

Improving the Performance of Data Mining Models with Data Preparation Using SAS® Enterprise Miner

Ricardo Galante, SAS Institute Brasil, São Paulo, SP

ABSTRACT

In data mining modelling, data preparation is the most crucial, most difficult, and longest part of the mining process. A lot of steps are involved. Consider the simple distribution analysis of the variables, the diagnosis and reduction of the influence of variables' multicollinearity, the imputation of missing values, and the construction of categories in variables. In this presentation, we use data mining models in different areas like marketing, insurance, retail and credit risk. We show how to implement data preparation through SAS® Enterprise Miner™, using different approaches. We use simple code routines and complex processes involving statistical insights, cluster variables, transform variables, graphical analysis, decision trees, and more.

INTRODUCTION

The process of knowledge discovery in data mining involves three main parts: data preprocessing, data selection and data information.

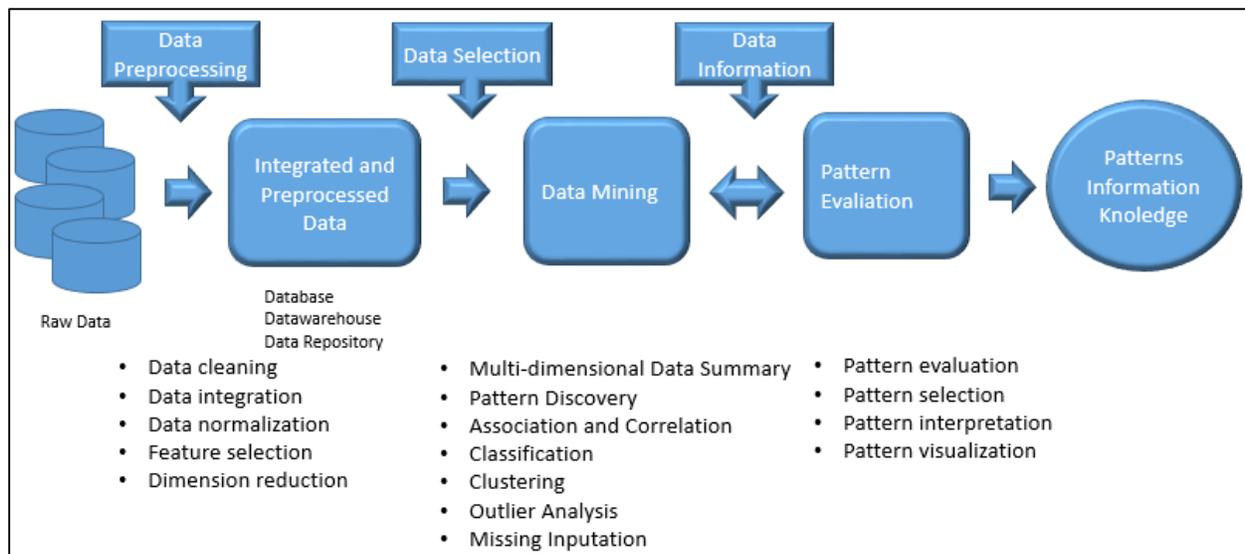


Figure 1: Process of Knowledge Discovery

This article focus on data selection and covers techniques used in analytical data preparation. Rather than addressing the ETL (Extraction, Transform and Load) process, our goal is to show the fundamental steps involved in preparing and selecting context relevant variables. This will ensure an improved data mining process as a consequence of a less polluted set of variables. It will prevent problems, such as multicollinearity or missing values. In addition, processing time will be shortened.

The success of predictive models largely depends on input selection. Our goal is to show some techniques to reduce the input count without compromising the quality of the final model.

The data set used in this article is called t97NK. This data set was taken and adapted from the Association for Computing Machinery's (ACM) 1998 KDD - Cup competition. Details of the set and the competition are publicly available at the UCI Archive at <http://kdd.ics.uci.edu>. This data set has 49 variables and 251842 rows, a binary Target.

USING DATA SOURCE CREATION

A data source is a link between an existing SAS Table and the project in SAS Enterprise Miner.

Figure 2 shows how to use the Metadata Advisor Options to improve variable selection, during the creation of data source in data mining modelling.

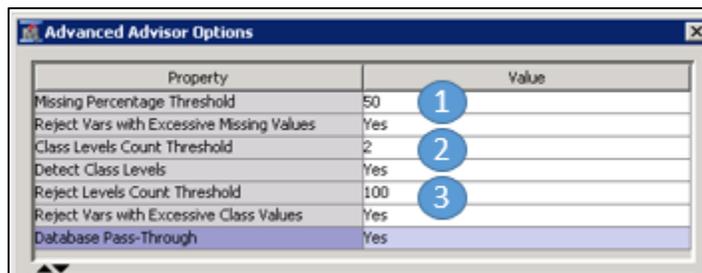


Figure 2: Advanced Advisor Options

On the Advanced Advisor Options panel we have:

- 1) Reject variables with excessive number of missing values. The default here is 50%. At first, variables missing more than 50% of their values are not taken into account.
- 2) Detect the number of levels of Numeric variables and assign Nominal role to those with class count below the selected threshold. The default is 20, but in this case I chose 2 as threshold.
- 3) Detect the number of levels of Character variables and assign a rejected role to those with class count above the selected threshold. The default is also 20, but I chose 100.

This option helps to eliminate unwanted variables.

Figure 3 shows the group of variables considering the settings chosen on the Advanced advisor Options panel.

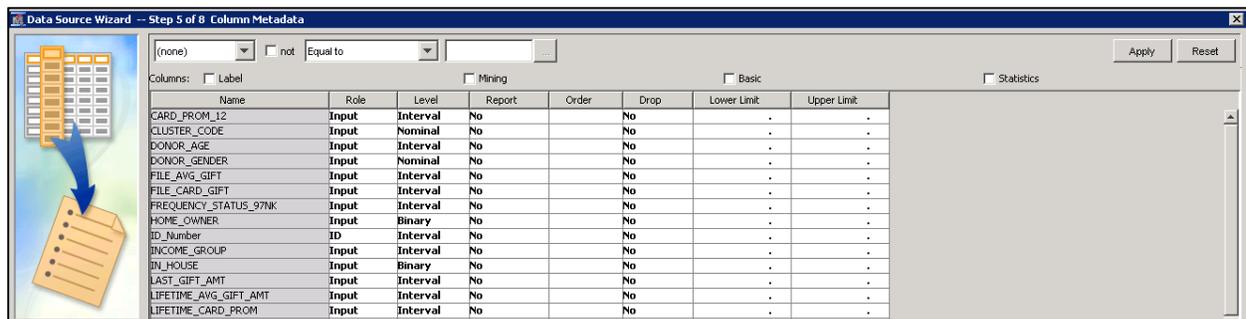


Figure 3: Data Source Wizard

Setting the Advanced Advisory Option allows you to see some descriptive statistics, such as Number of Levels, Minimum, Maximum, Mean, Standard Deviation, Skewness and Kurtosis, by checking the Statistics box as shown in Figure 4.

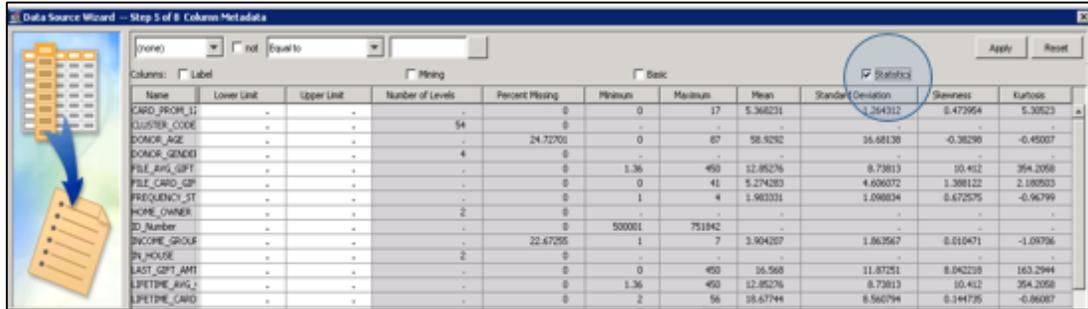


Figure 4: Statistics Option in Data Source Wizard

STATEXPLORE NODE: USING METRICS AND GRAPHS TO EXPLORE DATA

The StatExplore node is a multipurpose tool that you use to examine variable distributions and statistics in your data sets. The StatExplore node generates summarization statistics. In this section we use the StatExplore node to find relevant information about the variables.

Figure 4 shows the StatExplore node connected at the data source T97Nk.

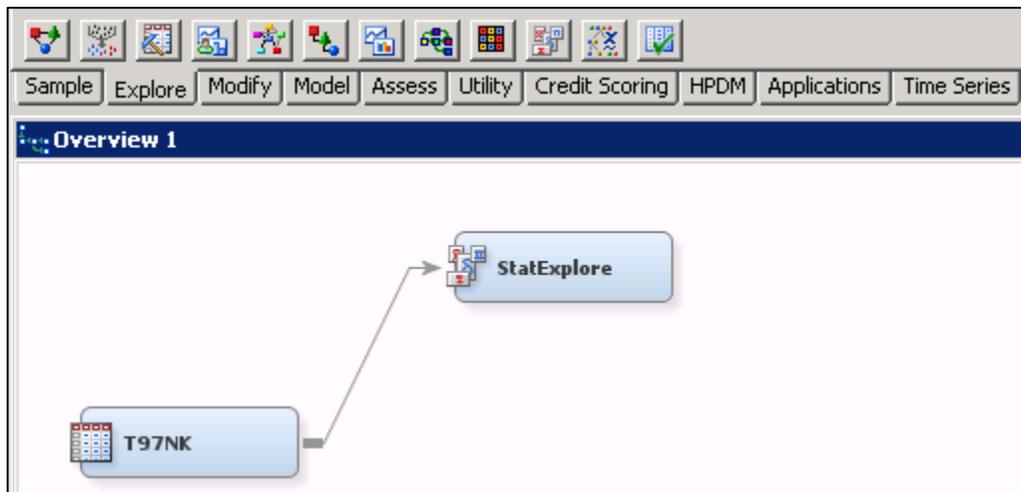


Figure 4: Using the StatExplore node in the flow

Select the StatExplore node and examine its Properties Panel. To have the Chi-Square between all variables and the Target change the option Interval Variables below Chi-Square Statistics to Yes as shown in Figure 5.

As you know, the Chi-Square test is used for categorical variables. One way to include interval variables in this test is doing as mentioned above. This will generate chi-square statistics including the interval variables by binning the type of variables. The default is 5 bins.

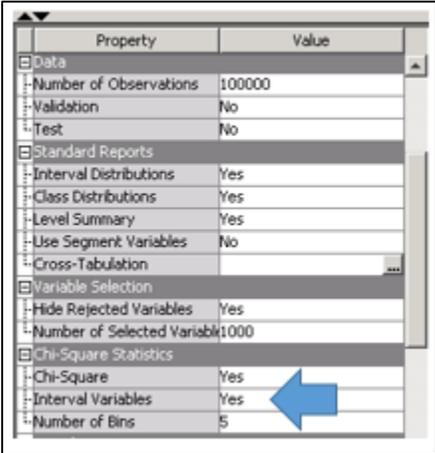


Figure 5: Panel Properties to StatExplore Node

Figure 6 shows the chi-square test including the “transformed” interval variables.

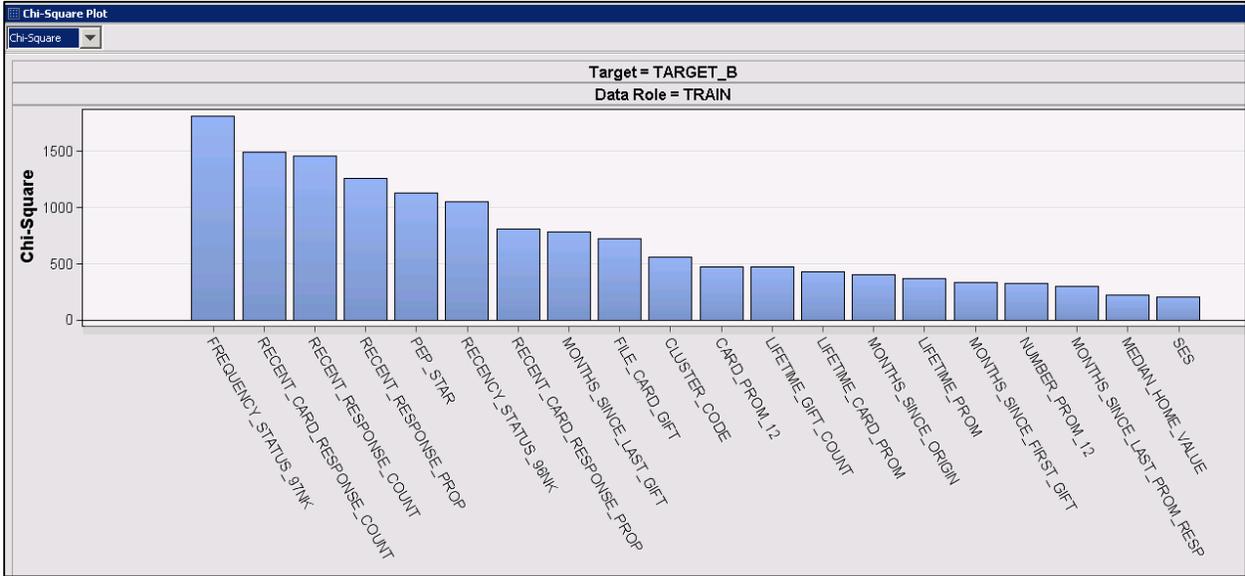


Figure 6: Chi_Square Plot Results

The chart shows the association between each variable and the Target in decreasing order of association.

If you want to see these “created” bins, go to the results part and select View > Summary Statistics > Cell Chi-Squares. Figure 7 shows the bins to the variable Card_Prom_12.

Input	Target: Formatted Value	Input: Formatted Value	Frequency Count	Target: Numeric Value	Input: Numeric Value	Chi-Square
CARD_PROM_12	0	10.2 -13.6	433	0	11	47.23084
CARD_PROM_12	0	13.6 - HIGH	4	0	14	13.06633
CARD_PROM_12	0	3.4 -6.8	67962	0	4	6.462228
CARD_PROM_12	0	6.8 -10.2	2336	0	7	47.72877
CARD_PROM_12	0	LOW-3.4	4366	0	0	3.003646
CARD_PROM_12	1	10.2 -13.6	368	1	11	142.4589
CARD_PROM_12	1	13.6 - HIGH	23	1	14	39.411
CARD_PROM_12	1	3.4 -6.8	21654	1	4	19.49154
CARD_PROM_12	1	6.8 -10.2	1252	1	7	143.9607
CARD_PROM_12	1	LOW-3.4	1602	1	0	9.059672

Figure 7: Card_Prom_12 Bins

Figure 8 shows the results of each Chi-Square test with all the variables.

Data Role=TRAIN Target=TARGET_B			
Input	Chi-Square	Df	Prob
RECENT_RESPONSE_COUNT	524.0171	17	<.0001
RECENT_CARD_RESPONSE_COUNT	488.8189	9	<.0001
FREQUENCY_STATUS_97NK	460.6909	3	<.0001
PEP_STAR	361.9954	1	<.0001
REGENCY_STATUS_96NK	231.8164	5	<.0001
WEALTH_RATING	70.5646	10	<.0001
SES	58.6349	4	<.0001
INCOME_GROUP	29.4664	7	0.0001
URBANICITY	15.8314	5	0.0073
HOME_OWNER	12.9099	1	0.0003
DONOR_GENDER	5.2304	2	0.0732
OVERLAY_SOURCE	4.1391	3	0.2468
IN_HOUSE	3.9921	1	0.0457
PUBLISHED_PHONE	2.0506	1	0.1521

Figure 8: Chi_Square Results

Figure 9 shows the Scale Mean Deviation plot. This plot shows the scale mean deviation between the overall mean of an interval input variable and its mean for each level of the target variable. Each variable is represented vertically by two squares, one blue and the other red. The blue is the scale mean deviation of the variable when the target is 0 and the red is the scale mean deviation of the same variable when the target is 1.

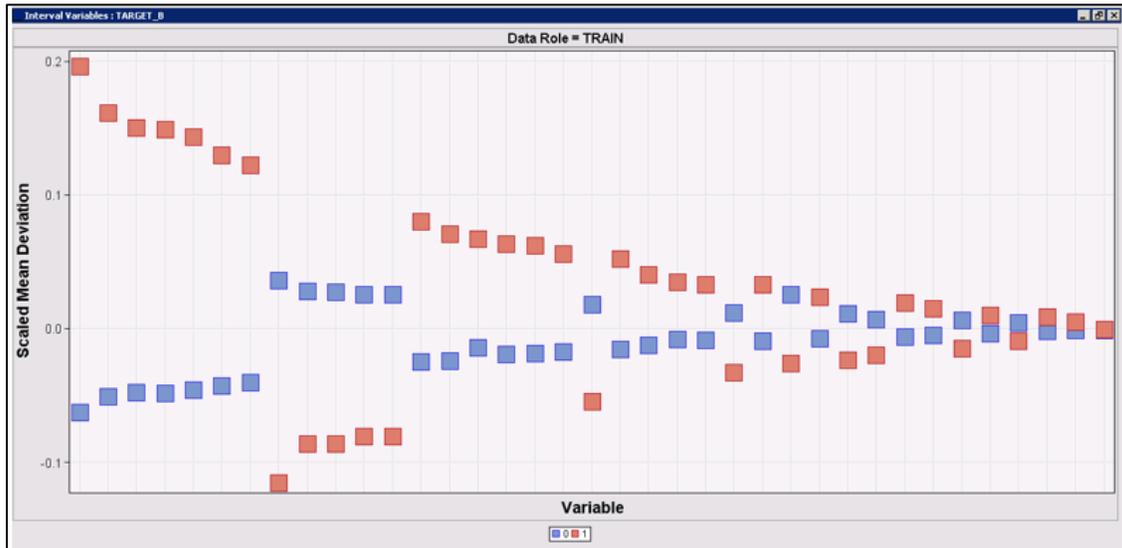


Figure 9: Scale Mean Deviation Plot

For example, let's assume there is a class target with level 0 and 1 and that there is an interval variable called x , the scaled mean deviation is calculated as follows:

$$\text{SMD}(x, 0) = [\text{mean}(x, \text{ where target}=0) - \text{mean}(x)] / \text{mean}(x)$$

$$\text{SMD}(x, 1) = [\text{mean}(x, \text{ where target}=1) - \text{mean}(x)] / \text{mean}(x)$$

The results from the StatExplore node help us find the best set of variables which could be used in the process of data mining.

TRANSFORM VARIABLES NODE

The Transform Variables node allows the transformation of all variables. For interval variables, simple transformations, binning transformations and best transformations are available. For class variables, the grouping of rare levels and the creation of dummy indicators are possible. In general, with Transform Variables node, you can do both computed transformations and customized actions choosing the best transformation based on your choice.

Let's take a look at three very common automatic transformations.

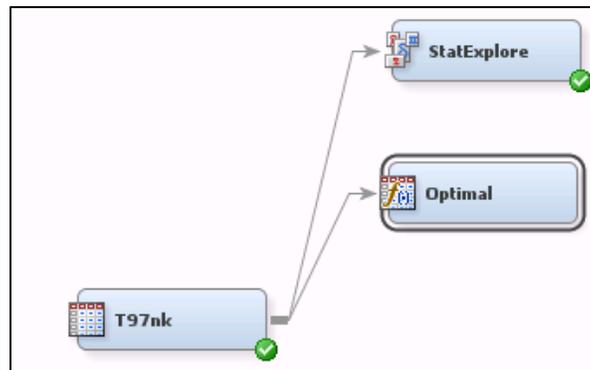


Figure 10: Transform Variable Node

Figure 10 shows the Transform Variable node connected to the data source. For didactic purposes I renamed the node to Optimal.

Default Methods	
Interval Inputs	Optimal Binning
Interval Targets	None
Class Inputs	None
Class Targets	None
Treat Missing as LeNo	

Figure 11: Transform Variable Properties Panel

Figure 11 shows the Default Methods including the Optimal Binning for interval inputs. The optimal binning allows you to transform continuous variables into an ordered set of bins. The binning is done so that the log odds of the predicted categorical variable is monotonically increasing or decreasing. The optimal binning transformation is useful when there is a nonlinear relationship between an input variable and the target.

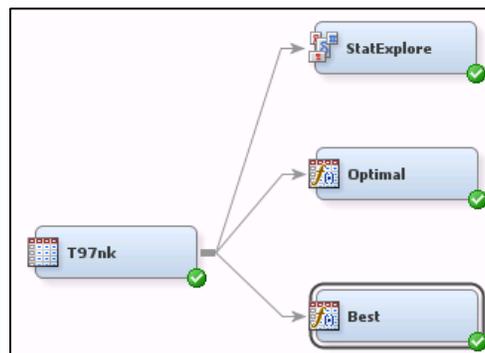


Figure 12: Transform Variable Node

Figure 12 shows the Transform Variable node connected to the data source that I renamed to Best.

Default Methods	
Interval Inputs	Best
Interval Targets	None
Class Inputs	None
Class Targets	None
Treat Missing as	LeNo

Figure 13: Transform Variable Properties Panel

Figure 13 shows the Default Methods including the Best option for interval inputs. The Best option perform several transformations and uses the transformation that has the best association with the target variable considering the Chi Squared test.

Input Name	Role	Input Level	Name	Level	Formula
CARD_PROM_12	INPUT	INTERVAL	OPT_CARD_PROM_12	NOMINAL	Optimal Binning(4)
DONOR_AGE	INPUT	INTERVAL	LOG_DONOR_AGE	INTERVAL	$\log(\text{DONOR_AGE} + 1)$
FILE_AVG_GIFT	INPUT	INTERVAL	OPT_FILE_AVG_GIFT	NOMINAL	Optimal Binning(4)
FILE_CARD_GIFT	INPUT	INTERVAL	OPT_FILE_CARD_GIFT	NOMINAL	Optimal Binning(4)
FREQUENCY_STATUS_97MK	INPUT	INTERVAL	OPT_FREQUENCY_STATUS_97MK	NOMINAL	Optimal Binning(4)
INCOME_GROUP	INPUT	INTERVAL	INV_INCOME_GROUP	INTERVAL	$1 / (\text{INCOME_GROUP} + 1)$
LAST_GIFT_AMT	INPUT	INTERVAL	OPT_LAST_GIFT_AMT	NOMINAL	Optimal Binning(4)
LIFETIME_AVG_GIFT_AMT	INPUT	INTERVAL	OPT_LIFETIME_AVG_GIFT_AMT	NOMINAL	Optimal Binning(4)
LIFETIME_CARD_PROM	INPUT	INTERVAL	LG10_LIFETIME_CARD_PROM	INTERVAL	$\log_{10}(\text{LIFETIME_CARD_PROM} + 1)$
LIFETIME_GIFT_AMOUNT	INPUT	INTERVAL	INV_LIFETIME_GIFT_AMOUNT	INTERVAL	$1 / (\text{LIFETIME_GIFT_AMOUNT} + 1)$
LIFETIME_GIFT_COUNT	INPUT	INTERVAL	LG10_LIFETIME_GIFT_COUNT	INTERVAL	$\log_{10}(\text{LIFETIME_GIFT_COUNT} + 1)$
LIFETIME_GIFT_RANGE	INPUT	INTERVAL	OPT_LIFETIME_GIFT_RANGE	NOMINAL	Optimal Binning(4)
LIFETIME_MAX_GIFT_AMT	INPUT	INTERVAL	INV_LIFETIME_MAX_GIFT_AMT	INTERVAL	$1 / (\text{LIFETIME_MAX_GIFT_AMT} + 1)$
LIFETIME_MIN_GIFT_AMT	INPUT	INTERVAL	OPT_LIFETIME_MIN_GIFT_AMT	NOMINAL	Optimal Binning(4)
LIFETIME_PROM	INPUT	INTERVAL	OPT_LIFETIME_PROM	NOMINAL	Optimal Binning(4)
MEDIAN_HOME_VALUE	INPUT	INTERVAL	SQRT_MEDIAN_HOME_VALUE	INTERVAL	$\text{Sqrt}(\text{MEDIAN_HOME_VALUE} + 1)$
MEDIAN_HOUSEHOLD_INCOME	INPUT	INTERVAL	SQRT_MEDIAN_HOUSEHOLD_INCOME	INTERVAL	$\text{Sqrt}(\text{MEDIAN_HOUSEHOLD_INCOME} + 1)$
MONTHS_SINCE_FIRST_GIFT	INPUT	INTERVAL	OPT_MONTHS_SINCE_FIRST_GIFT	NOMINAL	Optimal Binning(4)
MONTHS_SINCE_LAST_GIFT	INPUT	INTERVAL	SQRT_MONTHS_SINCE_LAST_GIFT	INTERVAL	$\text{Sqrt}(\text{MONTHS_SINCE_LAST_GIFT} + 1)$
MONTHS SINCE LAST PROM RESP	INPUT	INTERVAL	OPT_MONTHS SINCE LAST PROM RESP	NOMINAL	Optimal Binning(4)

Figure 14: Results of Computed Transformations

Figure 14 shows some results after we select the Best option.

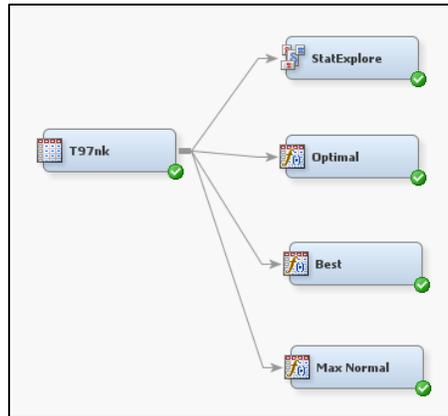


Figure 15: Transform Variable Node

Figure 15 shows the Transform Variable node connected to the data source that I renamed to Max Normal. In this case, in Default Methods I changed from none to Maximum Normal as shown in Figure 16.

Default Methods	
Interval Inputs	Maximum Normal
Interval Targets	None
Class Inputs	None
Class Targets	None
Treat Missing as Level	No

Figure 16: Transform Variable Properties Panel

Computed Transformations (maximum 500 observations printed)						
Input Name	Role	Input Level	Name	Level	Formula	
CARD_PROM_12	INPUT	INTERVAL	SQRT_CARD_PROM_12	INTERVAL	$\sqrt{\max(\text{CARD_PROM_12}-0, 0.0)/17}$	
DONOR_AGE	INPUT	INTERVAL	PWR_DONOR_AGE	INTERVAL	$(\max(\text{DONOR_AGE}-0, 0.0)/87)**4$	
FILE_AVG_GIFT	INPUT	INTERVAL	LOG_FILE_AVG_GIFT	INTERVAL	$\log(\max(\text{FILE_AVG_GIFT}-1.36, 0.0)/448.64 + 1)$	
FILE_CARD_GIFT	INPUT	INTERVAL	LOG_FILE_CARD_GIFT	INTERVAL	$\log(\max(\text{FILE_CARD_GIFT}-0, 0.0)/41 + 1)$	
FREQUENCY_STATUS_97NK	INPUT	INTERVAL	SQRT_FREQUENCY_STATUS_97NK	INTERVAL	$\sqrt{\max(\text{FREQUENCY_STATUS_97NK}-1, 0.0)/3}$	
LAST_GIFT_AMT	INPUT	INTERVAL	LOG_LAST_GIFT_AMT	INTERVAL	$\log(\max(\text{LAST_GIFT_AMT}-0, 0.0)/450 + 1)$	
LIFETIME_AVG_GIFT_AMT	INPUT	INTERVAL	LOG_LIFETIME_AVG_GIFT_AMT	INTERVAL	$\log(\max(\text{LIFETIME_AVG_GIFT_AMT}-1.36, 0.0)/448.64 + 1)$	
LIFETIME_CARD_PROM	INPUT	INTERVAL	SQRT_LIFETIME_CARD_PROM	INTERVAL	$\sqrt{\max(\text{LIFETIME_CARD_PROM}-2, 0.0)/54}$	
LIFETIME_GIFT_AMOUNT	INPUT	INTERVAL	LOG_LIFETIME_GIFT_AMOUNT	INTERVAL	$\log(\max(\text{LIFETIME_GIFT_AMOUNT}-15, 0.0)/3760 + 1)$	
LIFETIME_GIFT_COUNT	INPUT	INTERVAL	SQRT_LIFETIME_GIFT_COUNT	INTERVAL	$\sqrt{\max(\text{LIFETIME_GIFT_COUNT}-1, 0.0)/94}$	
LIFETIME_GIFT_RANGE	INPUT	INTERVAL	LOG_LIFETIME_GIFT_RANGE	INTERVAL	$\log(\max(\text{LIFETIME_GIFT_RANGE}-0, 0.0)/997 + 1)$	
LIFETIME_MAX_GIFT_AMT	INPUT	INTERVAL	LOG_LIFETIME_MAX_GIFT_AMT	INTERVAL	$\log(\max(\text{LIFETIME_MAX_GIFT_AMT}-5, 0.0)/995 + 1)$	
LIFETIME_MIN_GIFT_AMT	INPUT	INTERVAL	LOG_LIFETIME_MIN_GIFT_AMT	INTERVAL	$\log(\max(\text{LIFETIME_MIN_GIFT_AMT}-0, 0.0)/450 + 1)$	

Figure 17: Results of Computed Transformations

Figure 17 shows the results. When the option Maximum Normal is chosen, the node will look for the best transformation to maximize the normality of the variable.

In addition to the computed transformation options, the Transform Variable node allows us to do customized transformations when selecting the Formulas option, as shown in Figure 18.

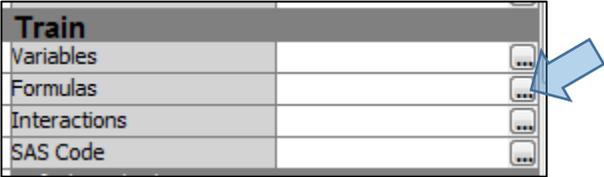


Figure 18: Properties Panel

This option shows graphics with the distribution of each variable as shown in Figure 19.

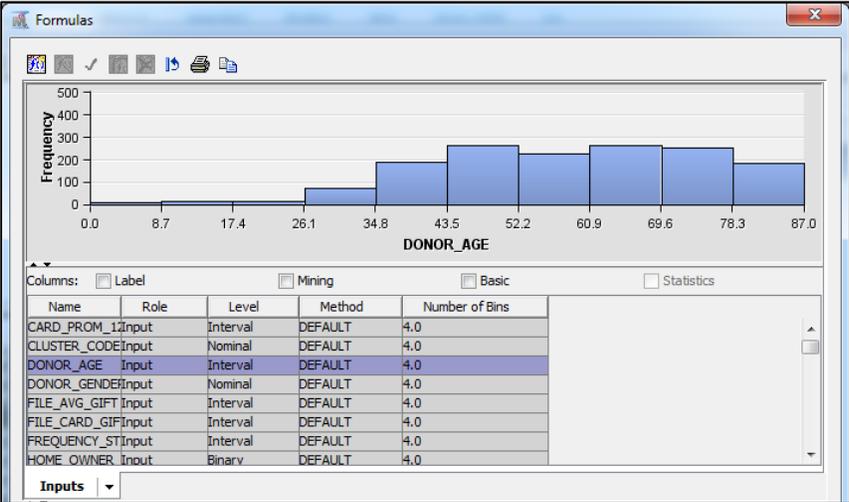


Figure 19: Distribution of Each Variable

In addition to providing the distribution of each variable graphically, if you select the Expression Builder from the options Create > Build, you can do customized transformations based on your requirement as shown in Figure 20.

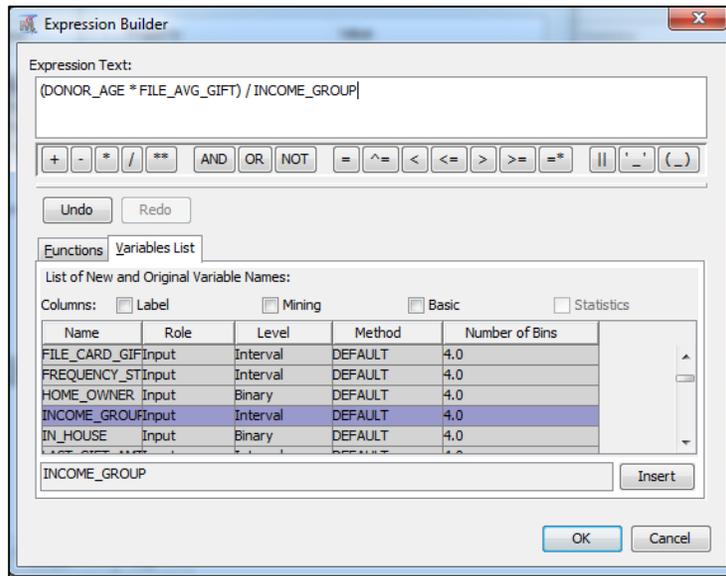


Figure 20: Expression Builder

INTERACTIVE BINNING NODE

This node groups the values of each variable, both class and interval, in bins or buckets to be used in predictive modeling as a way to improve the predictive power of each input. In addition, this node can be used as a variable selection that applies Gini statistic as metric. This metric shows the predictive power of a characteristic, for example ability to separate high-risk applicants from low-risk ones. Through this you can configure the node so that the missing values are treated as a level for each variable.

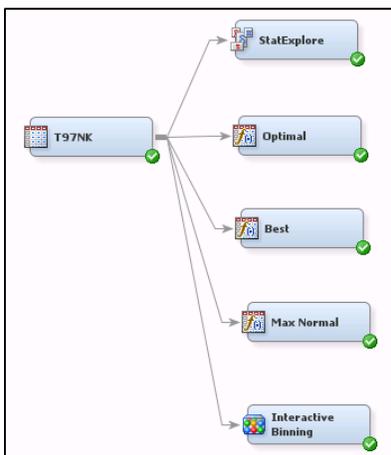


Figure 21: Interactive Binning Node

Figure 21 shows the Interactive Binning node results. In it, both rejected and approved variables are arranged after going through Gini statistics.

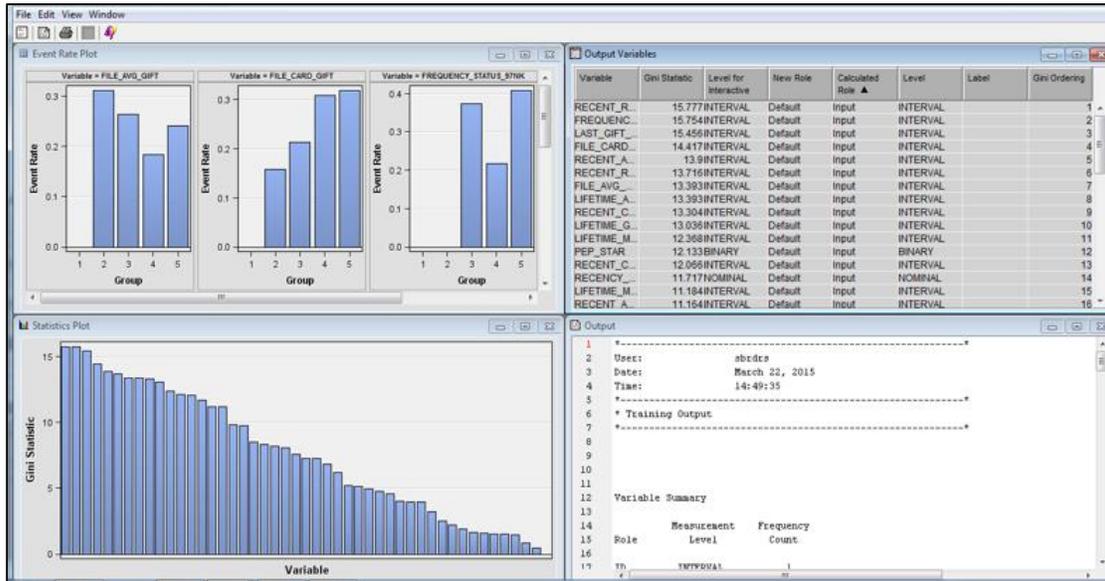


Figure 22: Interactive Binning Results

Figure 22 shows the results from Interactive Binning node results. These results show which variables were rejected and which variables were approved regarding the results from Gini statistics.

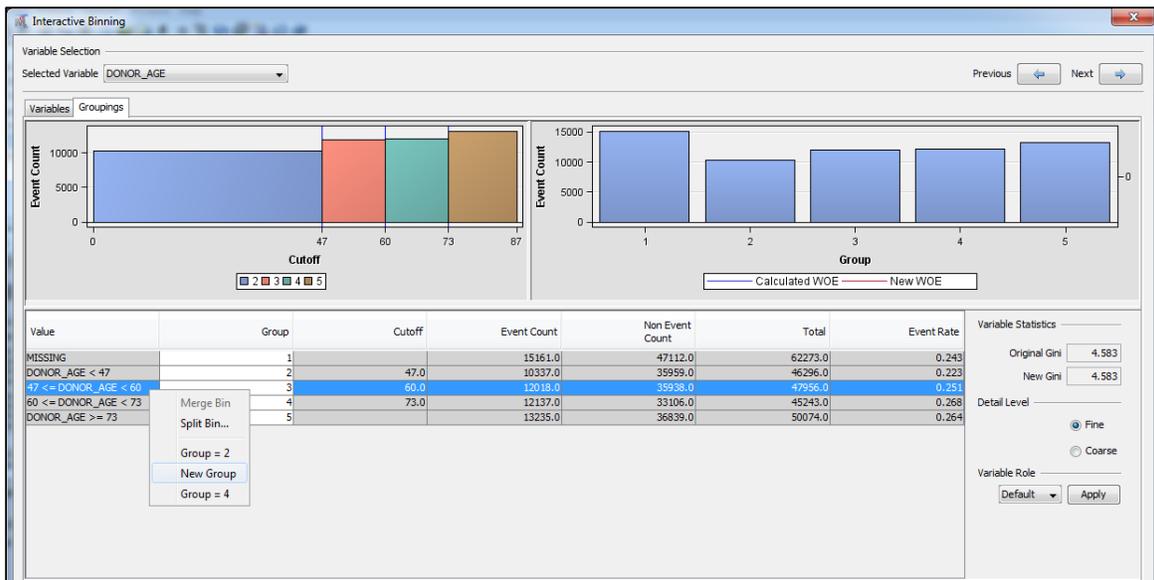


Figure 23: Interactive Binning Options

If you want to change the ranges or the groups, this node allows us to interact with the results found, as shown in Figure 23.

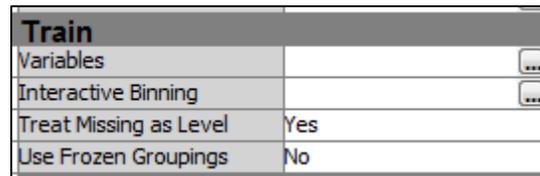


Figure 24: Interactive Binning Panel

This can be done by selecting the square in Interactive Binning panel as in Figure 24.

IMPUTE NODE: IMPUTING MISSING VALUES

In the processing of Data Mining, databases often contain observations that have missing values for one or more variables. Missing values can result from data collection errors, incomplete customer responses, actual system and measurement failures, or from a revision of the data collection scope over time, such as tracking new variables that were not included in the previous data collection scheme.

In SAS Enterprise Miner, models such as regressions and neural networks ignore the whole record that contains missing values. This reduces the size of the training data set. Less training data can substantially weaken the predictive power of these models. To overcome this obstacle of missing data, you can impute missing values before you fit the models.

How should missing data values be treated? There is no single correct answer. Choosing the "best" missing value replacement technique inherently requires the researcher to make assumptions about the true (missing) data. For example, researchers often replace a missing value with the mean of the variable. This approach assumes that the variable's data distribution follows a normal population response. Replacing missing values with the mean, median, or another measure of central tendency is simple, but it can greatly affect a variable's sample distribution. You should use these replacement statistics carefully and only when the effect is minimal.

The Imputation Node does impute a value based on various metrics like mean, mode, or median as well as Decision Trees (with or without Surrogates) to capture the essence of the variable with missing entries using other variables in the data set.

Let's take a look in some way to do the imputation values through Impute node. Figure 25 shows the Impute node connected on the Transformation node renamed as Default.

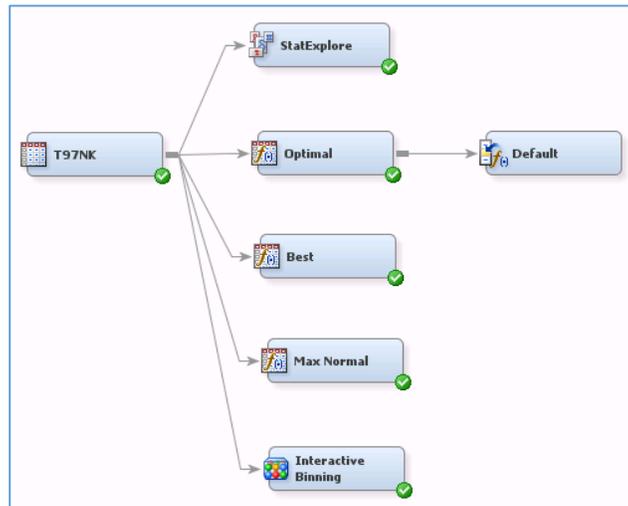


Figure 25: Impute Node

Class Variables	
Default Input Method	Count
Default Target Method	None
Normalize Values	Yes
Interval Variables	
Default Input Method	Mean
Default Target Method	None

Figure 26: Impute Node Panel Properties

Figure 26 shows the default option in the panel properties. If the variable has an interval role then the missing values will be replaced by the average of the variable. On the other hand, if the variable has a class role then the missing values will be replaced by the count or mode of the variable.

Because a predicted response might be different for cases with a missing input value, a binary imputation indicator variable is often added to the training data. Adding this variable enables a model to adjust its predictions in cases where “missingness” itself is correlated with the target.

Score	
Hide Original Variables	Yes
Indicator Variables	
Type	Single
Source	Imputed Variables
Role	Rejected

Figure 27: Indicator Variables in Impute Node Panel Properties

Figure 27 shows where to choose this option in Impute Properties Panel.

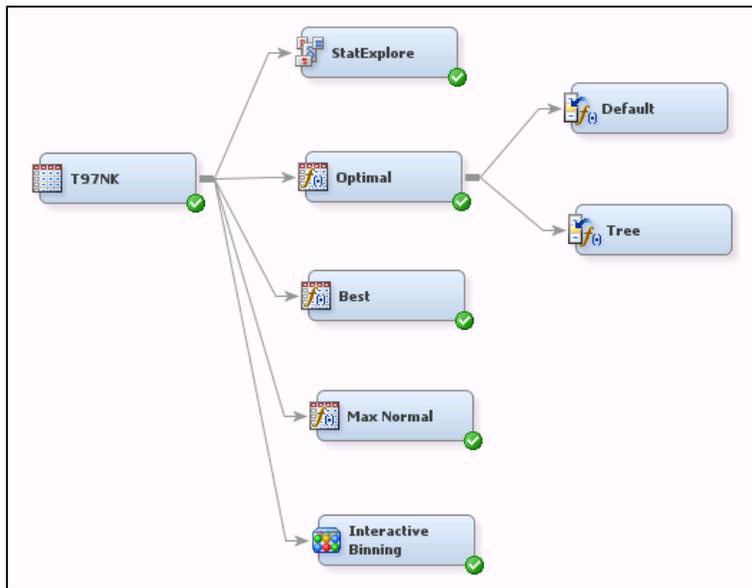


Figure 28: Indicator Variables in Impute Node Panel Properties

Figure 28 shows the Impute node connected on the Transformation node renamed as Tree. On Properties Panel I replace both Default Input Method Class and Interval variables to Tree.

Using this option the replacement values are estimated by analyzing each input as a target, and the remaining input and rejected variables are used as predictors. Because the imputed value for each input variable is based on the other input variables, this imputation technique may be more accurate than simply using the variable mean or median to replace the missing tree values.

PRINCIPAL COMPONENT NODE

The Principal Component node calculates eigenvalues and eigenvectors from the uncorrected covariance matrix, corrected covariance matrix or the correlation matrix of input variables. Principal components are calculated from the eigenvectors and are usually treated as the new set of input variables for successor nodes. This approach is useful for data interpretation and data dimension reduction.

Figure 29 shows the Principal Components node connected with the previous nodes.

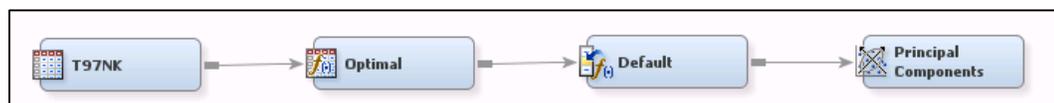


Figure 29: Principal Components

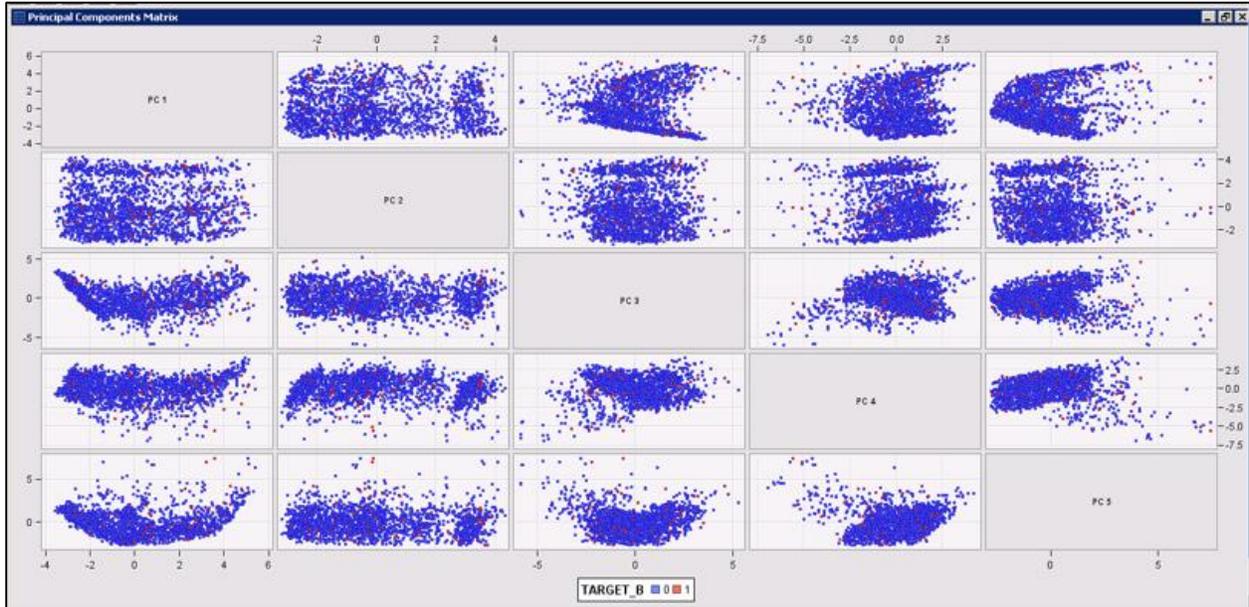


Figure 30: Principal Components Matrix

The Principal Components Matrix chart displays scattered plots for all pairs of the selected principal components. By default, the initial matrix chart shows the scattered plots for the first five principal components considering the Target variable as in Figure 30.

On the Eigenvalue Plot, you can see some useful metrics like eigenvalues, proportional eigenvalues, cumulative proportional eigenvalue and log eigenvalue. You can select an item from the drop-down menu to change the values.



Figure 31: Eigenvalue Plot

Figure 31 shows the Eigenvalue Plot with values. The vertical reference line indicates the number of principal components that will be used in the subsequent node. In this case we have 14 principal components.

VARIABLE CLUSTERING NODE

The Variable Clustering node is useful for data reduction, such as choosing the best variables or cluster components for analysis purposes. Variable Clustering removes collinearity, decreases variable redundancy, and helps to reveal the underlying structure of input variable in a data set.

You can use both Correlation Matrix and Covariance Matrix as source of information to build the variables clusters. Using the Correlation Matrix, the SAS Enterprise Miner uses the correlation matrix of standardized variables. When the Cluster Component property is set to Principal Components, the source matrix provides eigenvalues. Using the Covariance Matrix the SAS Enterprise Miner uses the covariance matrix of raw variables. The Covariance property setting permits variables that have a large variance to have more effect on the cluster components than variables that have a small variance. The Covariance property setting permits variables that have a large variance to have more effect on the cluster components than variables that have a small variance. When the Cluster Component property is set to Principal Components, the source matrix provides eigenvectors.

Figure 32 shows the Variable Clustering node connected with the previous nodes.

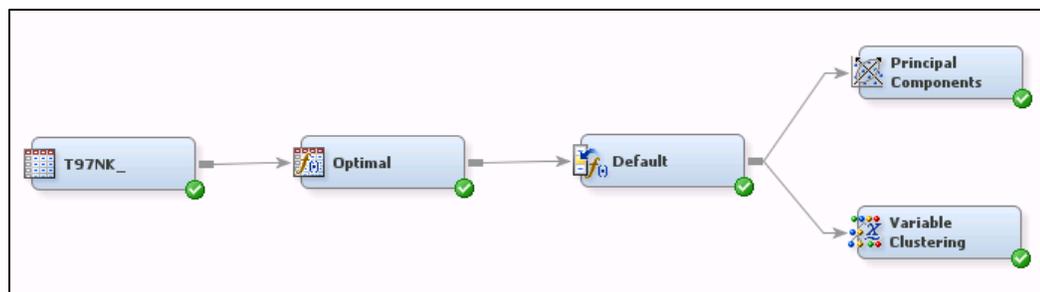


Figure 32: Variable Clustering

The Results window dendrogram uses a tree hierarchy to display how the clusters were formed. Figure 33 shows the dendrogram.

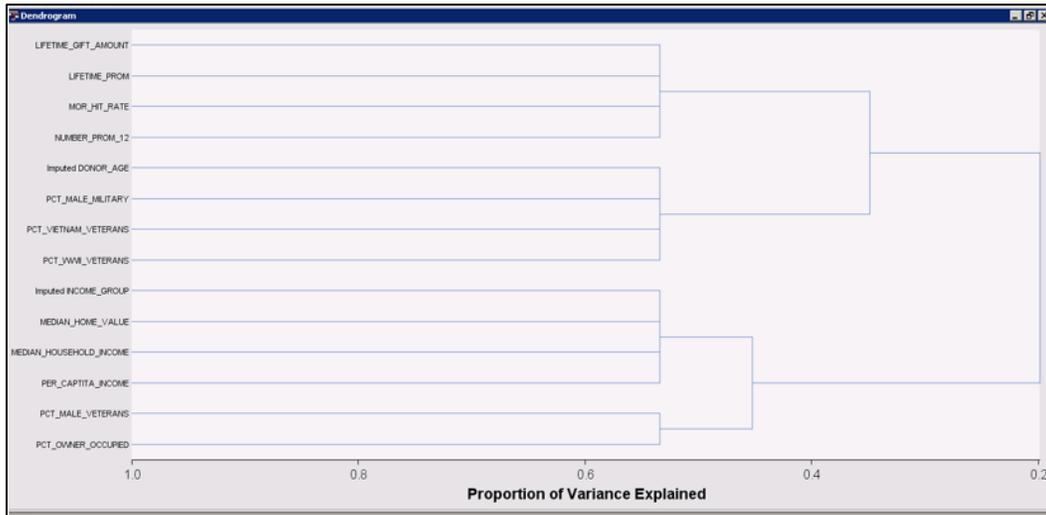


Figure 33: Dendrogram

Figure 34 shows the Cluster Plot with four clusters that were created from the set of interval input variables in the data source.

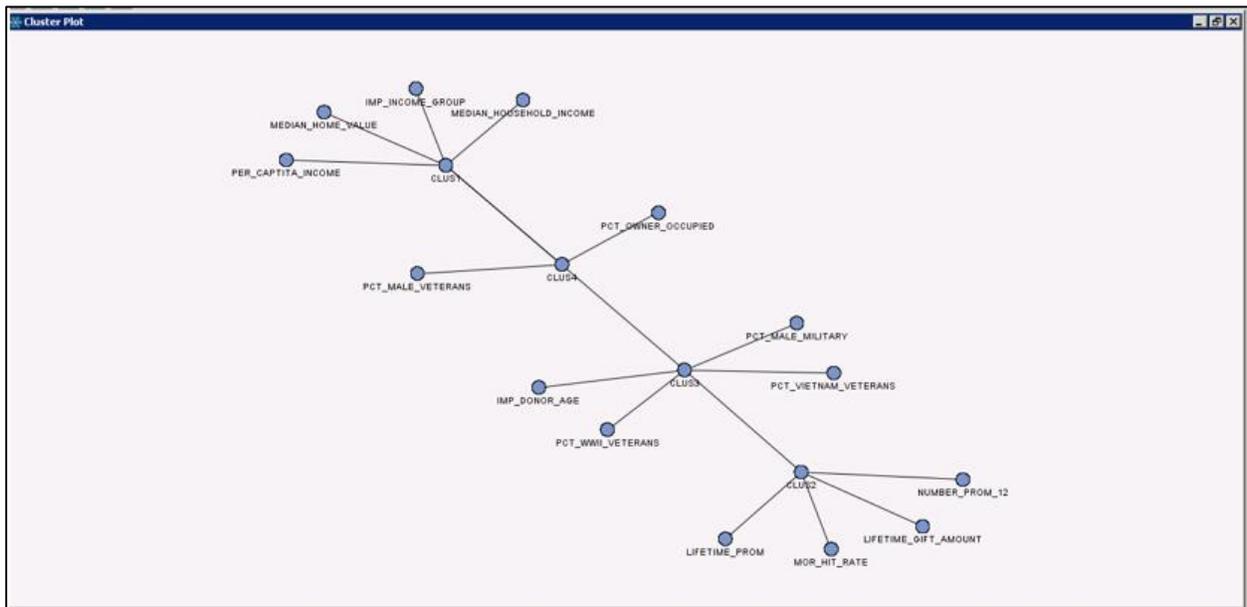


Figure 34: Cluster Plot

The components in the Variable Selection table will be exported to the node that follows the Variable Clustering node in a process flow diagram. The R-Square with Next Cluster Component column of the table indicates the R-square scores with the nearest cluster. If the clusters are well separated, R-square score values should be low. Small values in the $1-R^2$ Ratio column of the Variable Selection table also indicate good clustering.

VARIABLE SELECTION NODE

The Variable Selection node enables you to evaluate the importance of input variables in predicting or classifying the target variable. To select the important inputs, the tool uses either an R-Squared or a Chi-Squared selection criterion.

When the option Default is used this the uses target and model information to choose the selection method. If the target is binary and the model has greater than 400 degrees of freedom, the Chi-Square method is selected. Otherwise, the R-Square method is used.

The R-Square method can be used with a binary as well as with an interval-scaled target. In the R-Square method, variable selection is performed in two steps. In the first step R-Square between the input and the target is calculated. All variables with a correlation above a specified threshold are selected in the first step. Those variables which are selected in the first step enter the second step of variable selection. In the second step, a sequential forward selection process is used. This process starts by selecting the input variable that has the highest correlation coefficient with the target. A regression equation is estimated with the selected input.

The Chi-Square method can be used when the target is binary. When this criterion is selected, the selection process does not have two distinct steps, as in the case of the R-square criterion. Instead, a tree is constructed. The inputs selected in the construction of the tree are passed to the next node with the assigned Role of Input.

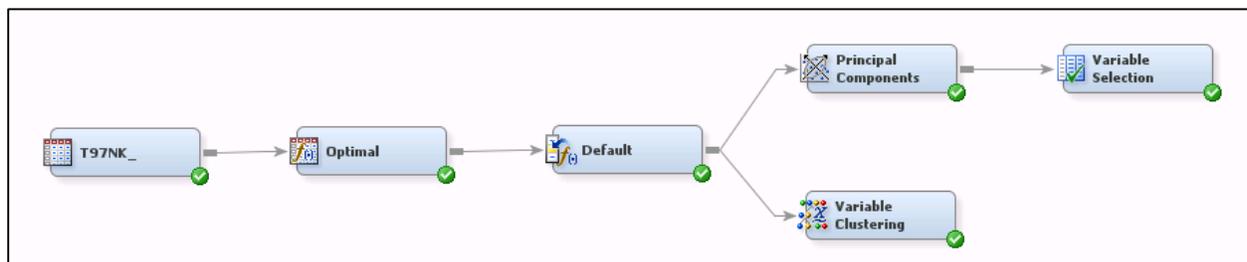


Figure 35: Variable Selection

Figure 35 shows the flow using the Variable Selection node with default options.

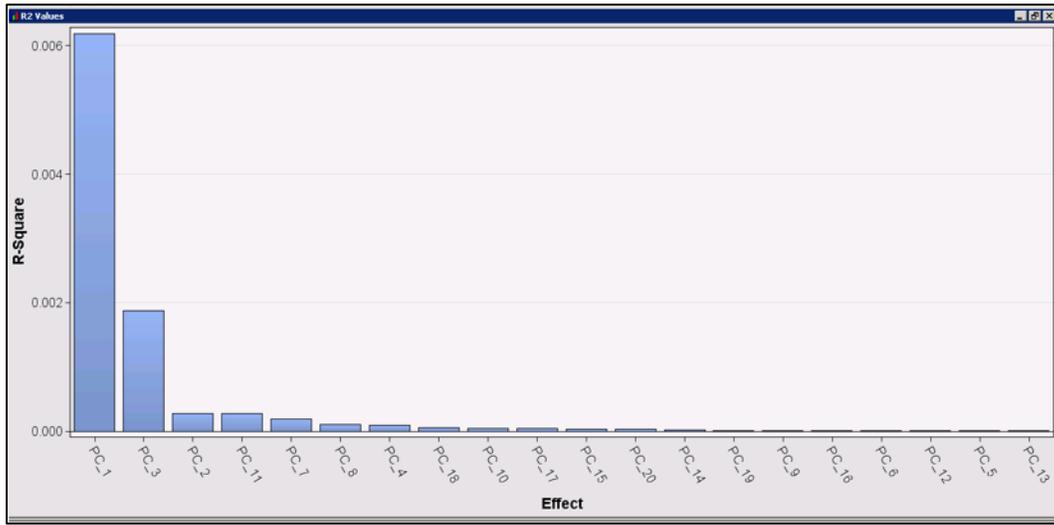


Figure 36: R² Values

Figure 36 shows the R² Values Plot. This plot displays a histogram with ranked variable effects. The ranking uses the variable R², sorted from highest to lowest.

CONCLUSION

This paper shows how you can use the SAS Enterprise Miner 13.2 to the process of preparing data for analytical modeling. My goal here is to show some ways to do this. These approaches should be used as a reference and not as a road map to follow.

ACKNOWLEDGMENTS

The author thanks Djalma de Azevedo for the input and help in reviewing the text.

REFERENCES

Kattamuri S. Sarma. Predictive Modeling with SAS® Enterprise Miner™: Practical Solutions for Business Applications, Second Edition.

Patricia B. Cerrito. Introduction to Data Mining Using SAS Enterprise Miner.

SAS Enterprise Miner 13.2: Reference Help.

CONTACT INFORMATION

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

Ricardo Galante.

SAS

SAS Institute Brasil

9º andar - R. Leopoldo Couto Magalhães Júnior, 700 - Itaim Bibi, São Paulo - SP, 04542-000

55 11 4501-5300

Ricardo.Galante@sas.com

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