Credit Risk Dataset

Overview of the Predictive Modeling Case:

Under this project, I have analyzed the CREDIT dataset which has loan information about customers with details such as the loan amount in last 24 months, bankruptcy indicator, public derogatory etc. My goal on this project is to use SAS Enterprise Miner to build and compare predictive models and choose the best fitted model that can predict whether an applicant will default on loan or not. This kind of a model can be used in real applications to identify customers who can be offered better interest rates and more credit facilities. There are primarily three phases under this project.

Phase 1: Data Exploration and Study of Data Distribution

Phase 2: Prediction modeling (Decision Tree, Regression and Neural Network). I have used Neural Network mainly for comparison purposes. If a neural network performs significantly better than regression model, then this will indicate the lack of fit of regression and will require more analysis and fixes to improve the regression model fit.

Phase 3: Model Comparison and Model Selection

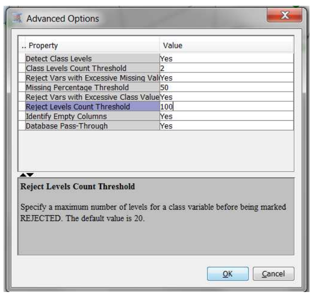
Table of Contents

Phase 1: Data Exploration and Study of Data Distributions

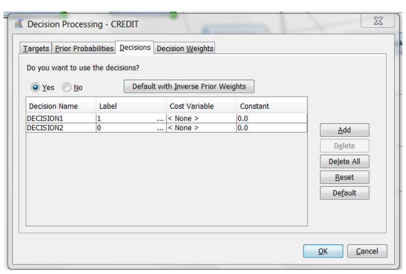
Input Data: AAEM.CREDIT

The dataset has 30 variables and 3000 line items. The response variable “**TARGET**” indicates whether an applicant defaulted on the line of credit. A value of 1 implies default. A value of 0 implies that the loan is paid-off.

1. Start Enterprise Miner and create a new project. Name the project “**Advanced Analytics Project 1**”. Link the project to SAS library AAEM. Create a new Diagram and name the new diagram “**Predictive Modeling**”.
2. Create a new Data Source by selecting SAS table AAEM.CREDIT. In “Advanced Advisor Options”, set "Class Levels Count Threshold" to 2. Set "Reject Levels Count Threshold" to 100. Select Options . Preferences. Set the Fetch Size to Max so that the whole data rather than the top 2,000 observations is fetched for later analysis. Select OK.



3. Set the role of all variables to **Input**, with the exception of the TARGET and ID variables. TARGET role is set as **target** and ID role is set as **ID**.
4. Set the measurement level for the BanruptcyInd and the target variable “TARGET” to binary. Set preferences for interactive sampling with the following settings: **12345 as the random seed, Random as sample method, and Max as the fetch size.**



5. Select “Decision Processing” in step 6 of the Data Source Wizard. The Decisions option “**Default with Inverse Prior Weights**” was selected to provide the values in the Decision Weights tab. The nonzero values used in the decision matrix are the inverse of the prior probabilities. Such a decision matrix, sometimes referred to as the central decision rule, forces a primary decision when the estimated primary outcome probability for a case exceeds the primary outcome prior probability.

Investigating Descriptive Statistics

6. Add the “StatExplore” node and RUN it for preliminary analysis of the data. We can see in the “RESULTS” window that there are 2 class variables - BanruptcyInd and TARGET.

```
Variable Summary

          Measurement    Frequency
Role         Level         Count

ID        NOMINAL            1
INPUT     BINARY             1
INPUT     INTERVAL          27
TARGET    BINARY             1
```

From output on left, there are 28 input variables, 1 ID variable to predict the binary target. In training data, 16.6 percent (or 500 observations) have TARGET as 1.

Class Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	BanruptcyInd	INPUT	2	0	0	84.67	1	15.33
TRAIN	TARGET	TARGET	2	0	0	83.33	1	16.67

Distribution of Class Target and Segment Variables
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Level	Frequency Count	Percent
TRAIN	TARGET	TARGET	0	2500	83.3333
TRAIN	TARGET	TARGET	1	500	16.6667

It can be determined from Interval Variable Summary (below) that 11 variables have missing data.

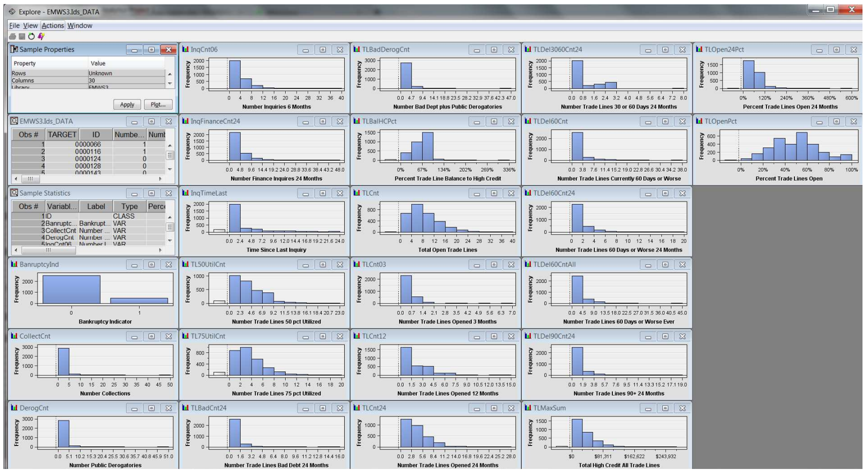
Interval Variable Summary Statistics
(maximum 500 observations printed)

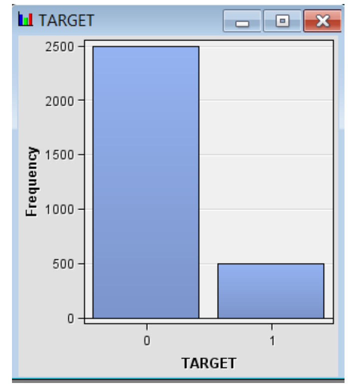
Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
CollectCnt	INPUT	0.857	2.161352	3000	0	0	0	50	7.556541	111.8365
DerogCnt	INPUT	1.43	2.731469	3000	0	0	0	51	5.045122	50.93801
InqCnt06	INPUT	3.108333	3.479171	3000	0	0	2	40	2.580016	12.82077
InqFinanceCnt24	INPUT	3.555	4.477536	3000	0	0	2	48	2.806893	13.05141
InqTimeLast	INPUT	3.108108	4.637831	2812	188	0	1	24	2.386563	5.626803
TL50UtilCnt	INPUT	4.077904	3.108076	2901	99	0	3	23	1.443077	3.350659
TL75UtilCnt	INPUT	3.121682	2.605435	2901	99	0	3	20	1.50789	3.686636
TLBadCnt24	INPUT	0.567	1.324423	3000	0	0	0	16	4.376858	28.58301
TLBadDerogCnt	INPUT	1.409	2.460434	3000	0	0	0	47	4.580204	48.24276
TLBalHCPct	INPUT	0.648178	0.266486	2959	41	0	0.6955	3.3613	-0.18073	4.015619
TLCnt	INPUT	7.879546	5.421595	2997	3	0	7	40	1.235579	2.195363
TLCnt03	INPUT	0.275	0.582084	3000	0	0	0	7	2.805575	12.66839
TLCnt12	INPUT	1.821333	1.925265	3000	0	0	1	15	1.623636	3.684793
TLCnt24	INPUT	3.882333	3.396714	3000	0	0	3	28	1.60771	4.379948
TLDel3060Cnt24	INPUT	0.726	1.163633	3000	0	0	0	8	1.381942	1.408509
TLDel60Cnt	INPUT	1.522	2.809653	3000	0	0	0	38	3.30846	17.76184
TLDel60Cnt24	INPUT	1.068333	1.806124	3000	0	0	0	20	3.080191	14.35044
TLDel60CntAll	INPUT	2.522	3.407255	3000	0	0	1	45	2.564126	12.70062
TLDel90Cnt24	INPUT	0.814667	1.609508	3000	0	0	0	19	3.623972	19.7006
TLMaxSum	INPUT	31205.9	29092.91	2960	40	0	24187	271036	2.061138	8.093434
TLOpen24Pct	INPUT	0.564219	0.480105	2997	3	0	0.5	6	2.779055	18.5329
TLOpenPct	INPUT	0.496168	0.206722	2997	3	0	0.5	1	0.379339	-0.01934
TLSatCnt	INPUT	13.51168	8.931769	2996	4	0	12	57	0.851193	0.690344
TLSatPct	INPUT	0.518331	0.234759	2996	4	0	0.5263	1	-0.12407	-0.48393
TLSum	INPUT	20151.1	19682.09	2960	40	0	15546	210612	2.276832	10.96413
TLTimeFirst	INPUT	170.1137	92.8137	3000	0	6	151	933	1.031307	2.860035
TLTimeLast	INPUT	11.87367	16.32141	3000	0	0	7	342	6.447907	80.31043

<u>Inspecting Distribution</u>

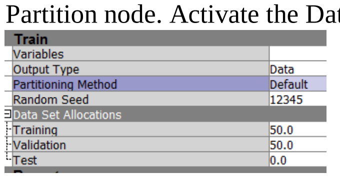
With **Data Partition node's 'Variable' property**, using the Explore window , I plotted a histogram for each of the interval variables, and a bar chart for each class variable. I found that several of the interval inputs show somewhat skewed distributions. Some of these input variables will need transformation during regression modeling. (Refer to the below Histogram plots)



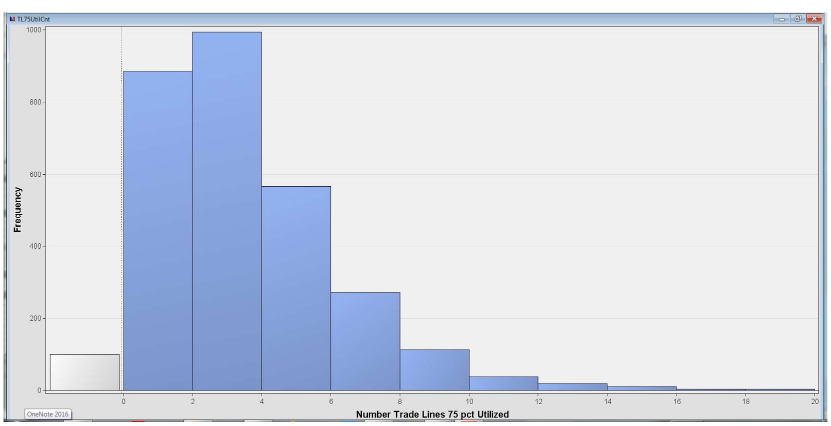


On the bar chart (above right), it can seen that approximately 80% of the observations in TARGET have a value of 0 and 20% have a value of 1. This means that approximately 20% of the customers in this sample data set defaulted on their loans.

7. Drag a Data Partition node onto the diagram and create an arrow from the CREDIT node to the Data Partition node. Activate the Data Partition node in the diagram and RUN.



- Specify the percentage of the data to allocate to training, validation, and testing data in the Properties Panel. Enter 50 for Training and 50 for Validation and 0 for Test.
- The default **Partitioning Method** for the Data Partition node is **stratified** because TARGET is a class variable. With stratified partitioning, specific variables form subgroups of the total population. Within each subgroup, all observations have an equal probability of being assigned to one of the partitioned data sets.
- Use the **Random Seed** property to specify the seed value for random numbers that generated to create the partitions.

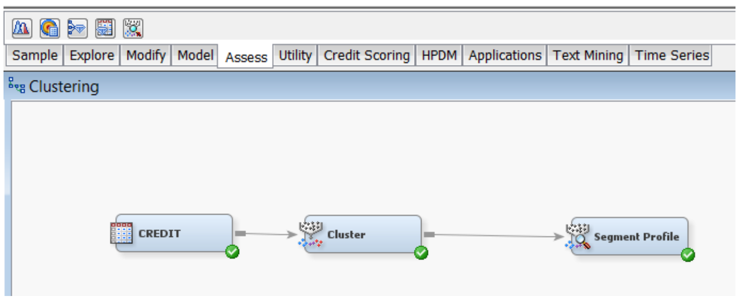


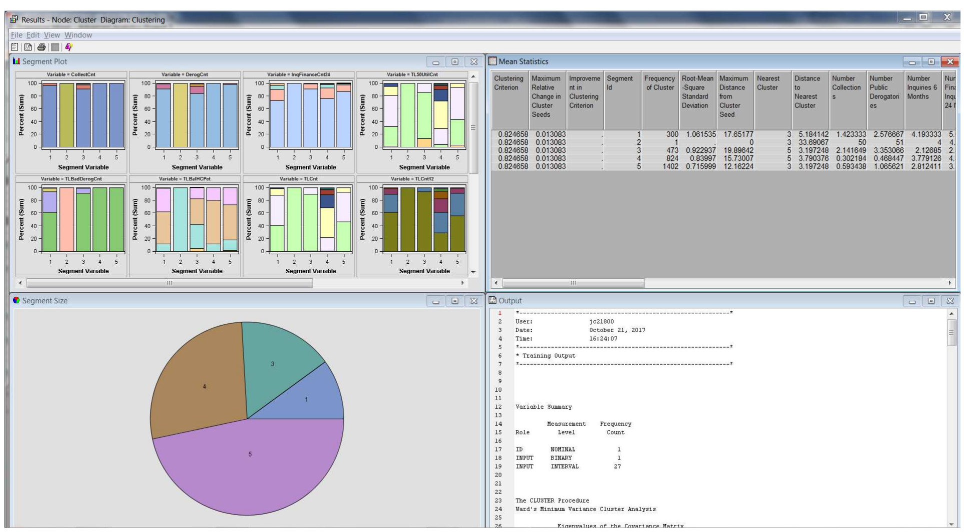
Maximize the TL75UtilCnt window. This variable indicates the Number of Trade lines 75% utilized. The gray bar on the left side of the histogram represents the missing values. Notice that the vast majority of the observations are less than 10. The TL75UtilCnt data set is skewed right.

CLUSTER ANALYSIS

Since the CREDIT dataset does not have any transaction data, doing Market Basket or Association Analysis is not an option. Here I have chosen to do cluster analysis to identify the cluster of customer with high bankruptcy rate in last 24 months and the cluster where customers had no bankruptcy. This information can be used to determine in real applications as to which customers should be offered better interest rates and more credit.

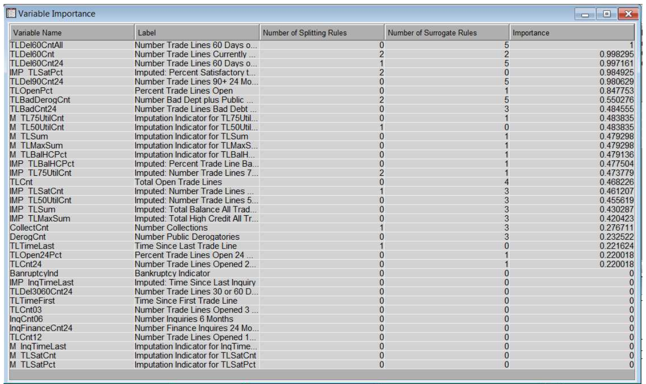
I have done some data exploration using cluster node and the Segment Profile node.
For CLUSTER node, the **Cluster Variable Role** is set to **Segment** because the next node is the Segment Profile node, which requires a variable with the role of Segment. The **Final Maximum** option has also been changed to 5 to set the maximum number of clusters to be created by the CLUSTER node. This property value depends on the dataset. Too many clusters do not allow for much insight. Too few will not properly separate the data. Now let's RUN the cluster node.





Notice in the Segment Size window that the Cluster node created 4 clusters. The Mean Statistics table in the Results includes statistics that describe each of the created clusters. The Segment Size chart shows how large each segment is. Clicking on a segment in this chart highlights the particular row in the Mean Statistics table.

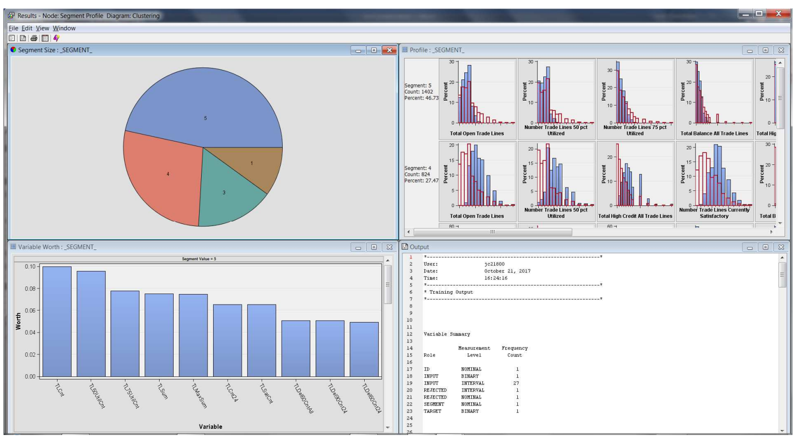
On the main menu, select **View -> Cluster Profile -> Variable Importance**. The Variable Importance window displays each variable that was used to generate the clusters and their relative importance. Variables with importance zero are not used by the cluster node.



Variable Importance

Variable Name	Label	Number of Splitting Rules	Number of Surrogate Rules	Importance
TLDel60CntAll	Number Trade Lines 60 Days o...	0	5	1
TLDel60Cnt	Number Trade Lines Currently ...	2	2	0.998295
TLDel60Cnt24	Number Trade Lines 60 Days o...	1	5	0.997161
IMP TLSatPct	Imputed: Percent Satisfactory t...	2	0	0.984925
TLDel90Cnt24	Number Trade Lines 90+ 24 Mo...	0	5	0.980629
TLOpenPct	Percent Trade Lines Open	0	1	0.847753
TLBadDerogCnt	Number Bad Dept plus Public ...	2	5	0.550276
TLBadCnt24	Number Trade Lines Bad Debt ...	0	3	0.484555
M TL75UtilCnt	Imputation Indicator for TL75Util...	0	1	0.483835
M TL50UtilCnt	Imputation Indicator for TL50Util...	1	0	0.483835
M TLSum	Imputation Indicator for TLSum	0	1	0.479298
M TLMaxSum	Imputation Indicator for TLMaxS...	0	1	0.479298
M TLBalHCPct	Imputation Indicator for TLBalH...	0	1	0.479136
IMP TLBalHCPct	Imputed: Percent Trade Line Ba...	0	1	0.477504
IMP TL75UtilCnt	Imputed: Number Trade Lines 7...	2	1	0.473779
TLCnt	Total Open Trade Lines	0	4	0.468226
IMP TLSatCnt	Imputed: Number Trade Lines ...	1	3	0.461207
IMP TL50UtilCnt	Imputed: Number Trade Lines 5...	0	3	0.455619
IMP TLSum	Imputed: Total Balance All Trad...	0	3	0.430287
IMP TLMaxSum	Imputed: Total High Credit All Tr...	0	3	0.420423
CollectCnt	Number Collections	1	3	0.276711
DerogCnt	Number Public Derogatories	0	3	0.232522
TLTimeLast	Time Since Last Trade Line	1	0	0.221624
TLOpen24Pct	Percent Trade Lines Open 24 ...	0	1	0.220018
TLCnt24	Number Trade Lines Opened 2...	0	1	0.220018
BanruptcyInd	Bankruptcy Indicator	0	0	0
IMP InqTimeLast	Imputed: Time Since Last Inquiry	0	0	0
TLDel3060Cnt24	Number Trade Lines 30 or 60 D...	0	0	0
TLTimeFirst	Time Since First Trade Line	0	0	0
TLCnt03	Number Trade Lines Opened 3 ...	0	0	0
InqCnt06	Number Inquiries 6 Months	0	0	0
InqFinanceCnt24	Number Finance Inquires 24 Mo...	0	0	0
TLCnt12	Number Trade Lines Opened 1...	0	0	0
M InqTimeLast	Imputation Indicator for InqTime...	0	0	0
M TLSatCnt	Imputation Indicator for TLSatCnt	0	0	0
M TLSatPct	Imputation Indicator for TLSatPct	0	0	0

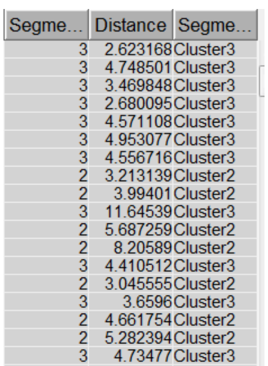
To see the decisions that the Cluster node used to segment the data, click **View >> Cluster Profile >> Tree**. After running the Segment Profile node and viewing Results window, we can scroll through charts in the Variable Worth plots to see how each variable affects the selected segment.



In the **Profile** plots, one can see a further breakdown based on the variables. For each segment, there is a sequence of charts for each variable, listed in decreasing order of importance. For each chart, the blue bars represent the distribution of the variable among observations in the segment, and the red outlines represent the overall distribution among the entire data set.

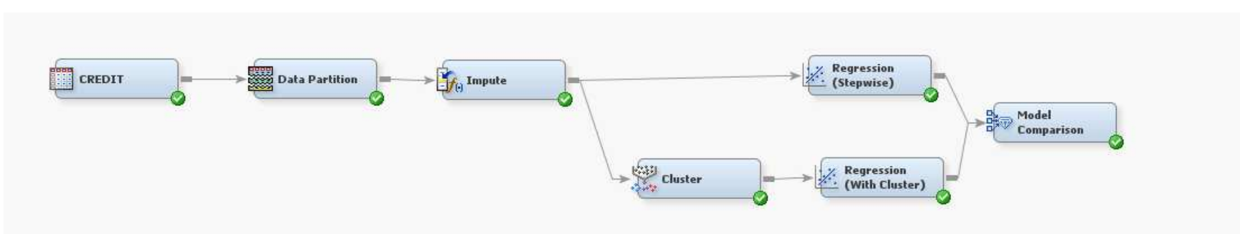
On the main menu, select **View -> Summary Statistics -> Input Means Plot.** This plot (below) displays the normalized mean value for each variable, both inside each cluster and for the complete data set.

Clicked the ellipsis button next to the **Exported Data** property on the **Cluster** node. The Exported Data — Cluster window appears. Clicked **TRAIN** and clicked **Explore**. The Clus_TRAIN window contains the entire input data set and three additional columns that are appended to the end. Scroll to the right end of the window to locate the **Segment ID**, **Distance**, and **Segment Description** columns. The **Segment ID** and **Segment Description** columns display what cluster each observation belongs to.

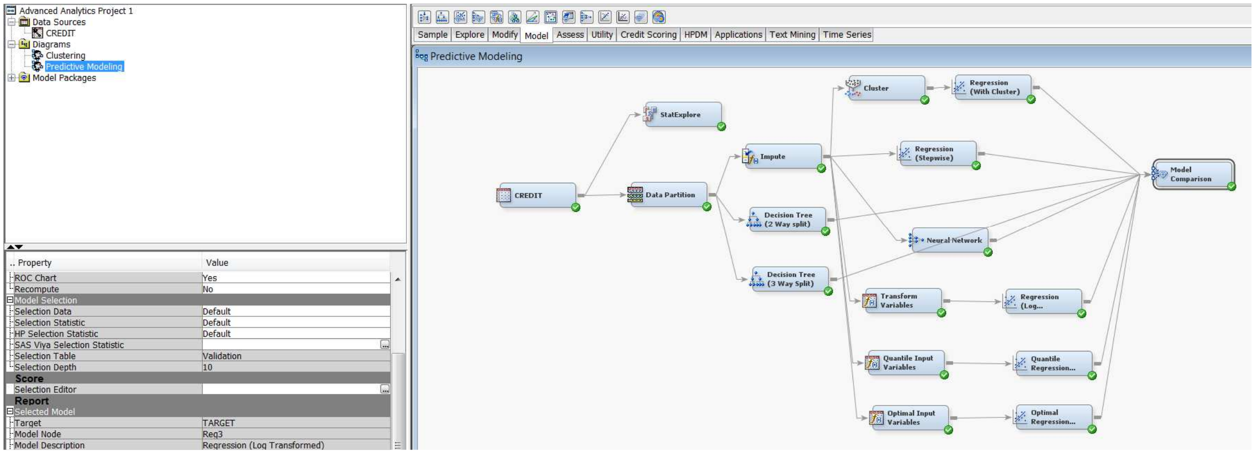


Segme...	Distance	Segme...
3	2.623168	Cluster3
3	4.748501	Cluster3
3	3.469848	Cluster3
3	2.680095	Cluster3
3	4.571108	Cluster3
3	4.953077	Cluster3
3	4.556716	Cluster3
2	3.213139	Cluster2
2	3.99401	Cluster2
3	11.64539	Cluster3
2	5.687259	Cluster2
2	8.20589	Cluster2
3	4.410512	Cluster3
2	3.045555	Cluster2
3	3.6596	Cluster3
2	4.661754	Cluster2
2	5.282394	Cluster2
3	4.73477	Cluster3

I intend to generate a prediction model that will have as input the segments from cluster node and will compare this model with other prediction models.



Phase 2: PREDICTIVE MODELING



The CREDIT node (an Input Data node) is connected to the Data Partition node. In Data Partition node properties panel, 50% of the data is used for training and 50% is used for validation.

After configuring the Data Partition node, one can create a suite that has as many modeling nodes as one wants. Next, I will add modeling nodes for Regression (Stepwise), Regression (with Cluster), Decision Tree and Neural Network.

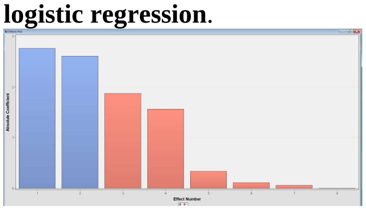
Data Imputation

After Data Partition node, I will add Impute node. Data Imputation is needed because regression and neural network models ignore incomplete observations. Tree models are able to handle missing values. Variable replacement or imputation is needed before fitting any regression or neural network model that we want to compare to a tree model. I have used the default imputation methods (**Count for class variables and Mean for interval variables**).

.

Prediction Model: Regression (Stepwise)

The Regression (Stepwise) node will use the **stepwise** method for input variable selection. and validation profit for complexity optimization. Because the target variable TARGET is a binary variable, the default model is a **binary logistic regression**.



The Effects Plot window (on left) contains a bar chart of the absolute value of the model effects. It can be seen that the most important variables, and thus best predictor variables, are IMP_TLSatPct, IMP_TLBalHCPct, TLOpenPct and TLDel3060Cnt24

The fitted model included 7 inputs (plus the intercept) as can be seen from below (from Output Window).

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq	Standardized Estimate	Exp(Est)
Intercept	1	-2.7602	0.4089	45.57	<.0001		0.063
IMP_TLBalHCPct	1	1.8759	0.3295	32.42	<.0001	0.2772	6.527
IMP_TLSatPct	1	-2.6095	0.4515	33.40	<.0001	-0.3363	0.074
InqFinanceCnt24	1	0.0610	0.0149	16.86	<.0001	0.1527	1.063
TLDel3060Cnt24	1	0.3359	0.0623	29.11	<.0001	0.2108	1.399
TLDel60Cnt24	1	0.1126	0.0408	7.62	0.0058	0.1102	1.119
TLOpenPct	1	1.5684	0.4633	11.46	0.0007	0.1792	4.799
TLTimeFirst	1	-0.00253	0.000923	7.50	0.0062	-0.1309	0.997

From the **Odds Ratio Estimates**, following interpretations can be made:

- Risk increases with increasing values of **IMP_TLBalHCPct**, **InqFinanceCnt2**4, **TLDel3060Cn**t24, **TLDel60Cnt** and **TLOpenPct**.
- Risk increases with decreasing values of **IMP_TLSatPct** and **TLTimeFirst**.

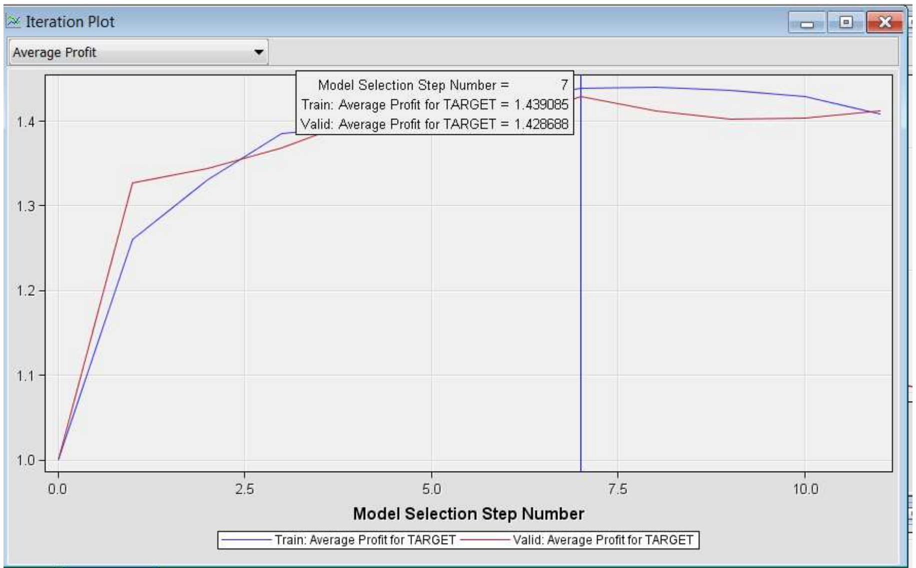
Odds Ratio Estimates

Effect	Point Estimate
IMP_TLBalHCPct	6.527
IMP_TLSatPct	0.074
InqFinanceCnt24	1.063
TLDel3060Cnt24	1.399
TLDel60Cnt24	1.119
TLOpenPct	4.799
TLTimeFirst	0.997

In the Score Rankings Overlay window (above right), select Cumulative % Response. It can be seen that this model shows a smooth decrease in the cumulative percent response. Notice that at the top 5% of the data, approximately 59% of the loan recipients default on their loan. Overall, on validation set, approx 16.6 % of the loan recipients default on their loan.

Depth = 100
Cumulative % Response = 16.66667
Data Role = VALIDATE

Below is the Iteration Plot (View -> Model -> Iteration Plot in Results Window). The iteration plot can be set to show average profit versus iteration. The plot showed how the profit varied with the model complexity. From the plot, maximum validation profit equals 1.428 at model selection step number 7.



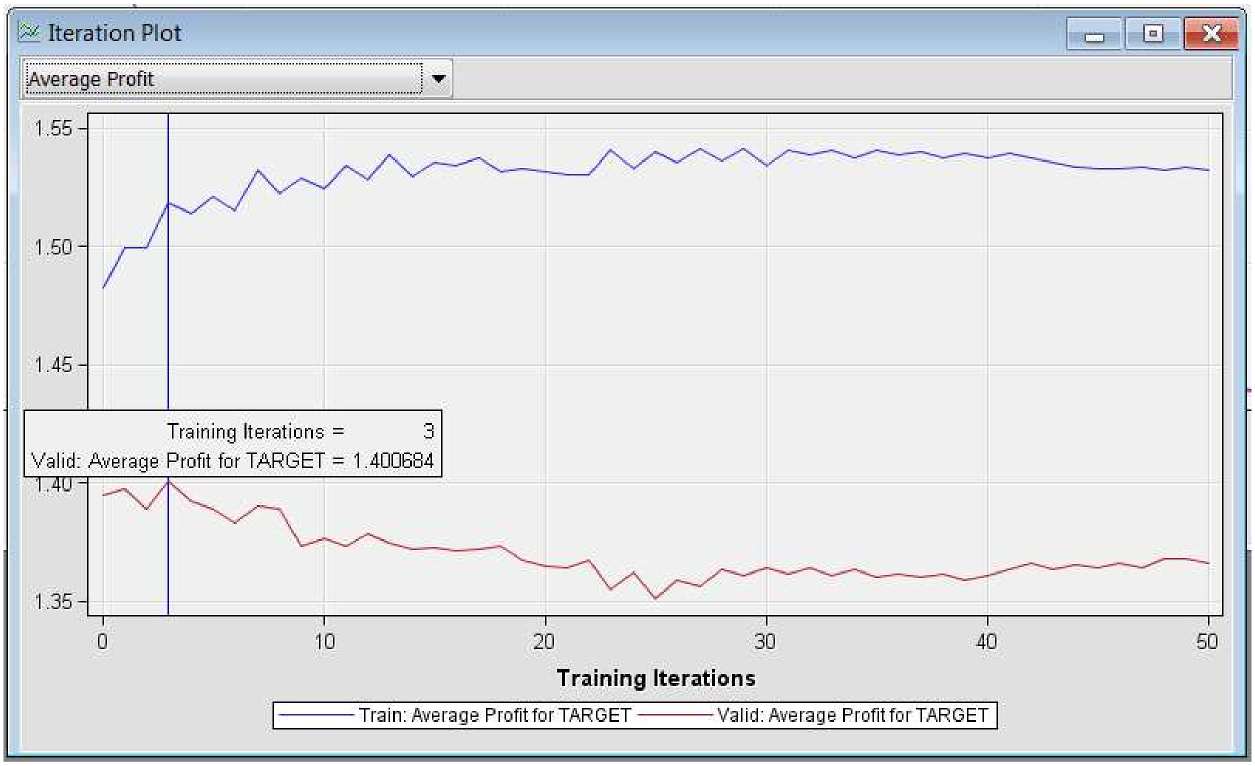
Prediction Model: Regression (with Cluster)

A CLUSTER node is added after the impute node. The Cluster node will capture the higher-level interactions in the data and will pass that information on to the regression model. The Final Maximum property is set to 5. The **Cluster Variable** Role property is set to **Input** instead of **Segment** because the next node will use the cluster input for modeling. The **Internal Standardization** is set to **Standardization** (which is default). Some columns will have more variance than others due to the scale of measurements (such as feet versus miles or cents versus dollars). This property scales the columns of the data, such that the scale of the columns does not affect the model.

The Regression (With Cluster) node uses the same properties as the Regression (Stepwise) node. The only difference is that the Regression (With Cluster) node has the segments as input that the Cluster node creates.

Prediction Model: Neural Network

Neural Network is used to investigate regression lack of fit. The default settings of the Neural Network node are used. The iteration plot showed slightly lower validation average profit compared to the stepwise regression model. Hence, Regression (Stepwise) is a better model fit.



Performing Variable Transformation

As I explored the data, I identified that some variables have highly skewed distributions. This can. Hence, I fitted the regression model after performing the variable transformation.

During data exploration, it was noticed that many interval inputs had skewed distributions. Such distributions create high leverage points that cause a small percentage of the data points to have a large amount of influence on the final model.

Prediction Model: Log Transformed Stepwise Regression

Name	Skewness
CollectCnt	7.556541
TLTimeLast	6.447907
DerogCnt	5.045122
TLBadDerogCn	4.580204
TLBadCnt24	4.376858
TLDel90Cnt24	3.623972
TLDel60Cnt	3.30846
TLDel60Cnt24	3.080191
InqFinanceCnt2	2.806893
TLCnt03	2.805575
TLOpen24Pct	2.779055
InqCnt06	2.580016
TLDel60CntAll	2.564126

The **Transform Variables** node can be used to regularize the distributions of the model inputs before fitting the stepwise regression.

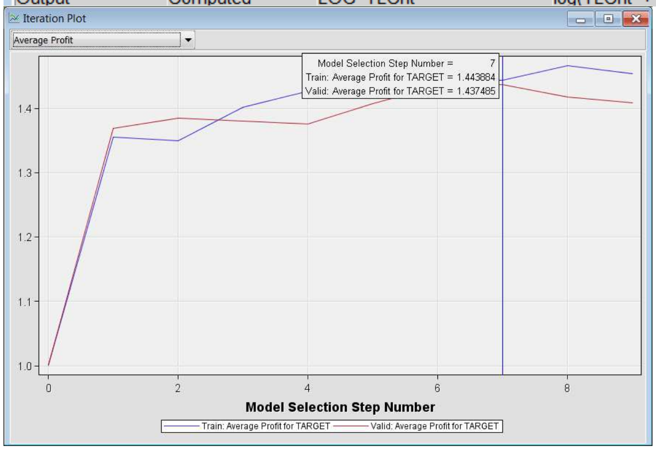
Click next to the **Variables** property of the Transform Variables node. The Variables window appears. In the Variables window, select the **Statistics** option and see the **Skewness** and **Kurtosis** statistics. (See here on left side). Note the most skewed variables are listed in order.

On **Transform Variables** node, set the **Interval Inputs** property to "**Log**" to apply log transformation to all of the input variables. RUN the Regression node. In RESULTS window, maximize the Transformations Statistics window. The **Formula** column indicates the expression used to transform each variable. Note that the absolute value of the **Skewness** statistic for the transformed values is smaller than that of the original variables.

Transformations Statistics

Source	Method	Variable Name	Formula	Skewness
Output	Computed	LOG_CollectCnt	log(CollectCnt + 1)	1.707518
Output	Computed	LOG_DerogCnt	log(DerogCnt + 1)	1.106167
Output	Computed	LOG_IMP_InqTimeLast	log(IMP_InqTimeLast + 1)	0.71025
Output	Computed	LOG_IMP_TL50UtilCnt	log(IMP_TL50UtilCnt + 1)	-0.30818
Output	Computed	LOG_IMP_TL75UtilCnt	log(IMP_TL75UtilCnt + 1)	-0.24553
Output	Computed	LOG_IMP_TLBalHCPct	log(IMP_TLBalHCPct + 1)	-0.93928
Output	Computed	LOG_IMP_TLMaxSum	log(IMP_TLMaxSum + 1)	-3.1587
Output	Computed	LOG_IMP_TLSatCnt	log(IMP_TLSatCnt + 1)	-0.65172
Output	Computed	LOG_IMP_TLSatPct	log(IMP_TLSatPct + 1)	-0.4741
Output	Computed	LOG_IMP_TLSum	log(IMP_TLSum + 1)	-2.79933
Output	Computed	LOG_InqCnt06	log(InqCnt06 + 1)	0.056314
Output	Computed	LOG_InqFinanceCnt24	log(InqFinanceCnt24 + 1)	0.210319
Output	Computed	LOG_TLBadCnt24	log(TLBadCnt24 + 1)	1.805432
Output	Computed	LOG_TLBadDerogCnt	log(TLBadDerogCnt + 1)	0.985677
Output	Computed	LOG_TLCnt	log(TLCnt + 1)	-0.36478

The Transformed Regression node performed stepwise selection from the transformed inputs.



Odds Ratio Estimates

Effect	Point Estimate
LOG_IMP_TL75UtilCnt	1.746
LOG_IMP_TLBalHCPct	8.362
LOG_IMP_TLSatPct	0.030
LOG_InqFinanceCnt24	1.326
LOG_TLDel3060Cnt24	1.902
LOG_TLDel60Cnt24	1.637
LOG_TLTimeFirst	0.510

From the Iteration Plot on left, it can be seen that the log transformed stepwise regression model **show slight performance improvement compared to stepwise regression model**. Hence log transformation does fetch minor improvement in model fit.

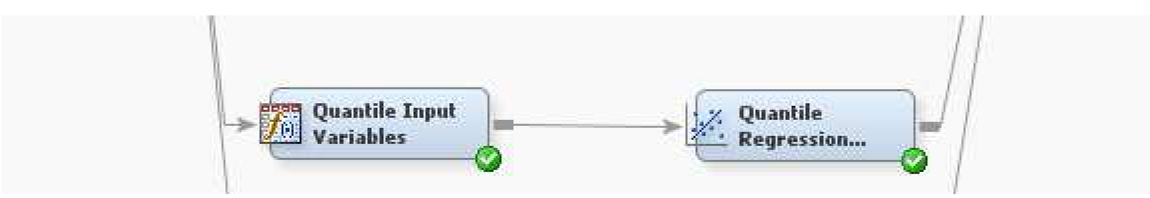
Prediction Model: Regression (Quantile Transformation)

With this regression approach, I will use Bin Input Variables node to partition each interval input into bins with equal **sizes**. This is a discretization approach where input variables are partitioned into discrete ranges.

Model Selection	
Selection Model	Forward
Selection Criterion	Default
Use Selection Defaults	No
Selection Options	...
Optimization Options	
Technique	Default
Default Optimization	Yes
Max Iterations	0
Max Function Calls	0

Entry Significance Level	0.05
Stay Significance Level	0.025
Start Variable Number	10
Stop Variable Number	0
Force Candidate Effects	0
Hierarchy Effects	Class
Moving Effect Rule	None
Maximum Number of Steps	0

Set the value of the **Selection Model** property to **Forward, Use Selection Defaults** property to **No, Stay Significance Level** to **0.025 and Start Variable Number** to **10.**

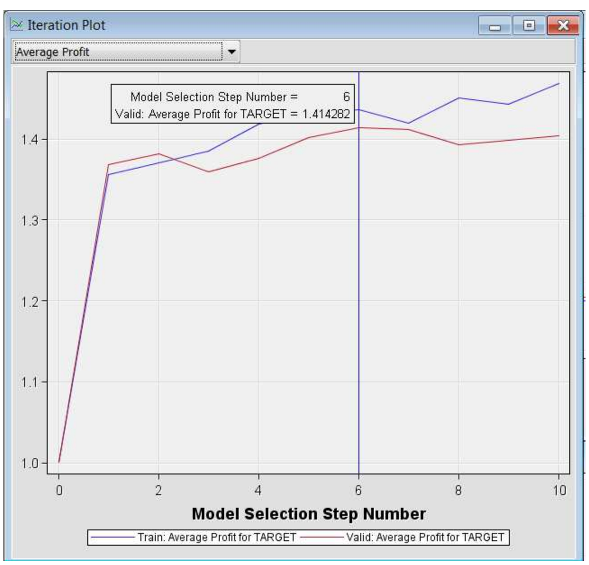


Property	Value
General	
Node ID	Trans2
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Formulas	
Interactions	
SAS Code	
Default Methods	
Interval Inputs	Quantile
Interval Targets	None
Class Inputs	None
Class Targets	None
Treat Missing as Level	No
Sample Properties	
Method	First N
Size	Default
Random Seed	12345
Optimal Binning	
Number of Bins	4
Missing Values	Use in Search
Grouping Method	
Cutoff Value	0.1
Group Missing	No

Odds Ratio Estimates

Effect		Point Estimate
PCTL_IMP_TLBalHCPct	01:low-0.513 vs 04:0.8389-high	0.272
PCTL_IMP_TLBalHCPct	02:0.513-0.7041 vs 04:0.8389-high	0.452
PCTL_IMP_TLBalHCPct	03:0.7041-0.8389 vs 04:0.8389-high	0.630
PCTL_IMP_TLSatPct	01:low-0.3529 vs 04:0.6886-high	1.860
PCTL_IMP_TLSatPct	02:0.3529-0.5333 vs 04:0.6886-high	1.130
PCTL_IMP_TLSatPct	03:0.5333-0.6886 vs 04:0.6886-high	1.040
PCTL_InqFinanceCnt24	01:low-1 vs 04:5-high	0.599
PCTL_InqFinanceCnt24	02:1-2 vs 04:5-high	0.404
PCTL_InqFinanceCnt24	03:2-5 vs 04:5-high	0.807
PCTL_TLDel3060Cnt24	03:0-1 vs 04:1-high	0.453
PCTL_TLDel60Cnt24	03:0-1 vs 04:1-high	0.357
PCTL_TLTimeFirst	01:low-107 vs 04:230-high	1.688
PCTL_TLTimeFirst	02:107-152 vs 04:230-high	1.477
PCTL_TLTimeFirst	03:152-230 vs 04:230-high	0.837

I also tested by fitting the model with “***Bucket***” transformation, however, the model showed poor performance in comparison to other models. This was caused by small size of CREDIT data set that resulted in small number of observations in each bin. With “***Quantile***” transformation, each bin included a reasonable number of cases and hence more stable parameter estimates and improved model fit was achieved. Although, as can be seen in below Iteration Plot, the average profit is still slightly smaller than stepwise regression model.



Prediction Model: Regression (Optimal Transformation)

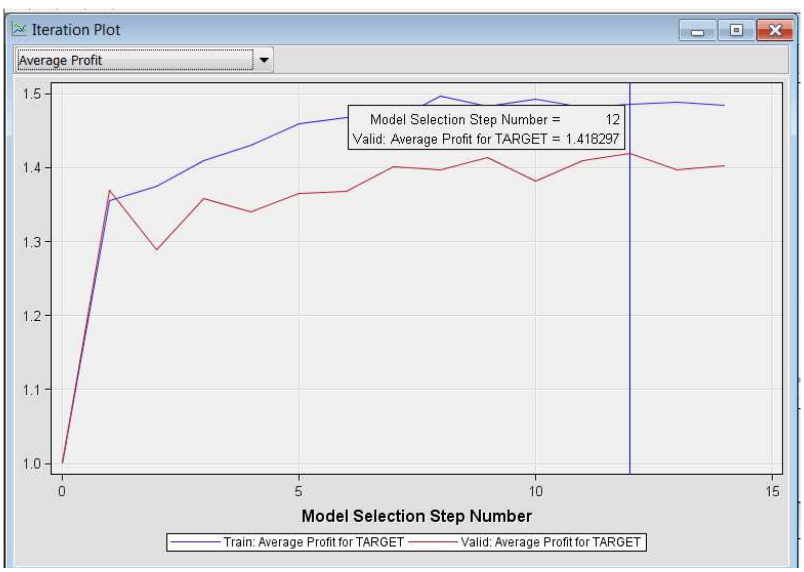
SAS Code	
Default Methods	
Interval Inputs	Optimal Binning
Interval Targets	None
Class Inputs	None
Class Targets	None
Treat Missing as Level	No

I tested with another type of transformation called ***"Optimal Binning".*** The final fitted model included 10 input variables with 18 degree-of-freedom.

Odds Ratio Estimates

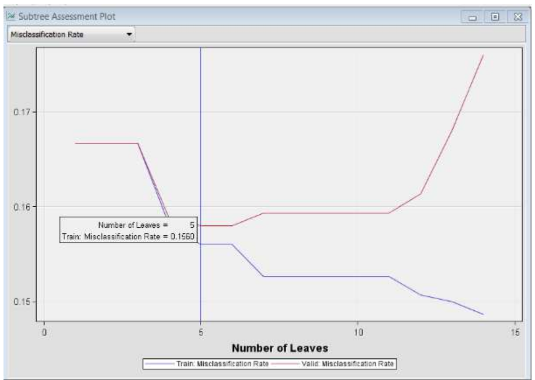
Effect		Point Estimate
BanruptcyInd	0 vs 1	2.267
OPT_IMP_TL75UtilCnt	01:low-1.5 vs 03:8.5-high	0.270
OPT_IMP_TL75UtilCnt	02:1.5-8.5, MISSING vs 03:8.5-high	0.409
OPT_IMP_TLBalHCPct	01:low-0.6706, MISSING vs 04:1.0213-high	0.090
OPT_IMP_TLBalHCPct	02:0.6706-0.86785 vs 04:1.0213-high	0.155
OPT_IMP_TLBalHCPct	03:0.86785-1.0213 vs 04:1.0213-high	0.250
OPT_IMP_TLSatPct	01:low-0.2094 vs 03:0.4655-high, MISSING	5.067
OPT_IMP_TLSatPct	02:0.2094-0.4655 vs 03:0.4655-high, MISSING	1.970
OPT_InqFinanceCnt24	01:low-2.5, MISSING vs 03:7.5-high	0.353
OPT_InqFinanceCnt24	02:2.5-7.5 vs 03:7.5-high	0.657
OPT_TLDel3060Cnt24	01:low-1.5, MISSING vs 02:1.5-high	0.499
OPT_TLDel60Cnt	01:low-0.5, MISSING vs 03:14.5-high	0.084
OPT_TLDel60Cnt	02:0.5-14.5 vs 03:14.5-high	0.074
OPT_TLDel60Cnt24	01:low-0.5, MISSING vs 03:5.5-high	0.327
OPT_TLDel60Cnt24	02:0.5-5.5 vs 03:5.5-high	0.882
OPT_TLTimeFirst	01:low-154.5, MISSING vs 02:154.5-high	1.926
TLOpenPct		3.337

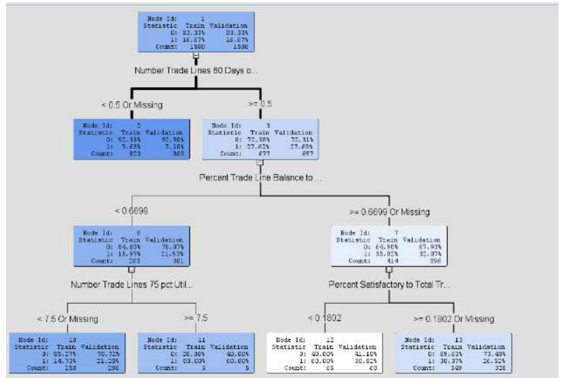
The validation average profit is slightly smaller than the original model. A substantial difference in profit between the training and validation data suggests the overfitting by the model.



Prediction Model: Decision Tree (2 way split)

Add a Decision Tree node in the diagram and select “Misclassification” as Assessment Measure and RUN the node. From the “Subtree Assessment Plot” on left, using “Misclassification” results in a tree with 14 leaves. The optimal tree is with 5 leaves.





From below, it can be seen that the variable TLDel60Cnt24 is the variable chosen for the first split. This choice is based on the highest chi-square based Log worth among all candidate variables.

Variable Importance

Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
TLDel60Cnt24	Number Trade Lines 60 Days or Worse 24 Months	1	1.0000	1.0000	1.0000
TLBalHCPct	Percent Trade Line Balance to High Credit	1	0.6279	0.1481	0.2358
TLSatPct	Percent Satisfactory to Total Trade Lines	1	0.5699	0.6021	1.0565
TL75UtilCnt	Number Trade Lines 75 pct Utilized	1	0.3756	0.1453	0.3867

Following inferences can be made from the model:

- if Number Trade Lines 60 Days or Worse 24 Months (TLDel60Cnt) < 0.5 or missing, then model predicts on validation data that there is 7.10 percent cases that will default on loan (bad debt).
- if Percent Trade Line Balance to High Credit (TLBalHCPct) < 0.66985 AND Number Trade Lines 75 pct Utilized (TL75UtilCnt) < 7.5 or MISSING AND Number Trade Lines 60 Days or Worse 24 Months (TLDel60Cnt) >= 0.5 then model predicts on validation data that there is 21.28 percent cases that will default on loan (bad debt).

More decision rules can be analyzed from Results window: View -> Model -> Node Rules

Fit Statistics

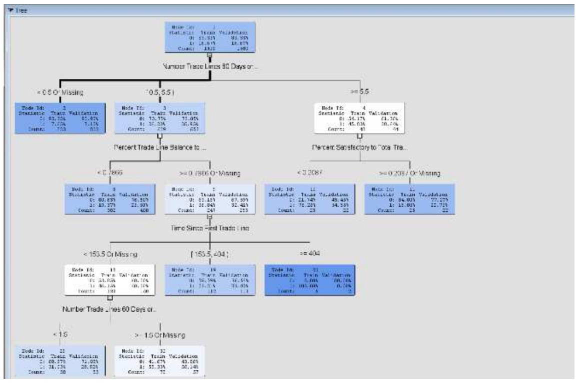
Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Te
TARGET		NOBS	Sum of Frequencies	1500	1500	
TARGET		MISC	Misclassification Rate	0.156	0.158	
TARGET		MAX	Maximum Absolute Error	0.923451	0.923451	
TARGET		SSE	Sum of Squared Errors	361.5713	372.2628	
TARGET		ASE	Average Squared Error	0.120524	0.124088	
TARGET		RASE	Root Average Squared Error	0.347165	0.352261	
TARGET		DIV	Divisor for ASE	3000	3000	
TARGET		DFT	Total Degrees of Freedom	1500	.	
TARGET		APROF	Average Profit for TARGET	1.379912	1.303127	
TARGET		PROF	Total Profit for TARGET	2069.868	1954.691	

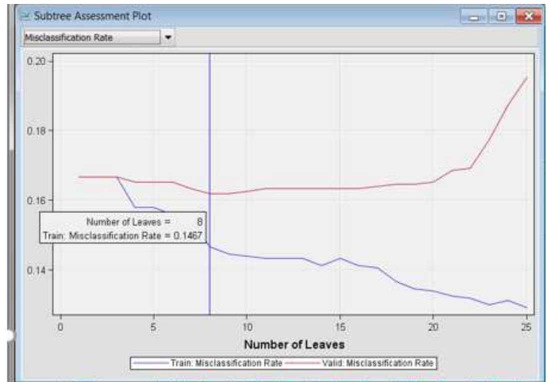
From above Fit Statistics window, the Misclassification Rate for validation set is 0.158.

Prediction Model: Decision Tree (3 way split)

Add a Decision Tree node in the diagram. Select “Misclassification” as Assessment Measure. Set “Maximum Branch” as 3. RUN the node.

From the “Subtree Assessment Plot”, using “Misclassification” results in a tree with 25 leaves. The optimal tree is with 8 leaves.





Fit Statistics

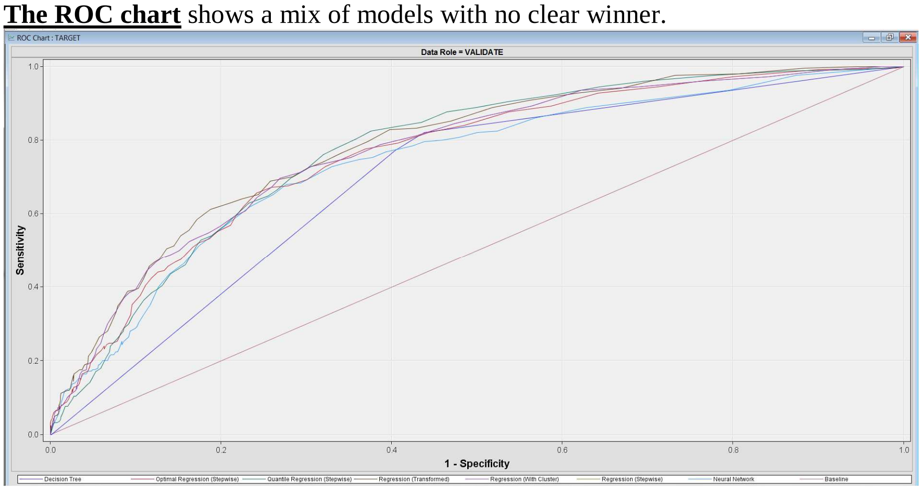
Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
TARGET		NOBS	Sum of Frequencies	1500	1500	
TARGET		MISC	Misclassification Rate	0.146667	0.162	
TARGET		MAX	Maximum Absolute Error	0.923451	1	
TARGET		SSE	Sum of Squared Errors	349.9869	378.5919	
TARGET		ASE	Average Squared Error	0.116662	0.126197	
TARGET		RASE	Root Average Squared Error	0.341559	0.355243	
TARGET		DIV	Divisor for ASE	3000	3000	
TARGET		DFT	Total Degrees of Freedom	1500		
TARGET		APROF	Average Profit for TARGET	1.356679	1.362274	
TARGET		PROF	Total Profit for TARGET	2035.018	2043.411	

The Misclassification rate for validation set is 0.162. It looks like the first tree with two-branch split has the lower validation misclassification rate (0.158) and hence is the better model than 3-branch split decision tree.

Phase 3: Model Comparison

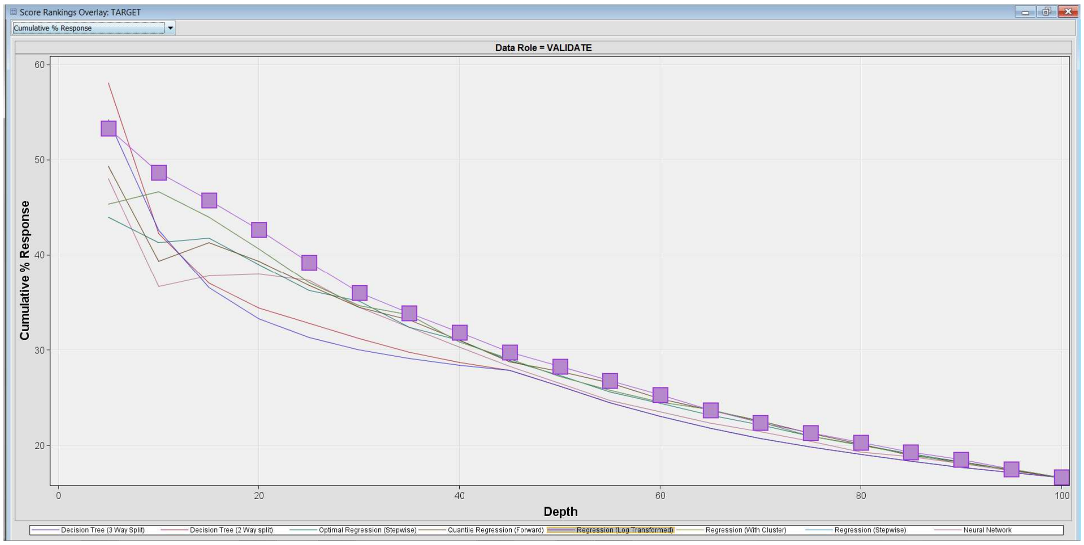
Model Comparison node is added to the Diagram. All modeling nodes are input to this comparison node which will compare each model and will select the best model that can be used for future predictions. The Selection Statistic property is set to Default, however any of the available options can be chosen as a criterion to determine the best model.

The ROC chart shows a mix of models with no clear winner.



Score Rankings Overlay Chart

"Cumulative % Response" is selected from dropdown. This chart groups individuals based on the predicted probability of response (TARGET = 1), and then plots the percentage of respondents. Regression (Log Transformation) model chart is highlighted and performs better at almost every depth.



Fit Statistics for Validation set from Results OUTPUT window

Fit Statistics

Selected Model	Predecessor Node	Model Description	Selection Criterion: Valid: Average Profit for TARGET	Valid: Average Squared Error	Valid: Misclassification Rate
Y	Reg3	Regression (Log Transformed)	1.437485	0.116578	0.165333
	Reg	Regression (Stepwise)	1.428688	0.1199	0.170667
	Reg2	Regression (With Cluster)	1.428688	0.1199	0.170667
	Reg5	Optimal Regression (Stepwise)	1.418297	0.123859	0.174667
	Reg4	Quantile Regression (Forward)	1.41429	0.123404	0.173333
	Neural	Neural Network	1.400684	0.131139	0.182667
	Tree2	Decision Tree (3 Way Split)	1.362274	0.126197	0.162
	Tree	Decision Tree (2 Way split)	1.303127	0.124088	0.158

In the Fit Statistics window, the models are listed in order with best model being at the top of the table and the worst model at the bottom. With Selection Criteria set to "Average Profit for TARGET", THE Stepwise Regression with Log Transformation is the best fitted model as can be seen with highest value for Average Profit on Validation set. Also, this model has lowest Average Squared Error and comparatively lower misclassification rate than any other regression model.

```
Data Role=Valid

Statistics

Valid: Kolmogorov-Smirnov Statistic
Valid: Average Profit for TARGET
Valid: Average Squared Error
Valid: Roc Index
Valid: Average Error Function
Valid: Bin-Based Two-Way Kolmogorov-Smirnov Probability Cutoff
Valid: Cumulative Percent Captured Response
Valid: Percent Captured Response
Valid: Divisor for VASE
Valid: Error Function
Valid: Gain
Valid: Gini Coefficient
Valid: Bin-Based Two-Way Kolmogorov-Smirnov Statistic
Valid: Kolmogorov-Smirnov Probability Cutoff
Valid: Cumulative Lift
Valid: Lift
Valid: Maximum Absolute Error
Valid: Misclassification Rate
Valid: Mean Squared Error
Valid: Sum of Frequencies
Valid: Total Profit for TARGET
Valid: Root Average Squared Error
Valid: Cumulative Percent Response
Valid: Percent Response
Valid: Root Mean Squared Error
Valid: Sum of Squared Errors
Valid: Sum of Case Weights Times Freq
Valid: Number of Wrong Classifications
```

Reg3	Reg	Reg2	Reg5	Reg4	Neural	Tree2	Tree
0.44	0.43	0.43	0.41	0.42	0.40	0.37	0.37
1.44	1.43	1.43	1.42	1.41	1.40	1.36	1.30
0.12	0.12	0.12	0.12	0.12	0.13	0.13	0.12
0.79	0.77	0.77	0.76	0.77	0.74	0.71	0.72
0.37	0.38	0.38	0.40	0.39	0.43	.	.
0.16	0.18	0.18	0.14	0.18	0.20	0.19	0.15
29.20	28.00	28.00	24.80	23.60	22.00	25.58	25.37
13.20	14.40	14.40	11.60	8.80	7.60	9.33	7.96
3000.00	3000.00	3000.00	3000.00	3000.00	3000.00	3000.00	3000.00
1117.57	1152.26	1152.26	1186.46	1174.17	1290.35	.	.
192.00	180.00	180.00	148.00	136.00	120.00	155.81	153.70
0.58	0.54	0.54	0.53	0.54	0.48	0.42	0.44
0.44	0.43	0.43	0.41	0.42	0.40	0.36	0.36
0.16	0.17	0.17	0.17	0.17	0.19	0.08	0.08
2.92	2.80	2.80	2.48	2.36	2.20	2.56	2.54
2.64	2.88	2.88	2.32	1.76	1.52	1.87	1.59
0.97	0.97	0.97	1.00	0.99	1.00	1.00	0.92
0.17	0.17	0.17	0.17	0.17	0.18	0.16	0.16

Model Selection

The best model, as measured by average profit, is the Regression (Log Transformed). ***This model also has the highest KS statistic, has the highest ROC-index. This model has lower misclassification rate than any other regression model, however, the rate is higher than Decision Tree models.***

Hence, the model of choice depends on statistical performance as well as the business need.

My choice of model is the Stepwise Regression with Log Transformation, because it offered consistently good performance across multiple assessment measures.