusters (six of the sixteen) is in Table 2. Typically, the reduction method used is to pick the best representative from each cluster.

TABLE 2

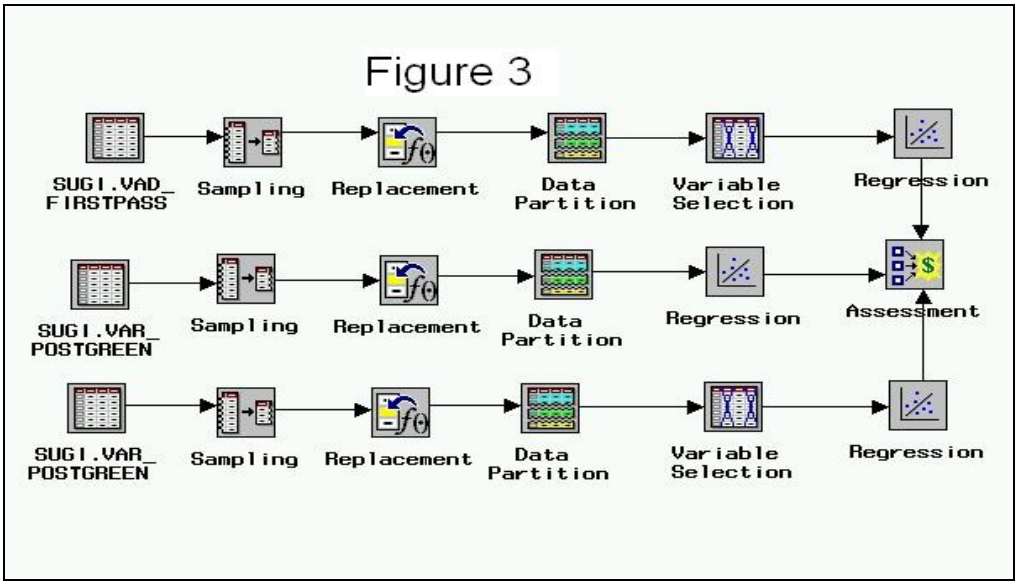
16 Clusters		R-squared with Own Cluster	R-squared with Next Closest	1-R**2 Ratio
Cluster 1	call_tot_qty	0.9053	0.1173	0.1073
	call_air_qty	0.9122	0.1177	0.0995
	call_locl_qty	0.7592	0.1056	0.2693
	min_tot_qty	0.8427	0.2029	0.1974
	min_air_qty	0.8605	0.2076	0.1761
	min_locl_qty	0.7493	0.1161	0.2836
Cluster 2	call_ela_qty	0.8314	0.0684	0.1810
	min_ela_qty	0.8779	0.0834	0.1332
	min_toll_qty	0.7214	0.1935	0.3454
	tot_orgnl_roam_amt	0.8284	0.0692	0.1843
Cluster 3	tot_chrg_amt	0.8317	0.2930	0.2380
	tot_air_chrg_amt	0.5949	0.0703	0.4358
	tot_tax_amt	0.8072	0.3503	0.2968
Cluster 4	call_lsa_qty	0.9983	0.0000	0.0017
	min_lsa_qty	0.9983	0.0000	0.0017
Cluster 5	tenurem	0.9976	0.0336	0.0025
	tenurey	0.9976	0.0297	0.0025
Cluster 6	call_eha_qty	0.9623	0.1197	0.0428
	min_eha_qty	0.9623	0.1364	0.0437

Using the output from these two processes, two SAS data sets were fed to EM. A “firstpass” data set which had the obvious irrelevant data (such as table update date stamps, etc.) removed, and a “postgreen” data set that had the Greenacre level collapse applied to the categorical variables generating new categorical data elements.

These were fed into EM as shown in Figure 3 to compare regression models, neural net models and decision tree models. This diagram is for the regression models. The other diagrams were similar and won't be show.

There were three regression models assessed. A model where the first pass data was just moderately cleaned (ie ID variables and target identified) and EM was allowed to do the rest with replacement and variable selection from EM (lazy man's model).

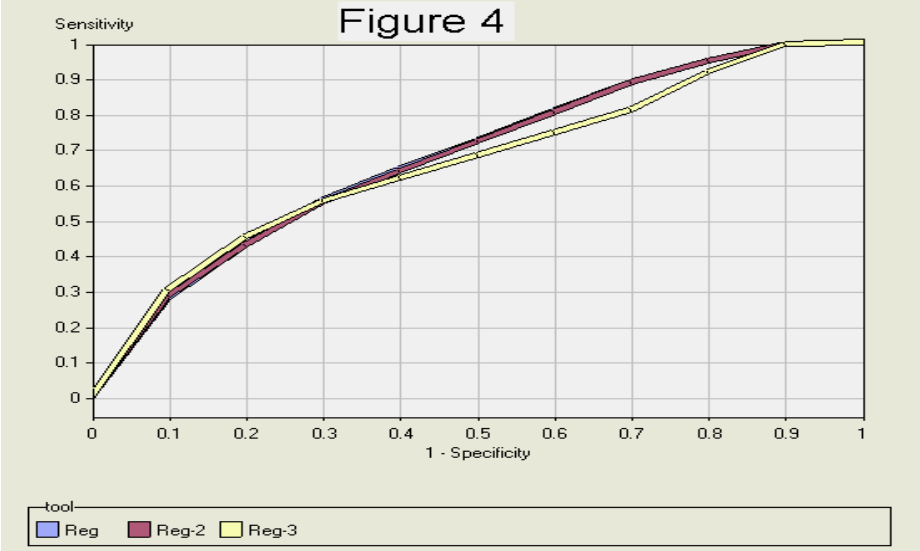
Figure 3



Next the dataset with the manually collapsed categorical levels was fed in. The only categorical variables allowed were those with the collapsed levels, and the only numeric variables (other than target and ID) were those selected by the VARCLUS method described above.

A final feed was the same "postgreen" dataset, but with no constraints except for target and ID. The EM Variable Selection module was allowed to pick from the whole set including the already collapsed replacement variables as well as the original categorical variables. The ROC charts from the three regression models are shown in Figure 4.

Figure 4



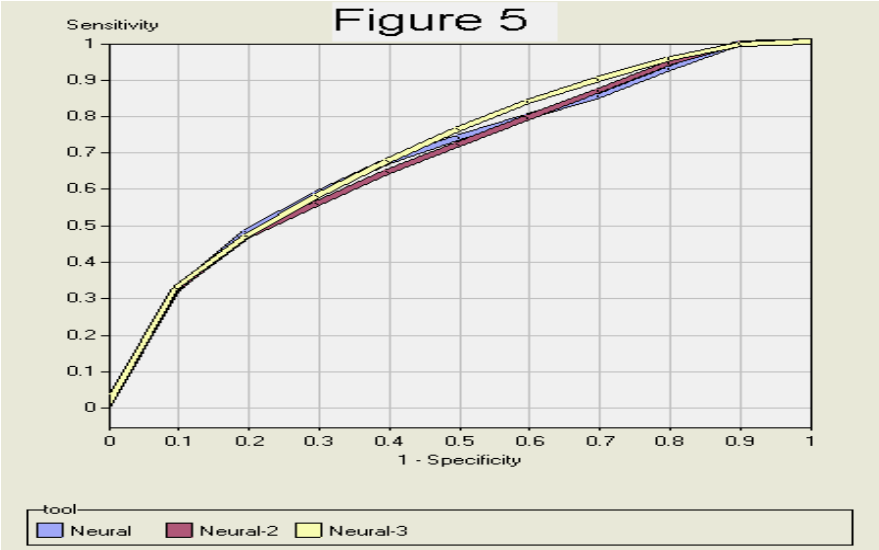
Reg and Reg-2 (blue and red respectively) are the ROC curves for the two regression models where the EM Variable Selection module selected the model variables. There is no significant difference between the two. The yellow line represents the ROC curve for the model that was done with manual selection. On first glance it appears to be poorer on the low end of the scale, and just a tiny amount better on the high end. However when the test misclassification rates are taken into account, the manually selected model has third decimal place improvement in the misclassification rate as shown in Table 3.

TABLE 3 - Regression

MODEL	TEST MISCLASSIFICATION RATE
EM Variable selection of raw data	0.2755911404
EM Variable selection of collapsed categorical data	0.2755911404
Manual selection	0.2718497456

The same data were then modeled with the neural net module and the ROC charts calculated. This is shown in Figure 5

Figure 5



In this case the EM Variable Selection (yellow line) on the base data gave the best result including the test misclassification. Also note that the neural net module gave a slightly better model (one percent lower) from the test misclassification viewpoint than the regression module.

TABLE 4 – Neural Net

MODEL	TEST MISCLASSIFICATION RATE
EM Variable selection of raw data	0.2608500449
EM Variable selection of collapsed categorical data	0.2694552529
Manual selection	0.2702035319

Finally the same data were fed to the decision tree module. The results for this are shown in Figure 6.

Figure 6

The three ROC curves shown in Figure 6 were not all that different with the manual data curve (yellow) being slightly better than the two curves produced from the data selected by the EM Variable Selection module. The test misclassification rates, shown in Table 5, also showed a slight advantage (about two tenths of a percent) to the manually prepared data.

TABLE 5 – Decision Tree

MODEL	TEST MISCLASSIFICATION RATE
EM Variable selection of raw data	0.2464830889
EM Variable selection of collapsed categorical data	0.2480544747
Manual selection	0.2440885962

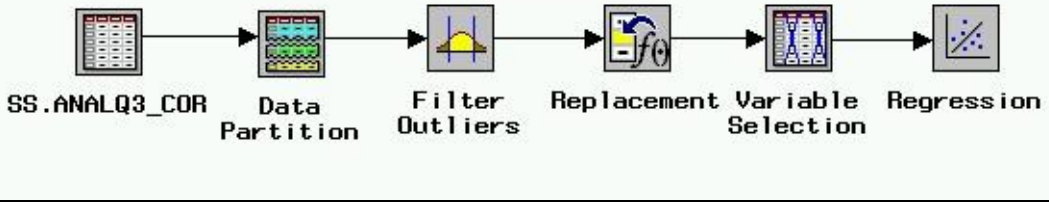
It is also worth noting that the decision tree gave the best overall model (as judged by lowest test misclassification rate) by a slight margin.

Here are the results for analytical model.

First some background on the model: The outcome for this model was assumed to be continuous and not necessarily linear, although as a first pass linear regression was used to determine the degree of model fit. It is an industry truism, born out by several internal and external studies that network issues account for roughly one fourth of the churn (customers leaving and taking their business to other carriers). It was felt by the board of directors that with the levels of capital spending on network build out that this relationship should be examined again with the most recent data. In this case we were expecting a model that would have an r-square of somewhere from 0.2 to 0.3.

The data were first run through a fairly standard diagram for regression. See Figure 7

Figure 7



The output section of the EM Regression Module results indicated a surprisingly high r-square of 0.74. See Table 6. The Mallow's C(p) of 8 was not unreasonable for a model of this size.

TABLE 6 – Linear Regression Model Fit from EM

Model Fit Statistics

R-Square	0.7417	Adj R-Sq	0.6793
AIC	-410.0117	BIC	-403.7501
SBC	-397.1243	C(p)	8.0000

However, since previous studies had indicated that 0.2 to 0.3 was the best r-square that you could expect from these data something was obviously wrong. Multicollinearity often leads to spuriously high r-squares, so SAS STAT was used to test for multicollinearity.

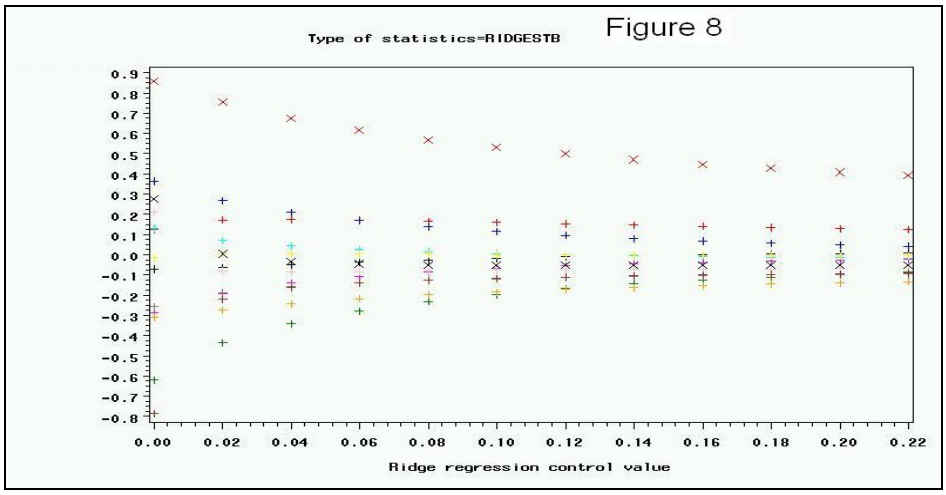
Going back to the original data and running SAS STAT's PROC REG with the collin vif and tol options on the model statement yielded the results shown in Table 7

TABLE 7 – Collinearity Statistics

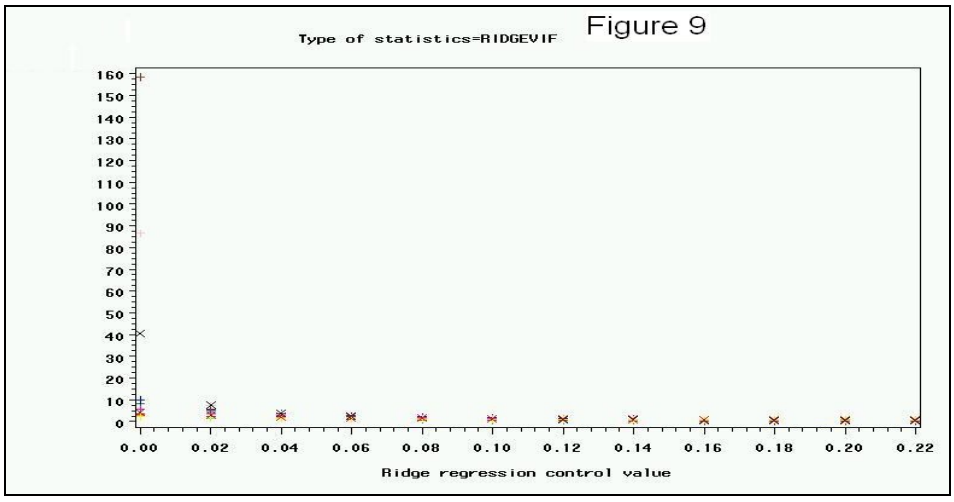
Variable	DF	Variance Inflation
Intercept	1	0
Max_Value_Group__dBm_	1	2.44885
Average_Value__dBm_	1	3.61294
Percent_of_ED_Group____98_dBm	1	8.32148
Avg__Value_Dense_Urban__dBm_	1	9.85550
Percent_of_Dense_Urban_Area____9	1	2.06232
Min__Value_Urban__dBm_	1	4.74733
Max__Value_Urban__dBm_	1	2.90615
Avg__Value_Urban__dBm_	1	89.15592
Max__Value_Dense_Suburban__dBm_	1	2.52458
Avg__Value_Dense_Suburban__dBm_	1	159.34306
Avg__Value_Suburban__dBm_	1	35.94983
Percent_of_Suburban_Area____98_d	1	4.2644

While there isn't any generally accepted absolute value for variance inflation factor to show multicollinearity, the highlighted values are indicative. In addition, the collinearity diagnostic table that shows several factors with a high condition index that contribute more than .50 to the proportion of variance of two or more variables although the table itself has been omitted for space reasons.

The data were then tested for multicollinearity using the ridge option of proc reg and the VIFs and the standardized parameter estimates were plotted against the ridge factor with the code in Appendix 3. The results of plotting the output of a second test using SAS STAT running ridge regression against the data are shown in Figures 8 which is the reduction in standardized parameter estimates vs. ridge parameter value and in Figure 9 which shows the drastic reduction in the VIF with increasing ridge parameter value.



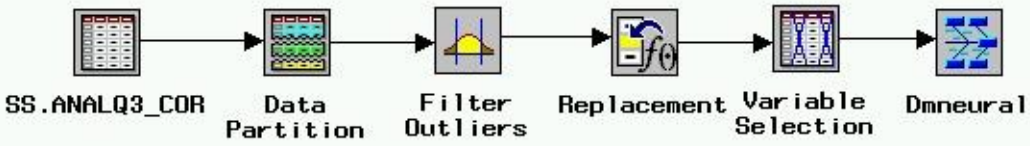
The various colored X's represent the estimates for the independent variables. It isn't important which ones they were so much as the pattern of decrease with increasing k. Figure 9 is a plot of VIF vs. ridge value.



Note the reduction of 160 to 10 in VIF when going from a ridge coefficient of zero to 0.02. Again this is indicative of multicollinearity.

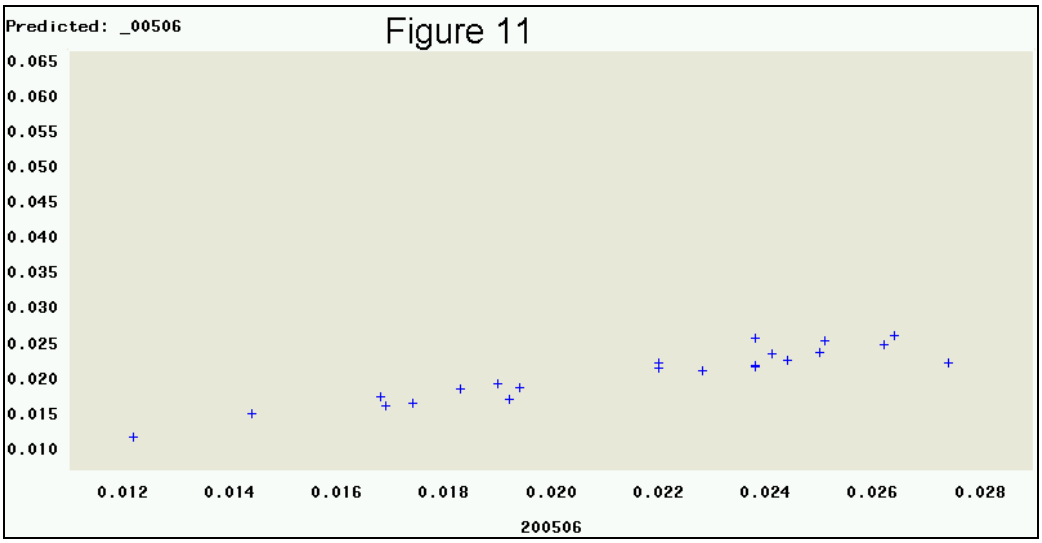
I could have used principal components, factoring, orthoreg, or partial least squares regression in SAS STAT to produce the model, but there is in EM a Princomp/Dmneural module that not only automatically does the principal components, but uses non-linear activation functions to account for non-linearity of response. Dmneural can be set to do the principal components only and combined with regression to give principal components regression without resorting to SAS STAT (see Appendix 4), but this was not done so as to keep the nonlinear link functions provided by the neural net.

Figure 10



Princomp/dmneural was used on the data as in the diagram in Figure 10. The final fit of the Princomp/Dmneural model is shown in Figures 11 and 12. Figure 11 is the EM Dmneural plot of predicted churn vs actual churn for June of 2005.

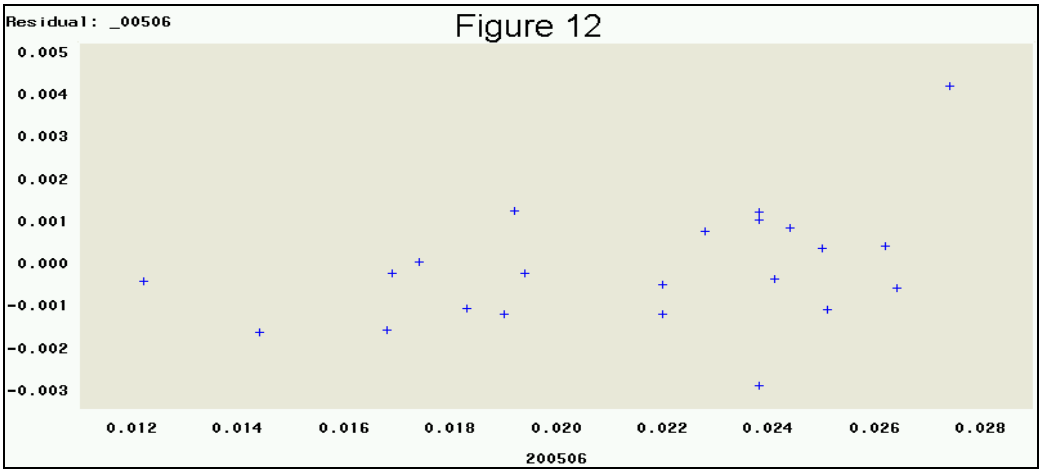
Figure 11



The plot of predicted vs. actual shows a distinct relationship with not all of the actual being accounted for in the predicted. This is what you would expect if network quality was only a partial component of churn.

Figure 12 is the EM Dmneural plot that shows the residuals for the data

Figure 12



The residuals show a slight pattern , but overall show that the model fit is not random.

CONCLUSIONS:

There were minor differences in the results using Enterprise Miner's variable selection module and some of the traditional methods of dimensionality reduction. If time is of the essence then EM alone provides a sufficient answer

for predictive modeling; however there are some slight improvements possible with preprocessing the data. If the campaign is large enough, then preprocessing can provide some additional benefit.

The standard route (figure 7) for setting up regression in EM doesn't flag multicollinearity. If it's suspected, there are a number of tests that can be done in SAS STAT to confirm it, and EM can deal with it by either doing a principal components regression (Appendix 4) or using the Princomp/Dmneural module.

APPENDIX 1

```
%macro green(clv,dtstp) ;
dm 'clear lst';
 proc means data = &dtstp   nway  ; var target;
class &clv ;
output out = levelspp mean = prop;
 run;

ods trace on /listing;
 proc cluster data = levelspp method = ward outtree = fortree;
freq _freq_;
var prop;
id &clv ;
run;
ods trace off;

ods listing close;
ods output clusterhistory = cluster;

 proc cluster data = levelspp method = ward   ;
freq _freq_;
var prop;
id &clv;
run;
ods listing;

proc print data = cluster;run;

proc freq data = &dtstp noprint;
table &clv * target/chisq;
output out = chi(keep = _pchi_) chisq;
run;

proc print data = chi;run;

data cutoff;
if _N_ = 1 then set chi;
set cluster;
chisquare = _pchi_*rsquared;
degfree = numberofclusters - 1;
logpval = logsdf('CHISQ',chisquare,degfree);
run;

proc means data = cutoff noprint;
var logpval;
output out = clusop minid(logpval(numberofclusters))= ncl;
run;
data null;
set clusop;
call symput ('ncl',ncl);
run;
proc tree data = fortree nclusters = &ncl out = clus h= rsq; id  &clv ;run;
proc sort data = clus ; by clusname;run;
proc print data = clus;run;
*end greenacre ;
%mend;
```

APPENDIX 2

```
proc varclus data = mdl.vad_firstpass (drop = call_solicit_ind)  maxeigen = .8 outtree = fortree
short;
 var call:min: tot: tenur:;
run;
```

APPENDIX 3

```
proc reg outest=ridge outvif outstb ridge=0 to .23 by .02
  data = ss.analq3_cor(where = (type ne 'Migrate')) ;
  model _00506 = Max_Value_Group__dBm_ Average_Value__dBm_ Percent_of_ED_Group____98_dBm
  Avg__Value_Dense_Urban__dBm_ Percent_of_Dense_Urban_Area____9 Min__Value_Urban__dBm_
  Max__Value_Urban__dBm_ Avg__Value_Urban__dBm_ Max__Value_Dense_Suburban__dBm_
  Avg__Value_Dense_Suburban__dBm_ Max__Value_Suburban__dBm_ Avg__Value_Suburban__dBm_
  Percent_of_Suburban_Area____98_d  ;
```

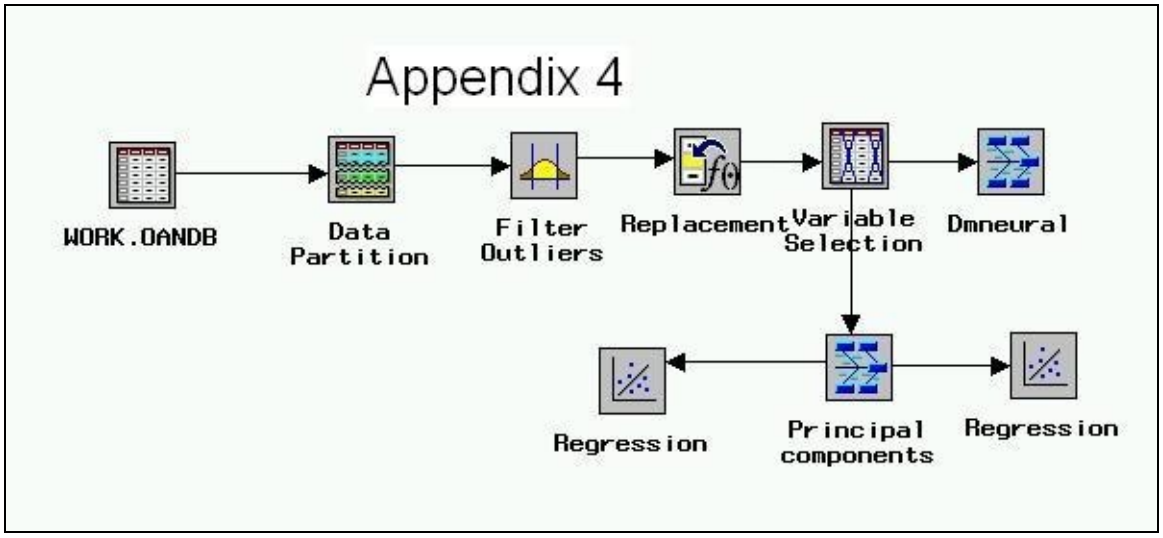
APPENDIX 3 (cont)

```
run;

data new; set ridge;
if _type_='RIDGESTB' or _type_='RIDGEVIF';
run;

proc sort data = new; by _type_;


proc gplot; by _type_;
plot (  Max_Value_Group__dBm_ Average_Value__dBm_ Percent_of_ED_Group____98_dBm
  Avg__Value_Dense_Urban__dBm_ Percent_of_Dense_Urban_Area____9 Min__Value_Urban__dBm_
  Max__Value_Urban__dBm_ Avg__Value_Urban__dBm_ Max__Value_Dense_Suburban__dBm_
  Avg__Value_Dense_Suburban__dBm_ Max__Value_Suburban__dBm_
  Avg__Value_Suburban__dBm_
  Percent_of_Suburban_Area____98_d)*_RIDGE_/overlay;
run;
```



TRADEMARK CITATION:

SAS and all other SAS Institute 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 registered trademarks or trademarks of their respective companies

REFERENCES

1. SAS Institute, 2001, Predictive Modeling Using Logistic Regression, SAS Institute, Cary, NC

2. SAS Institute,1999, SAS/STAT User's Guide, Version 8, SAS Institute, Cary, NC

ACKNOWLEDGEMENTS

The author would like to express his appreciation to the following:

Stephen Butler, MKIS Director, Cingular™ Wireless, LLC for making the data available and allowing these results to be presented.

Danny Black, Cingular Wireless, LLC. for proofreading the draft.

CONTACT INFORMATION:

Charles.manahan@cingular.com
Charles Manahan
Cingular Wireless, LLC.
Glenridge Highlands Two
5565 Glenridge Connector
Atlanta, GA 30342