Application Scoring Assignment. Muhammad Usman Majid

Below are the steps I followed in SAS miner for my dataset 16 to create application Score card for German bank.

Create a New Project

To create the project follow the steps below:

1. **Open** SAS Enterprise Miner.

2. In the Welcome to Enterprise Miner window, click **New Project**. The Create New

Project Wizard opens.

3. Proceed through the steps that are outlined in the wizard.

a. Select the logical workspace server to use. Click **Next**.

b. Enter your Project Name.

The **SAS Server Directory** is the directory on the server machine in which SAS

data sets and other files that are generated by the project will be stored. Click Next.

c. The **SAS Folder Location** is the directory on the server machine in which the

project itself will be stored. Click **Next**.

d. Click **Finish**.

Create a Data Source

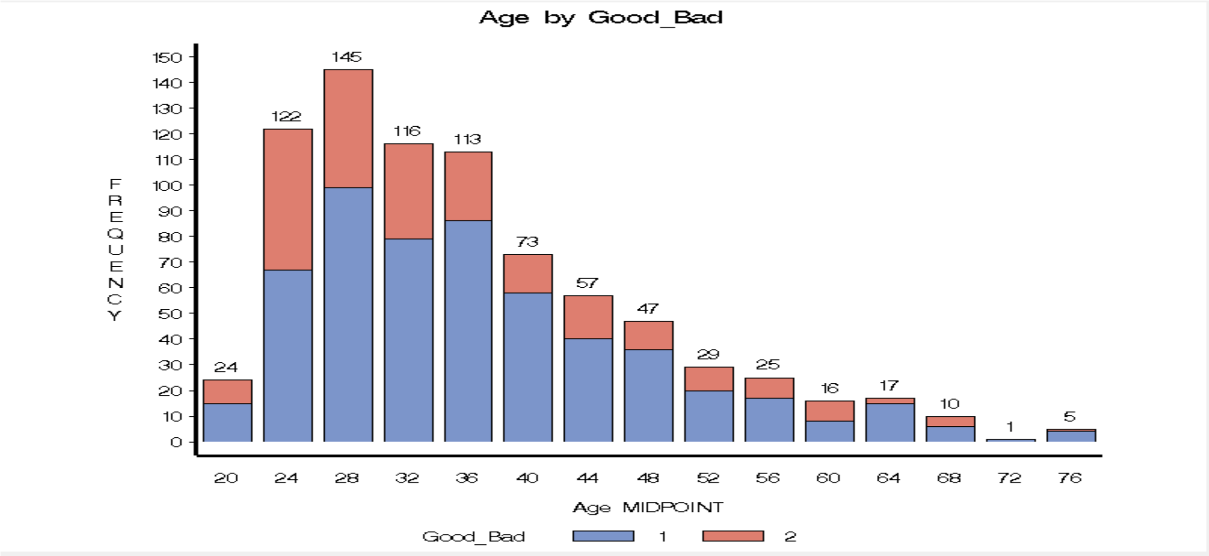
In order to create a data source in CSV format in SAS Miner, **click** on create diagram in the main tab and then write the diagram name. I gave Credit Scoring. Then **Drag and drop** File Import node and rename it as Accepts as we will use accepts data here. Now **go to** Import file in train property and upload Accepts file here, Next **right click** on the node and select edit variable. **Select role** of GoodBad variable as target and level as binary. **Select OK** and run the node.

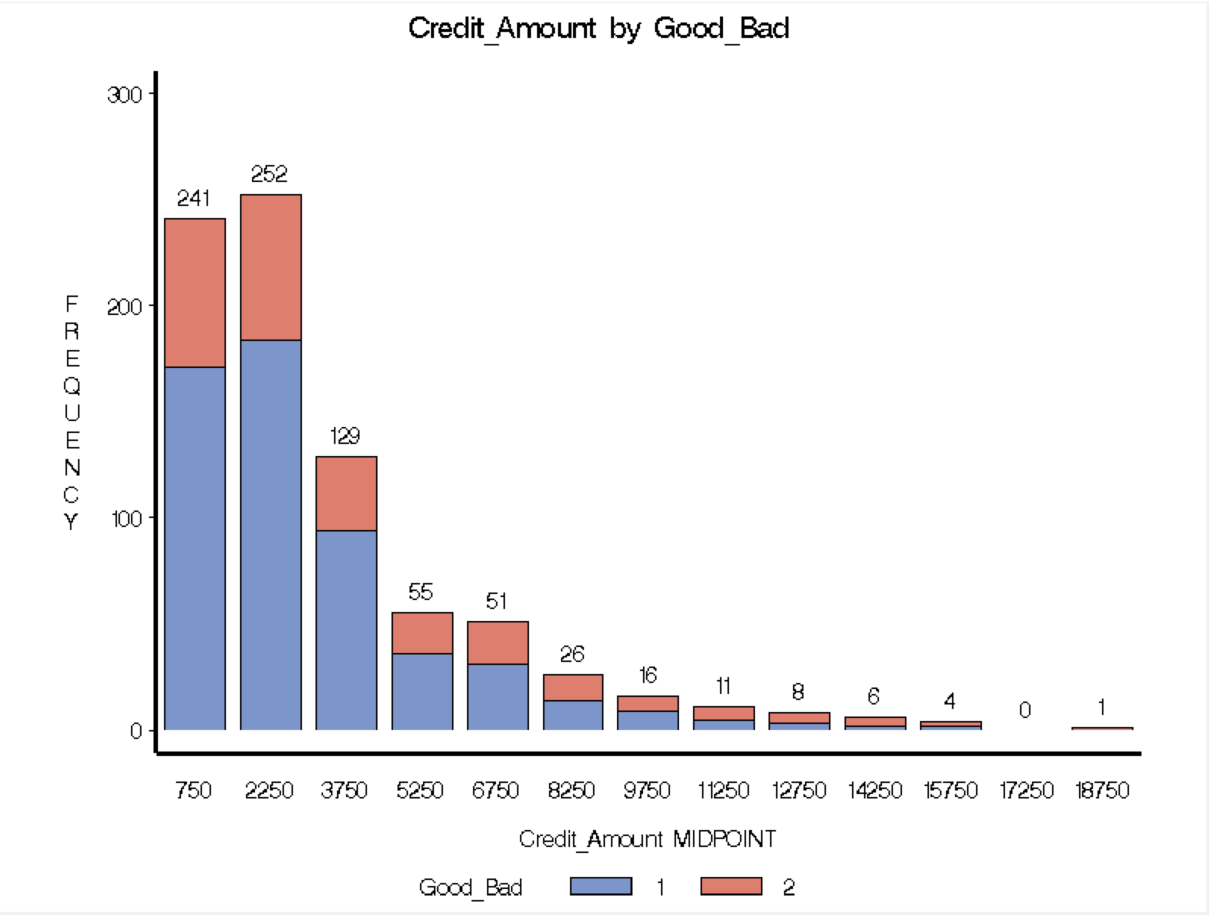
Exploring the Data Set.

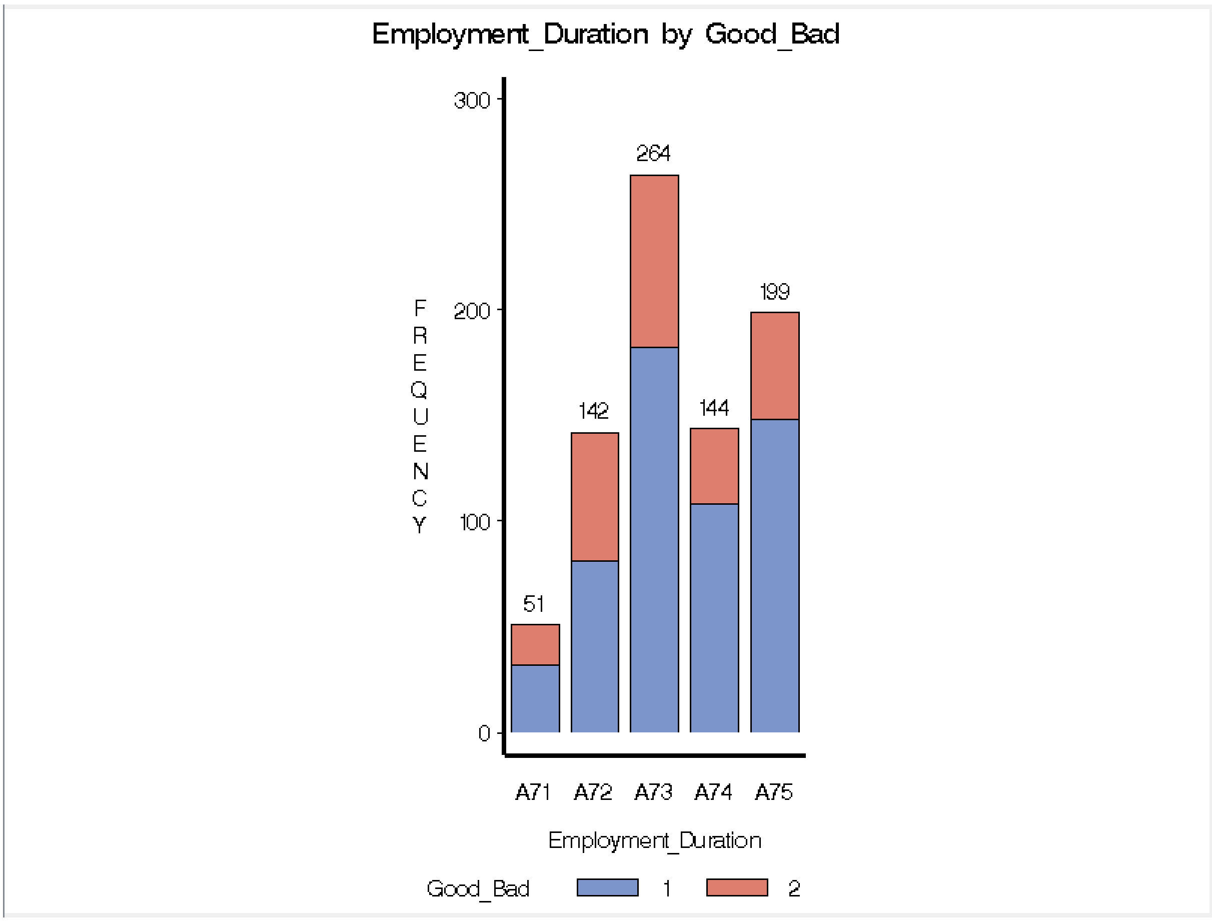
In order to explore dataset **Drag and Drop** Multiplot from explore tab. **Right click** on the node and **run.** Now below are some of the variables I explored per good and bad (target variable).

Variable Summary

Role	Measurement Level	Frequency Count
INPUT	INTERVAL	7
INPUT	NOMINAL	13
TARGET	BINARY	1


Age by Good_Bad
FREQUENCY
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
24
122
145
116
113
73
57
47
29
25
16
17
10
1
5
20
24
28
32
36
40
44
48
52
56
60
64
68
72
76
Age MIDPOINT
Good_Bad
1
2


Credit_Amount by Good_Bad
FREQUENCY
300
200
100
0
241
252
129
55
51
26
16
11
8
6
4
0
1
750
2250
3750
5250
6750
8250
9750
11250
12750
14250
15750
17250
18750
Credit_Amount MIDPOINT
Good_Bad
1
2



Partition the Data.

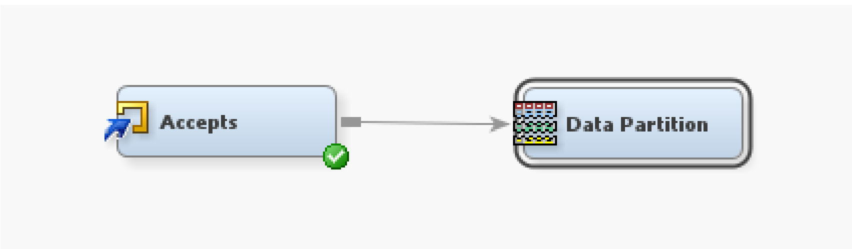
In data mining the strategy is to assess the quality of model generalization by partition the data source. A portion of the data, training data set, is used for preliminary model fitting. The rest is reserved for empirical validation and is often split into two parts: validation data and test data. The validation data set is used to prevent a modeling node from overfitting the training data and to compare models. The test data set is used for a final assessment of the model.

To use the Data Partition node to partition the input data into training and validation sets:

1. **Select** the Sample tab on the Toolbar.

2. **Select** the Data Partition node icon. **Drag** the node into the Diagram Workspace.

3. Connect the Accepts node to the Data Partition node



4. **Select** the Data Partition node. In the Properties Panel, scroll down to view the data

set allocations in the Train properties.

• Click on the value of Training, and enter 70.0

• Click on the value of Validation, and enter 30.0

• Click on the value of Test, and enter 0.0

These properties define the percentage of input data used in each type of mining data set. In current scenario, I am using a training data set and a validation data set, but we do not use a test data set.

5. In the Diagram Workspace, **right-click** the Data Partition node, and **select Run** from the resulting menu. **Click** Yes in the confirmation window that opens.

6. In the Run Status window, **click OK**.

Partition Summary

Type	Data Set	Number of Observations
DATA	EMWS1.FIMPORT_train	800
TRAIN	EMWS1.Part_TRAIN	558
VALIDATE	EMWS1.Part_VALIDATE	242

Summary Statistics for Class Targets

Data=DATA

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
Good_Bad	1	1	551	68.875	
Good_Bad	2	2	249	31.125	

Data=TRAIN

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
Good_Bad	1	1	385	68.9964	
Good_Bad	2	2	173	31.0036	

Data=VALIDATE

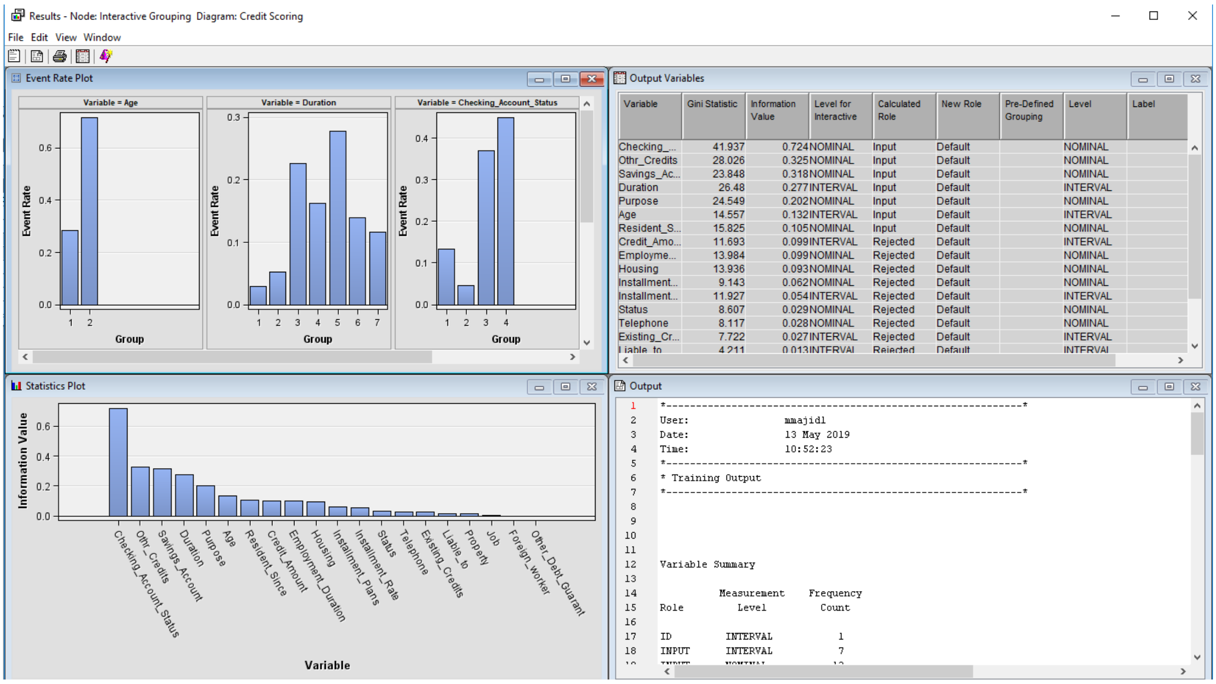
Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
Good_Bad	1	1	166	68.5950	
Good_Bad	2	2	76	31.4050	

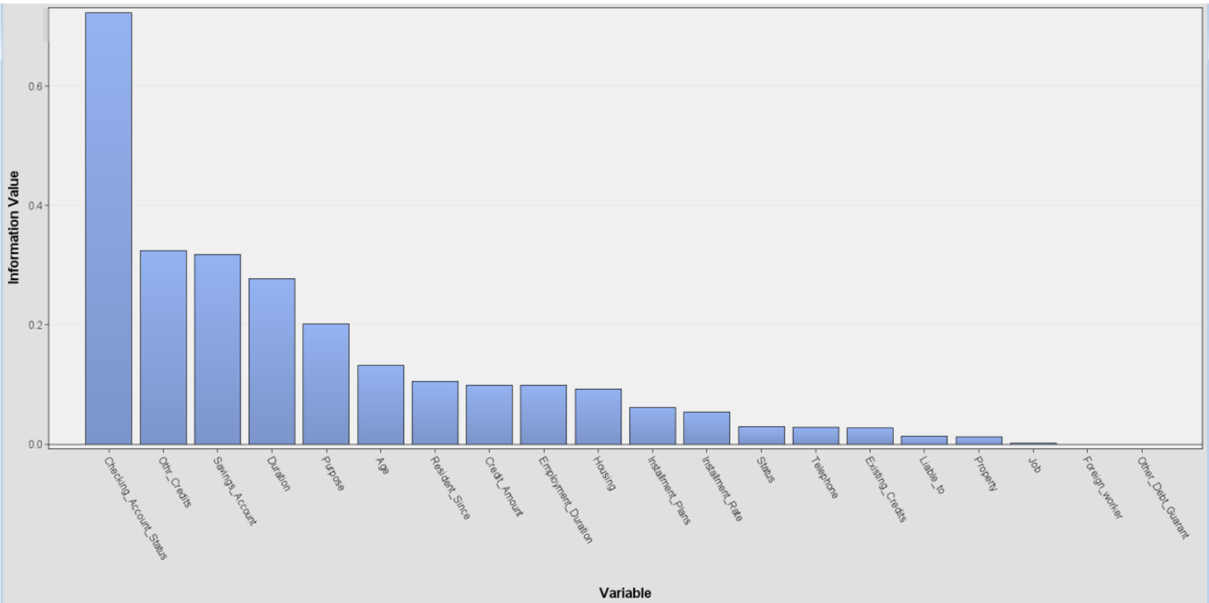
Here we can see that data is partitioned In Train and Validate with almost **equal percentage of Good and bad customers.** So we can **proceed**.

Group the Characteristic Variables into Attributes

We will use Interactive Grouping node to perform variable grouping which is a binning transformation on the input variables. Variable grouping is also called as classing.

1. From the Credit Scoring tab **drag an Interactive Grouping node** into the Workspace. **Connect** the Data Partition node to the Interactive Grouping node. Interactive Grouping node will performs the initial grouping automatically. These initial groupings can be used as a starting point to modify the classes interactively. By default the unbinned interval variables are grouped into 20 bins which are then grouped based on a decision tree. The Interactive Grouping node allows us to specify the properties of that decision tree.
2. **Select** Interactive Grouping node. **Set** value of Interval Grouping Method property and Ordinal Grouping Method property to Monotonic Event Rate. **Set** the value of the Maximum Number of Groups property to 10.
3. **Right-click** the Interactive Grouping node and **select Run**. In the Confirmation window that appears, **click Yes**. In the Run Status window that appears. **click Results**.





Output Variables

Variable	Gini Statistic	Information Value	Level for Interactive	Calculated Role	New Role	Pre-Defined Grouping	Level	Label	Information Value Ordering
Checking_Account_Status	41.937	0.724	NOMINAL	Input	Default		NOMINAL		1
Othr_Credits	28.026	0.325	NOMINAL	Input	Default		NOMINAL		2
Savings_Account	23.848	0.318	NOMINAL	Input	Default		NOMINAL		3
Duration	26.48	0.277	INTERVAL	Input	Default		INTERVAL		4
Purpose	24.549	0.202	NOMINAL	Input	Default		NOMINAL		5
Age	14.557	0.132	INTERVAL	Input	Default		INTERVAL		6
Resident_Since	15.825	0.105	NOMINAL	Input	Default		NOMINAL		7
Credit_Amount	11.693	0.099	INTERVAL	Rejected	Default		INTERVAL		8
Employment_Duration	13.984	0.099	NOMINAL	Rejected	Default		NOMINAL		9
Housing	13.936	0.093	NOMINAL	Rejected	Default		NOMINAL		10
Installment_Plans	9.143	0.062	NOMINAL	Rejected	Default		NOMINAL		11
Installment_Rate	11.927	0.054	INTERVAL	Rejected	Default		INTERVAL		12
Status	8.607	0.029	NOMINAL	Rejected	Default		NOMINAL		13
Telephone	8.117	0.028	NOMINAL	Rejected	Default		NOMINAL		14
Existing_Credits	7.722	0.027	INTERVAL	Rejected	Default		INTERVAL		15
Liable_to	4.211	0.013	INTERVAL	Rejected	Default		INTERVAL		16
Property	5.708	0.012	INTERVAL	Rejected	Default		INTERVAL		17
Job	2	0.002	NOMINAL	Rejected	Default		NOMINAL		18
Foreign_worker	0	0	NOMINAL	Rejected	Default		NOMINAL		19
Other_Debt_Guarant	0	0	NOMINAL	Rejected	Default		NOMINAL		20

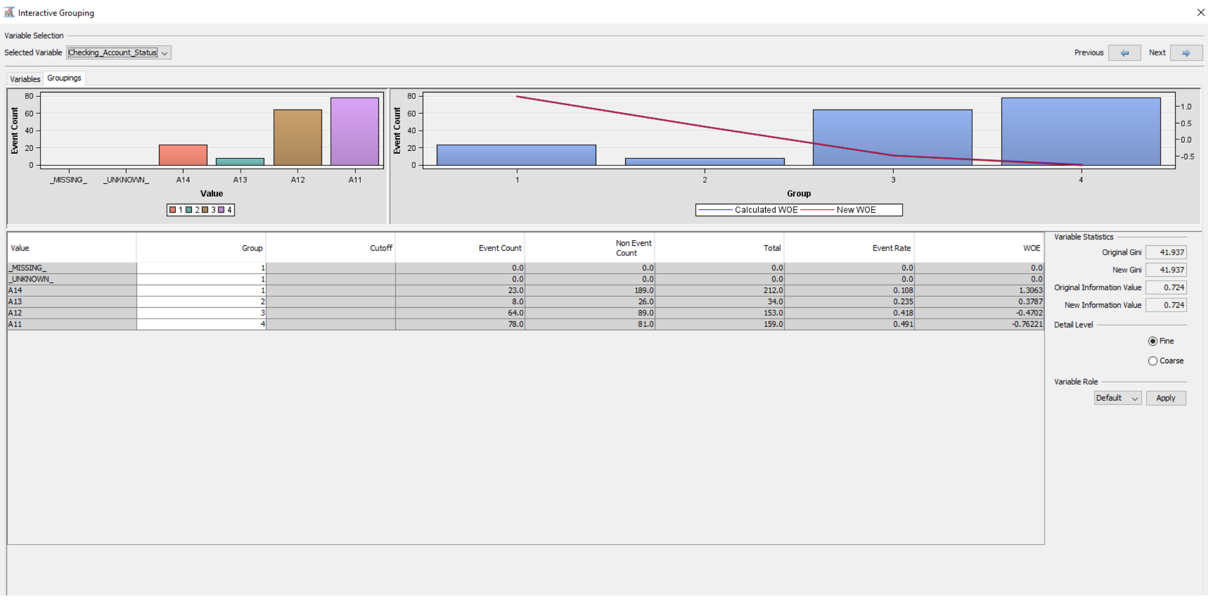
Output Variables window is displaying each variable’s Gini Statistic and information value (IV). A variable receives an Exported Role of Rejected if the variable’s IV is less than 0.10. IV is used to evaluate a characteristic’s overall predictive power (that is, the characteristic’s ability to separate between good and bad loans). Information value is calculated as follows:

$$IV = \sum_{i=1}^{L} \left(DistrGood_i - DistrBad_i \right) \cdot \ln \left(\frac{DistrGood_i}{DistrBad_i} \right)$$

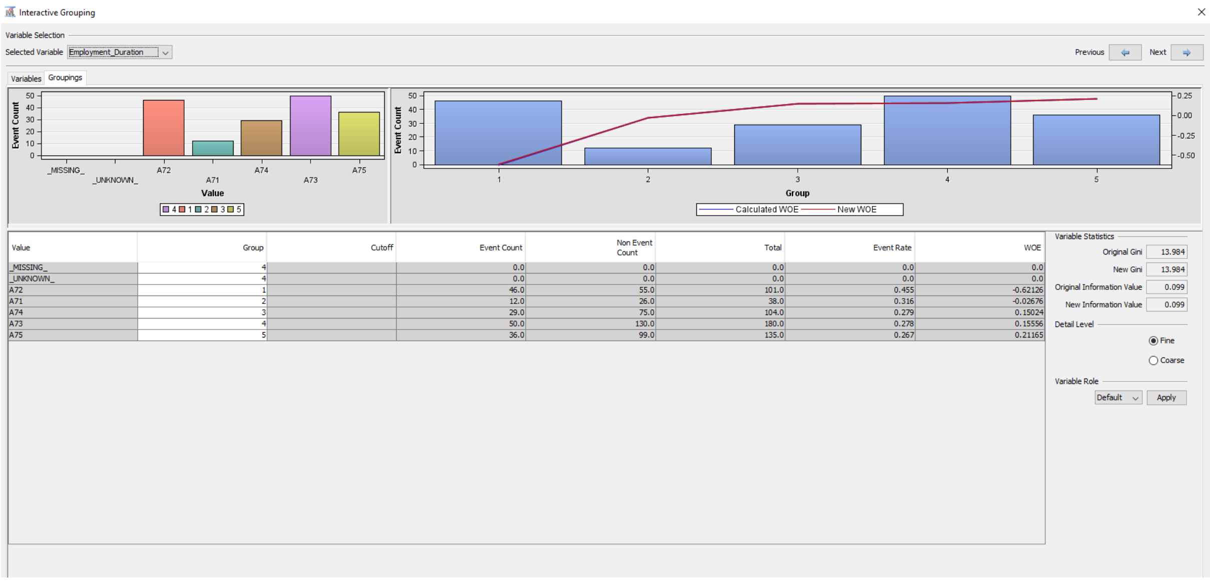
Here L is the number of attributes for the characteristic variable. In general an IV less than 0.02 is unpredictive, a value between 0.02 and 0.10 is weakly predictive, a value between 0.10 and 0.30 is moderately predictive, and a value greater than 0.30 is strongly predictive. The Gini statistic is used as an alternative to the IV.. In the Properties Panel of the Interactive Grouping node, we can specify the cutoff values for the Gini and IV statistics. For example, the default IV cutoff of 0.10 for rejecting a characteristic can be changed to another value using the Information Cutoff Value property.

Based on the IV, the variables **AGE, Duration, Checking_Account_Status, othr_Credits, Purpose, Resident_Since and Savings_Account** are considered the candidate inputs to build the final scorecard in the regression step. The IV and Gini statistics might change if the groupings of the attributes are changed. The initial, automatic binning provides a good starting point to create the groupings for the scorecard, but the groupings can be fine-tuned if we have the SME expertise available. **Close** the result window.

4. Open the Interactive Grouping application. Click the  button in the Interactive Grouping property. This opens the Interactive Grouping window.
5.



By default variables are sorted by their information value. Given in Original Information Value column. Variable that is selected by default is the variable with the greatest IV. Here that variable is **Checking_Account_Status**. Use the drop-down menu in the upper left corner of the Interactive Grouping window to select the variable **Employment_Duration**. The variable **Employment_Duration** represents the applicant's time at their current job. Select the Groupings tab in the Interactive Grouping window.



The plot on the right shows the weights of evidence for each group of the variable. Weight of evidence (WOE) measures the strength of an attribute of a characteristic in differentiating good and bad accounts. Weight of evidence is based on the proportion of good applicants to bad applicants at each group level. For each group i of a characteristic, WOE is calculated as follows:

$$WOE = \ln\left(\frac{DistrGood_i}{DistrBad_i}\right)$$

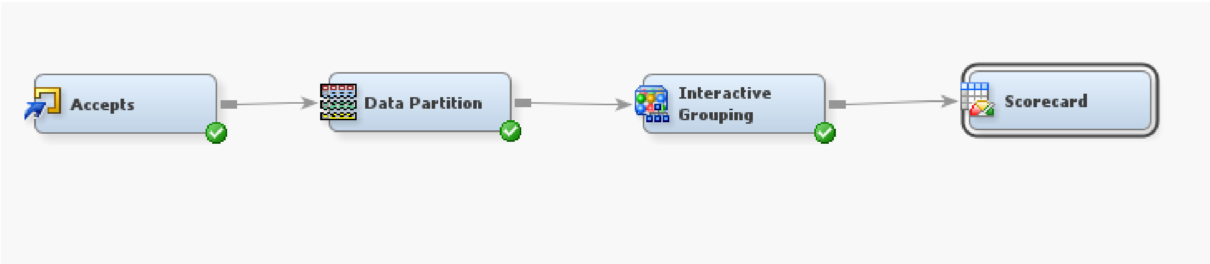
Negative values indicate that a particular grouping is isolating a higher proportion of bad applicants than good applicants. Which means negative WOE values are worse in the sense that applicants in that group present a greater credit risk. By default missing Values are assigned to their own group. The shape of the WOE curve is Representative of how the points in the scorecard are assigned. As we can see on the Groupings tab as time on the job increases WOE also increases. The plot on the left shows details of each group for the selected variable. It shows the distribution of the bad loans within each group.

Close the Interactive Grouping window

Create a Scorecard with a Logistic Regression Model

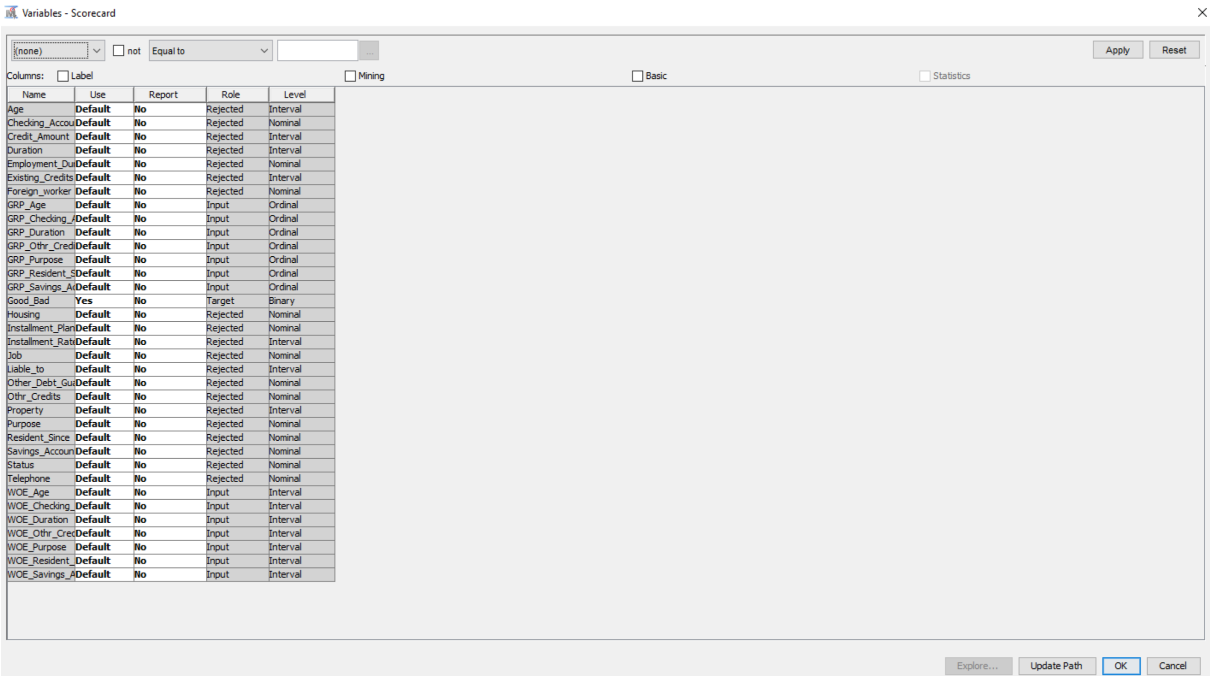
1. From the Credit Scoring tab, drag a Scorecard node to the Diagram Workspace.

Connect the Interactive Grouping node to the Scorecard node.



Scorecard node is used to develop a preliminary scorecard with logistic regression and scaling. In SAS Enterprise Miner there are three types of logistic regression selection methods to choose: forward, backward, and stepwise. There is also a selection in the Properties Panel of the Scorecard node for no selection method, so that all variable inputs enter the model. After the selection method is chosen, the regression coefficients are used to scale the scorecard. Scaling a scorecard refers to making the scorecard conform to a particular range of scores. Some reasons for scaling the scorecard are to enhance ease of interpretation, to meet legal requirements, and to have a transparent methodology that is widely understood among all users of the scorecard.

2. In the Diagram Workspace, select the Scorecard node. Click the  button in the Variables property.



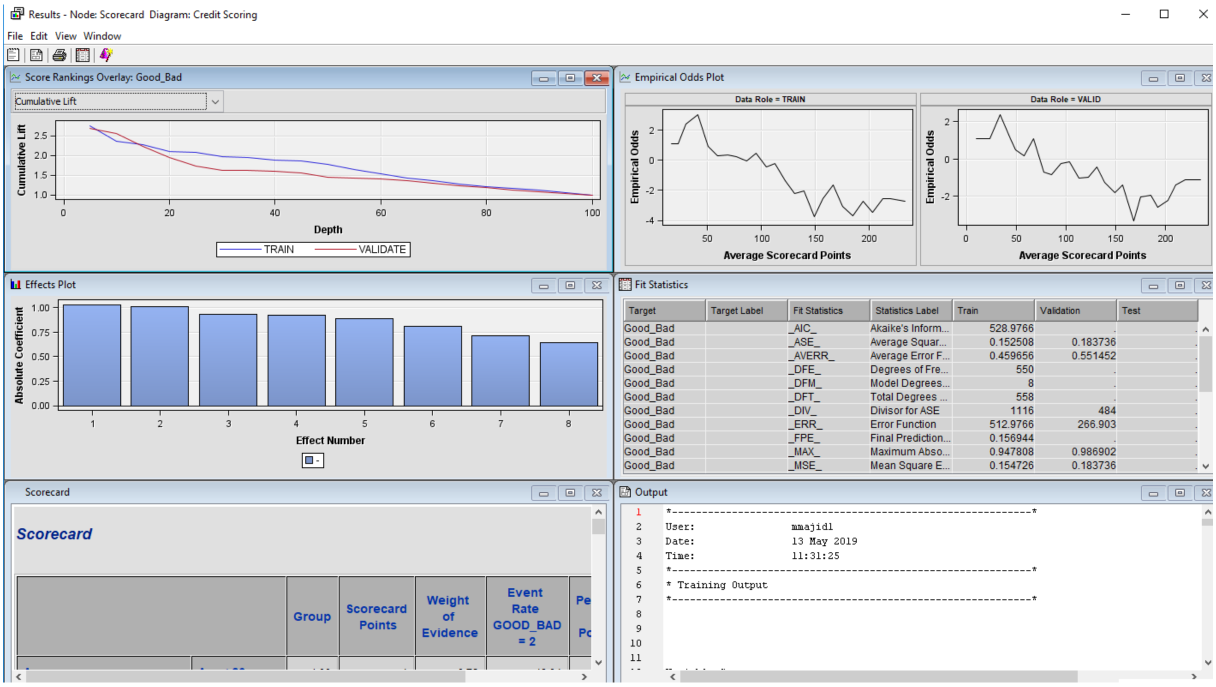
Name	Use	Report	Role	Level
Age	Default	No	Rejected	Interval
Checking_Accou	Default	No	Rejected	Nominal
Credit_Amount	Default	No	Rejected	Interval
Duration	Default	No	Rejected	Interval
Employment_Du	Default	No	Rejected	Nominal
Existing_Credits	Default	No	Rejected	Interval
Foreign_worker	Default	No	Rejected	Nominal
GRP_Age	Default	No	Input	Ordinal
GRP_Checking_A	Default	No	Input	Ordinal
GRP_Duration	Default	No	Input	Ordinal
GRP_Othr_Credi	Default	No	Input	Ordinal
GRP_Purpose	Default	No	Input	Ordinal
GRP_Resident_S	Default	No	Input	Ordinal
GRP_Savings_Ac	Default	No	Input	Ordinal
Good_Bad	Yes	No	Target	Binary
Housing	Default	No	Rejected	Nominal
Installment_Plan	Default	No	Rejected	Nominal
Installment_Rate	Default	No	Rejected	Interval
Job	Default	No	Rejected	Nominal
Liable_to	Default	No	Rejected	Interval
Other_Debt_Gua	Default	No	Rejected	Nominal
Othr_Credits	Default	No	Rejected	Nominal
Property	Default	No	Rejected	Interval
Purpose	Default	No	Rejected	Nominal
Resident_Since	Default	No	Rejected	Nominal
Savings_Accoun	Default	No	Rejected	Nominal
Status	Default	No	Rejected	Nominal
Telephone	Default	No	Rejected	Nominal
WOE_Age	Default	No	Input	Interval
WOE_Checking_	Default	No	Input	Interval
WOE_Duration	Default	No	Input	Interval
WOE_Othr_Cred	Default	No	Input	Interval
WOE_Purpose	Default	No	Input	Interval
WOE_Resident_	Default	No	Input	Interval
WOE_Savings_A	Default	No	Input	Interval

We can observe that for each original input variable there is now corresponding WOE and GRP variables. These were created by the Interactive Grouping node. Only the variables that exceed the Gini or IV cutoff set in the Interactive Grouping node are set to Input. All other inputs are set to Rejected.

Using the Scorecard node, we can use either the WOE variables, which contain the weight of evidence of each binned variable, or the GRP_ variables, which contain the group ID. The Analysis Variables property of the Scorecard node is used to specify whether regression is using WOE variables or GRP variables. The default is to use WOE variables. **Close** the Variables window.

3. **Change** the value of the Scorecard Type field to Detailed.

4. **Right-click** the Scorecard node and **click Run**. In the Confirmation window, **click Yes**. In the Run Status window, **click Results**.



Maximize the Fit Statistics window. The Fit Statistics window displays fit statistics such as the **average square error (ASE)**, the **area under the receiver operating characteristic curve (AUR)**, and the **Kolmogorov-Smirnov (KS) statistic**. The AUR is 0.747146 for the Validation data set.

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
Good_Bad		_AIC_	Akaike's Information Criterion	528.9766	.	.
Good_Bad		_ASE_	Average Squared Error	0.152508	0.183736	.
Good_Bad		_AVERR_	Average Error Function	0.459656	0.551452	.
Good_Bad		_DFE_	Degrees of Freedom for Error	550	.	.
Good_Bad		_DFM_	Model Degrees of Freedom	8	.	.
Good_Bad		_DFT_	Total Degrees of Freedom	558	.	.
Good_Bad		_DIV_	Divisor for ASE	1116	484	.
Good_Bad		_ERR_	Error Function	512.9766	266.903	.
Good_Bad		_FPE_	Final Prediction Error	0.156944	.	.
Good_Bad		_MAX_	Maximum Absolute Error	0.947808	0.986902	.
Good_Bad		_MSE_	Mean Square Error	0.154726	0.183736	.
Good_Bad		_NOBS_	Sum of Frequencies	558	242	.
Good_Bad		_NW_	Number of Estimate Weights	8	.	.
Good_Bad		_RASE_	Root Average Sum of Squares	0.390522	0.428645	.
Good_Bad		_RFPE_	Root Final Prediction Error	0.396162	.	.
Good_Bad		_RMSE_	Root Mean Squared Error	0.393352	0.428645	.
Good_Bad		_SBC_	Schwarz's Bayesian Criterion	563.5714	.	.
Good_Bad		_SSE_	Sum of Squared Errors	170.1984	88.92831	.
Good_Bad		_SUMW_	Sum of Case Weights Times Freq	1116	484	.
Good_Bad		_MISC_	Misclassification Rate	0.234767	0.297521	.
Good_Bad		_KS_	Kolmogorov-Smirnov Statistic	0.569192	0.384115	.
Good_Bad		_AUR_	Area Under ROC	0.829532	0.747146	.
Good_Bad		_Gini_	Gini Coefficient	0.659065	0.494293	.
Good_Bad		_ARATIO_	Accuracy Ratio	0.659065	0.494293	.

Maximize the Scorecard window. Initial Scorecard displays information such as the scorecard points for each attribute, WOE, event rate which is percentage of bad applicants in that score range, percentage of population, and the regression coefficient for each attribute. The Percentage of Population is the percentage of bad applicants who have a score higher than the lower limit of the score range.

Scorecard

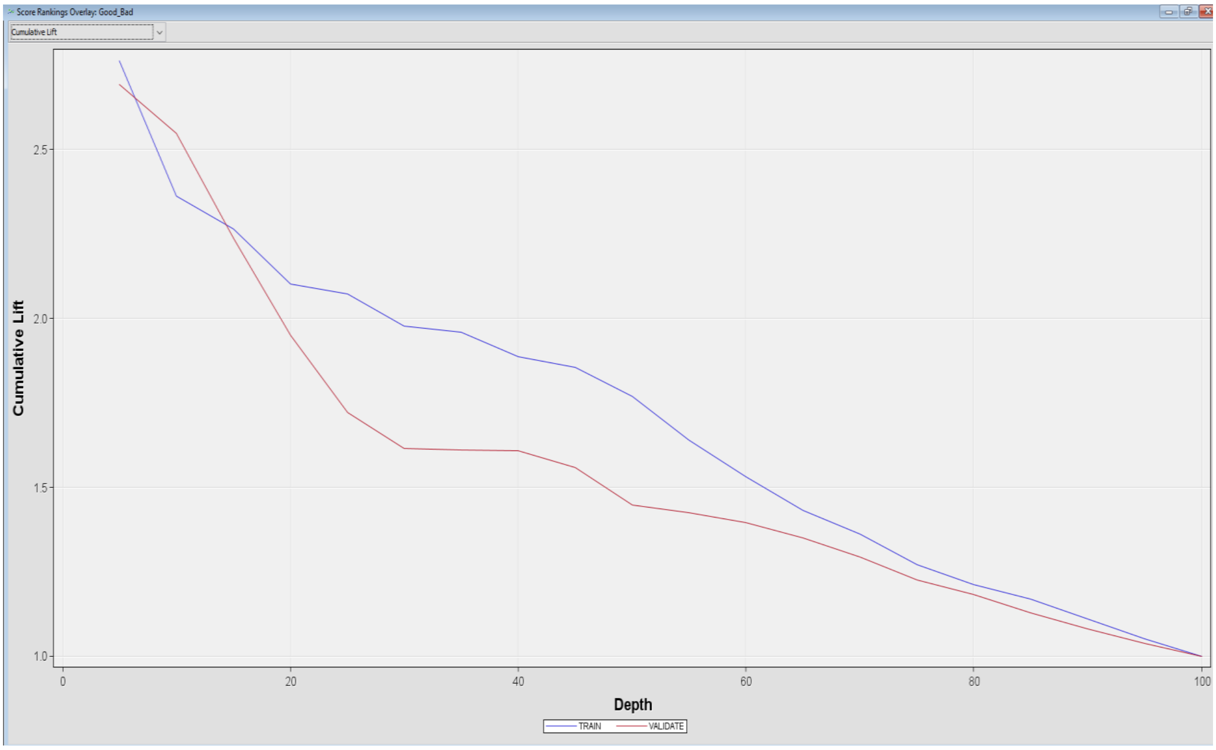
		Group	Scorecard Points	Weight of Evidence	Event Rate GOOD_BAD = 2	Percentage of Population	Coefficient
Age	Age< 26	1.00	-4	-0.72	48.04	18.28	-0.93
	26<= Age, _MISSING_	2.00	21	0.18	27.19	81.72	-0.93
Checking_Account_Status	A14, _MISSING_, _UNKNOWN_	1.00	43	1.31	10.85	37.99	-0.71
	A13	2.00	24	0.38	23.53	6.09	-0.71
	A12	3.00	6	-0.47	41.83	27.42	-0.71
	A11	4.00	0	-0.76	49.06	28.49	-0.71
Duration	Duration< 9	1.00	56	1.35	10.42	8.60	-1.03
	9<= Duration< 12	2.00	37	0.72	18.00	8.96	-1.03
	12<= Duration< 18, _MISSING_	3.00	21	0.18	27.27	25.63	-1.03
	18<= Duration< 24	4.00	14	-0.07	32.56	15.41	-1.03
	24<= Duration< 36	5.00	11	-0.15	34.29	25.09	-1.03
	36<= Duration< 48	6.00	1	-0.51	42.86	10.04	-1.03
	48<= Duration	7.00	-17	-1.09	57.14	6.27	-1.03
Othr_Credits	A30, A31	1.00	-12	-1.09	57.14	8.78	-0.88
	A32, _MISSING_, _UNKNOWN_	2.00	11	-0.20	35.35	53.23	-0.88
	A33	3.00	18	0.08	29.41	9.14	-0.88
	A34	4.00	39	0.89	15.53	28.85	-0.88
Purpose	A41, A48	1.00	32	0.57	20.31	11.47	-1.01
	A410, A43, _MISSING_, _UNKNOWN_	2.00	31	0.53	20.92	27.42	-1.01
	A44, A49	3.00	21	0.17	27.42	11.11	-1.01
	A42	4.00	10	-0.21	35.71	20.07	-1.01
	A40	5.00	4	-0.41	40.46	23.48	-1.01

In scorecard it can be observed that an applicant **aged** less than 26 has more chances to be bad so we are giving him less score than an applicant equal or older than 26.

An applicant for a **duration of less than 9 months** is getting the **highest scorecard points** as compared to others.

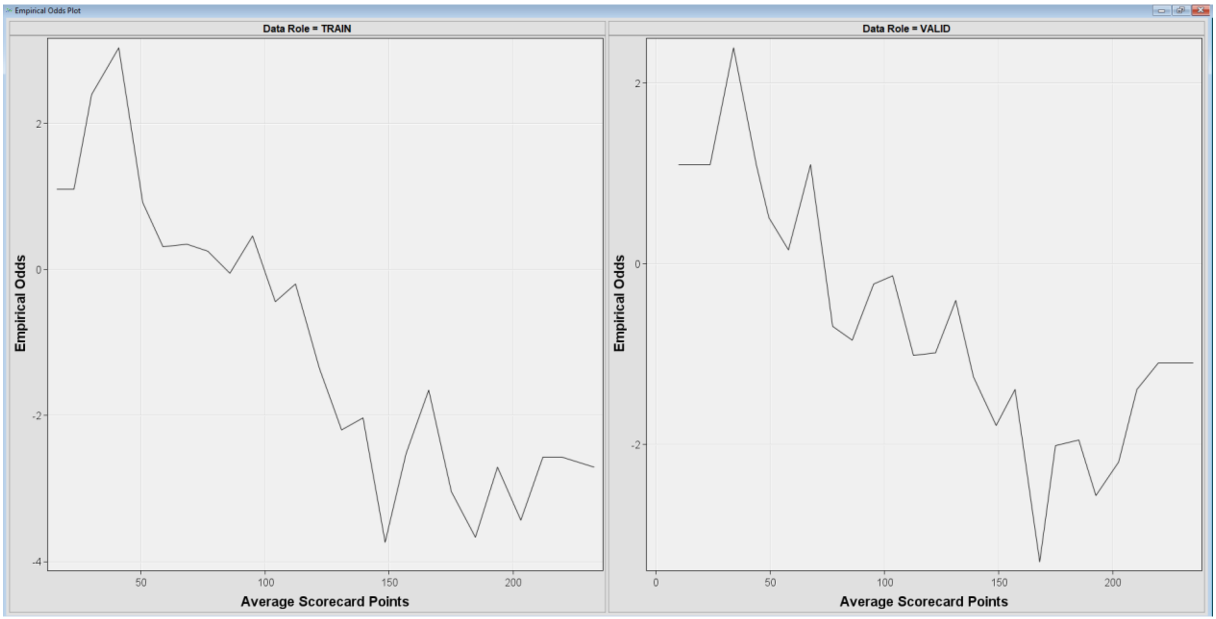
Now **maximize the Score Rankings Overlay window**. Score Rankings Overlay window plots the **Cumulative Lift chart by default**. Lift is the ratio of the percent of targets which is bad loans in each decile to the percent of targets in the entire data set. Cumulative lift is cumulative ratio of the percent of targets up to the decile of interest to the percent of targets in the entire data set.

For lift and cumulative lift, the higher value in the lower deciles indicates a predictive scorecard model. Notice that both Lift and Cumulative Lift for this scorecard have high lift values in the lower deciles.



Here we can see that the results of cumulative lift graphs are **satisfactory** as we can see we are getting almost **2 times average of number of bad applicants** if we select **20 percent** of the population. This performance of **validation set** is little bit poor which is a known difference.

Maximize the Empirical Odds Plot window. An empirical odds plot is usually used to evaluate the calibration of the scorecard. The chart plots the observed odds in a score bucket against the average score value in each bucket. The plot can help determine where the scorecard is or is not sufficiently accurate. The odds are calculated as the logarithm of the number of bad loans divided by the number of good loans for each scorecard bucket range. Thus, a steep negative slope implies that the good applicants tend to get higher scores than the bad applicants. As expected with the previous plot, as the scorecard points increase, so does the number of good loans in each score bucket.

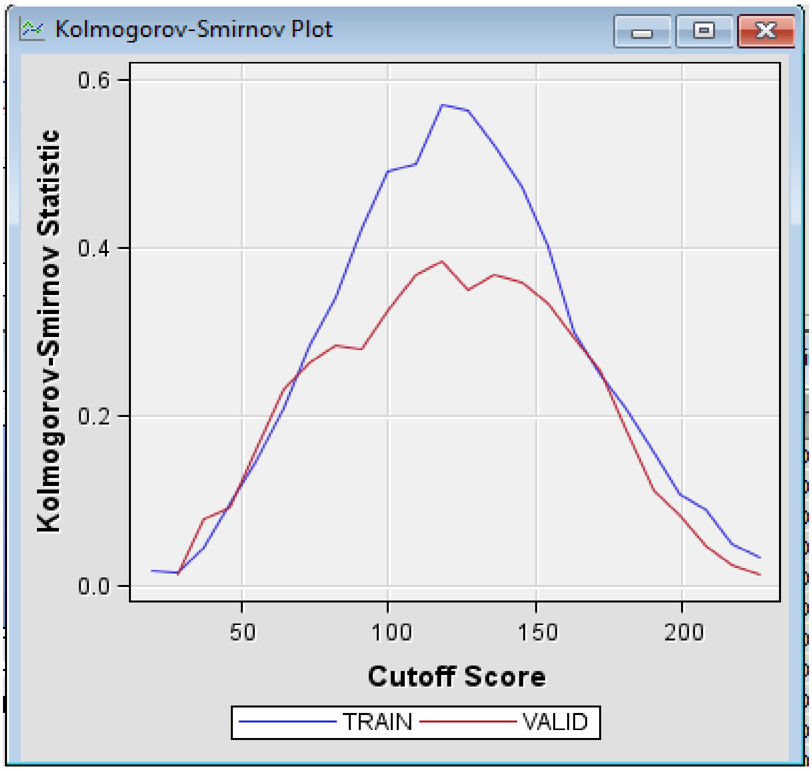


Here we can see that there is a **steep slope in both train and test set**. So **Good applicants are getting higher score than the bad applicants**. As the **Score card points increase, so does the number of good loan in each score bucket**.

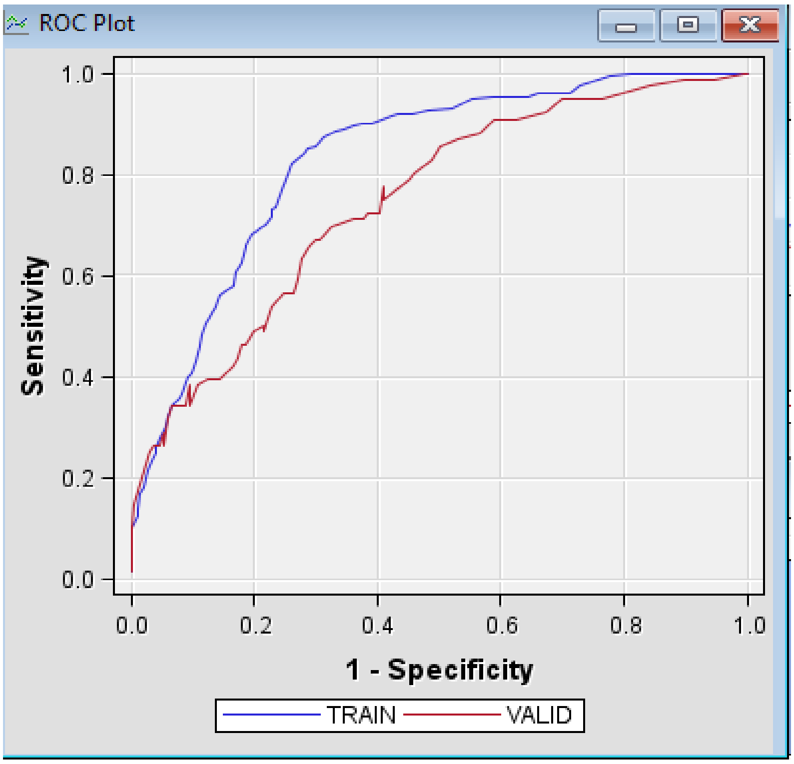
From the main menu, select **View Strength Statistics** than **select Kolmogorov-Smirnov Plot**. The Kolmogorov-Smirnov Plot shows the Kolmogorov-Smirnov statistics plotted against scorecard cutoff values. Kolmogorov-Smirnov statistic is the maximum distance between the empirical distribution functions for the good applicants and the bad applicants. The difference is plotted, for all cutoffs in the Kolmogorov-Smirnov Plot. The weakness of reporting only the maximum difference between the curves is that it provides only a measure of vertical separation at one cutoff value but not overall cutoff values.

According to the plot, **the best cutoff is approximately 125 (where the Kolmogorov-Smirnov score is at a maximum). At a cutoff value of 125, the scorecard best**

distinguishes between good and bad loans.



From the main menu, select View than select Strength Statistics and than click ROC Plot. The ROC plot is a graphical measure of sensitivity versus 1-specificity. The AUR which is close to 0.75 for the validation data from the previous Fit Statistics table measures the area below each of the curves that you see drawn in the plot. The AUR is generally known as providing better measure of the scorecard strength than the Kolmogorov-Smirnov statistic because the area being calculated encompasses all cutoff values. A scorecard that is no better than random selection has an AUR value equal to 0.50. The maximum value of the AUR is 1.0.



ROC curve is a plot of the true positive rate versus the false positive rate for all possible cut-off values. Which implies each point on the ROC curve represents a different cutoff value. The points are connected to form the curve. Cutoff values that result in low false-positive rates tend to result low true-positive rates as well. As the true-positive rate increases, the false positive rate increases. Here the true positive rate is more quickly nears 1 or 100 percent. So we are getting **good results** for both Train and Validation set. Our model is able to differentiate well between good and bad applicants.

Create Rejects Data Source

Drag and drop File Import node and rename it as Rejects as we will use rejects data here. Than In Import file in train property and upload Rejects file here after right click on the node and select edit variable. In property section change role from Train to Score. Select OK and run the node.

Perform Reject Inference on the Model

The preliminary scorecard that was designed before used known good and bad loans from only the accepted applicants. We have to apply scorecard to all applicants containing both accepted and rejected. Scorecard has to be generalized for "through the door" population. Now we will perform reject inference to solve the sample bias problem so that the developmental sample will be similar to the population to which the scorecard will be applied.

1. From the Credit Scoring tab, **drag a Reject Inference node** to the Diagram Workspace. Connect the Scorecard node to the Reject Inference node.

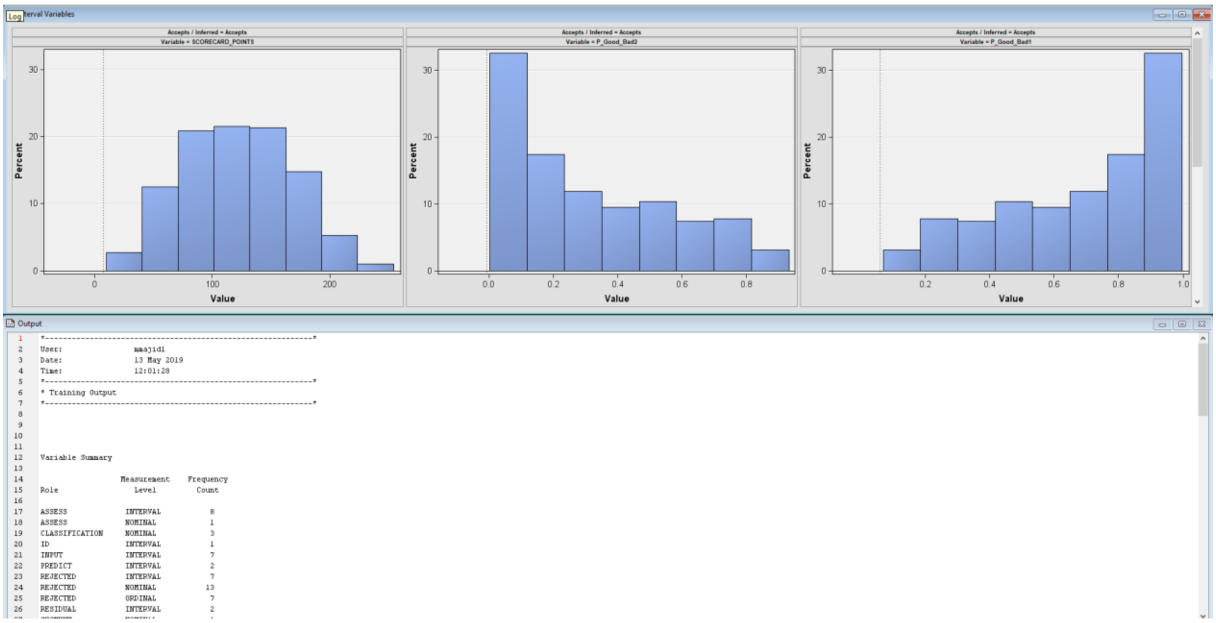
2. Connect the REJECTS data set to the Reject Inference node.

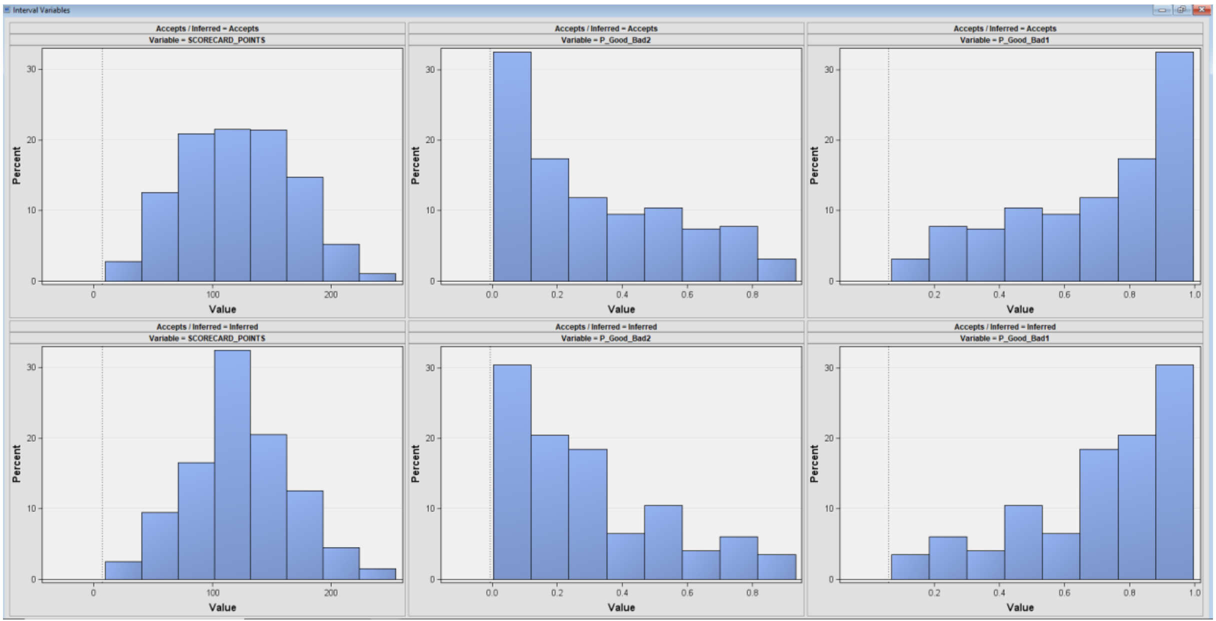
3. Here we are using the default inference method which is Fuzzy. Fuzzy classification uses partial classifications of "good" and "bad" to classify the rejected applicants in the augmented data set. Instead of classifying observations as "good" and "bad," fuzzy classification allocates weight to

observations in the augmented data set. The weight reflects the observation's tendency to be “good” or “bad.”

4. **Right-click the Reject Inference node** and **select Run**. In the Confirmation window**, click Yes**.

5. In the Run Status window, click **Results**.





The Results window includes distribution plots for the score and predicted probabilities for the accepted observations and the inferred samples side-by-side. **Expand** the Output window. Scroll down to the Reject Inference: Event Rates section to the event rates for various samples, including the augmented data.

Summary Statistics

_ACTUAL_INFERRED_=Accepts

Obs	VARIABLE	NMISS	MIN	MAX	MEAN	STD	SKEWNESS	KURTOSIS
1	P_Good_Bad1	0	0.0681	0.996	0.692	0.2525	-0.65012	-0.80232
2	P_Good_Bad2	0	0.0039	0.932	0.308	0.2525	0.65012	-0.80232
3	SCORECARD_POINTS	0	10.0000	247.000	121.928	45.9992	0.09639	-0.59989

_ACTUAL_INFERRED_=Inferred

Obs	VARIABLE	NMISS	MIN	MAX	MEAN	STD	SKEWNESS	KURTOSIS
4	P_Good_Bad1	0	0.1001	0.997	0.711	0.2356	-0.86641	-0.25833
5	P_Good_Bad2	0	0.0032	0.900	0.289	0.2356	0.86641	-0.25833
6	SCORECARD_POINTS	0	23.0000	254.000	124.050	43.9947	0.20713	0.00015

The output from the Reject Inference node is the augmented data containing both ACCEPTS and REJECTS appended together. The **Training Data Event Rate** and the **Validation Data Event Rate** are the event rates which is the bad rates for the accepted applicant's data set. The **Rejects Inferred Event Rate** is event rate for the rejected applicant's data set. The **Augmented Data Event Rate** is event rate for the augmented data set. The Summary Statistics sections display basic summary statistics for both the accepted and rejected applicants' data.

Reject Inference : Event Rates

Training Data Event Rate	Validation Data Event Rate	Rejects Inferred Event Rate	Augmented Data Event Rate
31.00	31.40	28.89	30.45

Here we can see that the event rate is almost same for all the four categories.

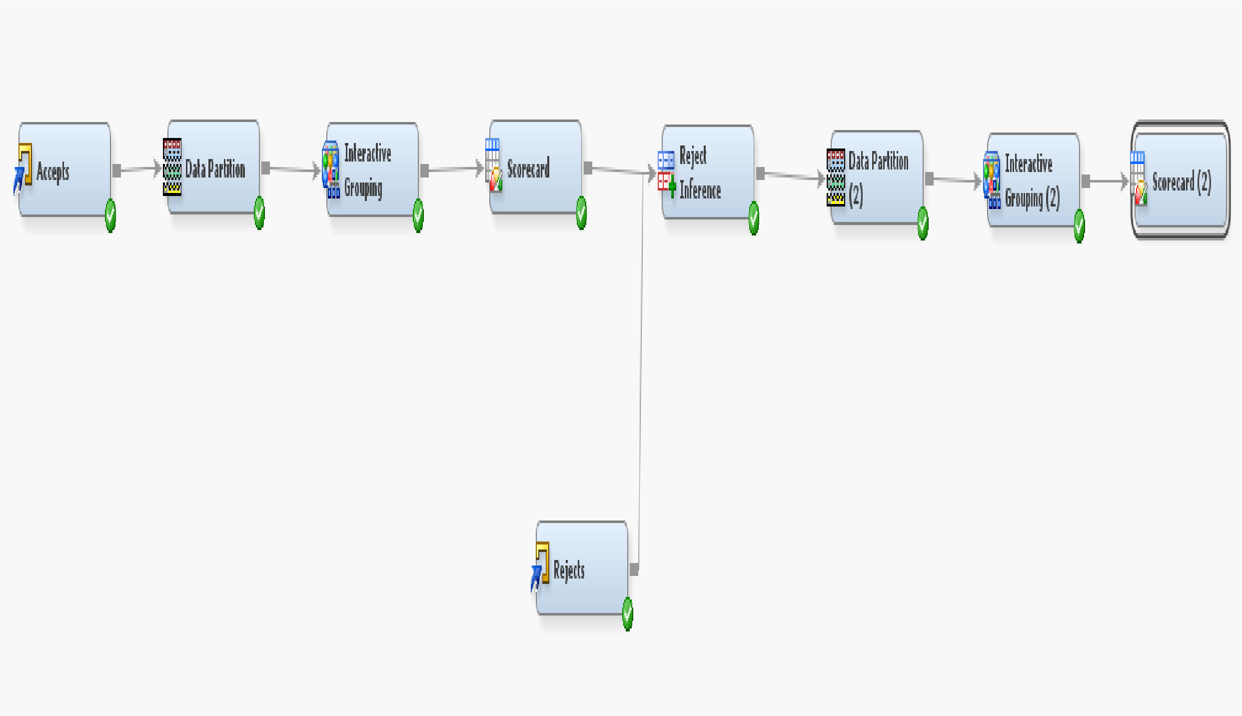
Create the Final Scorecard

In order to build final scorecard we repeat the data partitioning, grouping, and scorecard creation steps. We will do following steps on the augmented data set.

1. From the Sample tab, **drag a Data Partition node** to the Diagram Workspace**. Connect the Reject Inference node to the Data Partition (2) node**. In the Data Set Allocations property group set the value of Training to 70, Validation to 30, and Test to 0.

2. From the Credit Scoring tab**, drag an Interactive Grouping node to the Diagram Workspace**. **Connect** the Data Partition (2) node to the Interactive Grouping (2) node. This step will use the default values for the Interactive Grouping node.

3. From the Credit Scoring tab, **drag a Scorecard node** to the Diagram Workspace**. Connect** the Interactive Grouping (2) node to the Scorecard (2) node.



4. **Right-click** the Scorecard (2) node and **click Run**. In the Confirmation window **click Yes**. In the **Run Status window click Results**.We have now a scorecard that is based on both accepted and rejected applicants.

		Scorecard Points
Age	Age< 26	2
	26<= Age< 34, _MISSING_	13
	34<= Age< 42	26
	42<= Age< 47	12
	47<= Age	20
Checking_Account_Status	A11	-6
	A12	3
	A13	36
	A14, _MISSING_, _UNKNOWN_	39
Credit_Amount	Credit_Amount< 3590, _MISSING_	15
	3590<= Credit_Amount< 3959	18
	3959<= Credit_Amount< 4576	10
	4576<= Credit_Amount< 6887	13
	6887<= Credit_Amount	9
Duration	Duration< 8	59
	8<= Duration< 12	39
	12<= Duration< 18	20
	18<= Duration< 36, _MISSING_	11
	36<= Duration	-12
Housing	A152, _MISSING_, _UNKNOWN_	17
	A151	7
	A153	5
Othr_Credits	A34	30
	A33	11
	A32, _MISSING_, _UNKNOWN_	11
	A30, A31	-8
Purpose	A45, A49	1
	A40	5
	A42, A46	11
	A43, _MISSING_, _UNKNOWN_	24
	A41, A410, A44, A48	30
Savings_Account	A64, A65	37
	A63	24
	A61, _MISSING_, _UNKNOWN_	7
	A62	3

The variables selected for Scorecard are different than the initial one as our IV's were updated by inclusion of Rejects data from Reject inference node. From here we we can calculate the Application Score Card based on the applicant's credentials to be observed as mentioned in the scorecard and to decide a customer to be bad or good whether the customer has a score greater than threshold calculated from Kolmogorov-Smirnov Plot.

References:

- https://github.com/adriangasinski/datahacking_0001/blob/master/00_german_credit.R
- https://support.sas.com/documentation/cdl/en/emcsgs/66008/PDF/default/emcsgs.pdf
- Course Notes