WQD7005 Data Mining

Alternative Assessment 1 (G2)_SAS_Step

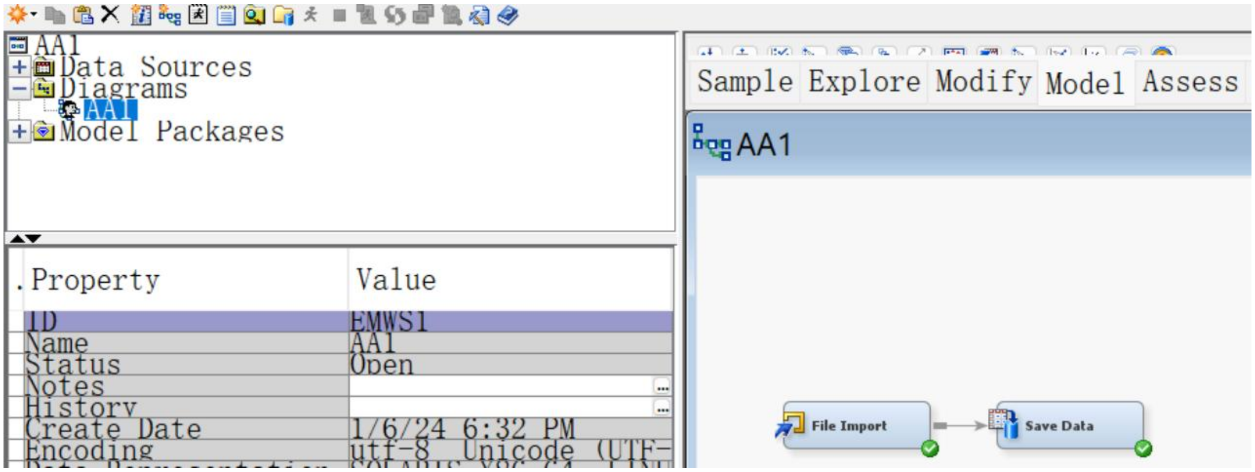
Matric ID	**Name**
22060214	Mei Zhu
Git Hub link	https://github.com/ChristineLzy/WQD7005_AA1

Table of Contents

1 Data Import and Preprocessing

1.1 Import dataset into SAS

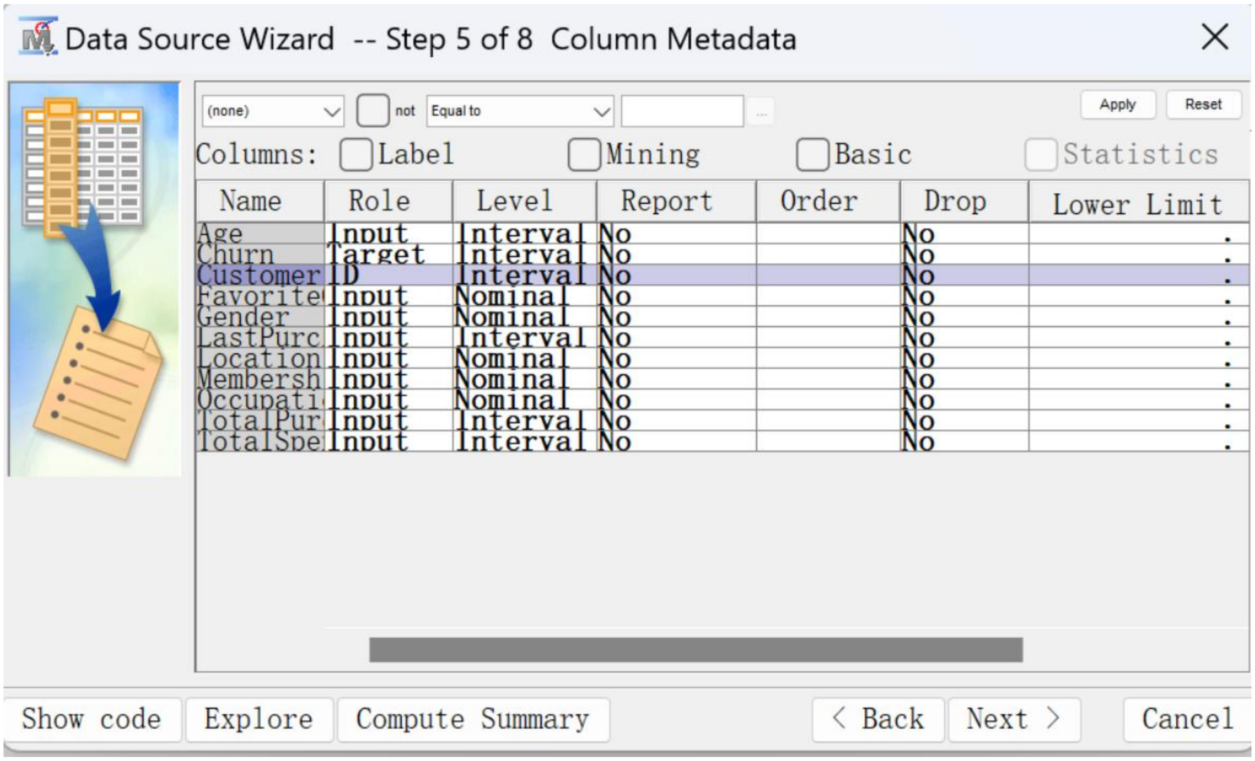
First, import the CSV dataset, and then use the "save data" node through drag-and-drop to save the CSV dataset as a .SAS dataset file.



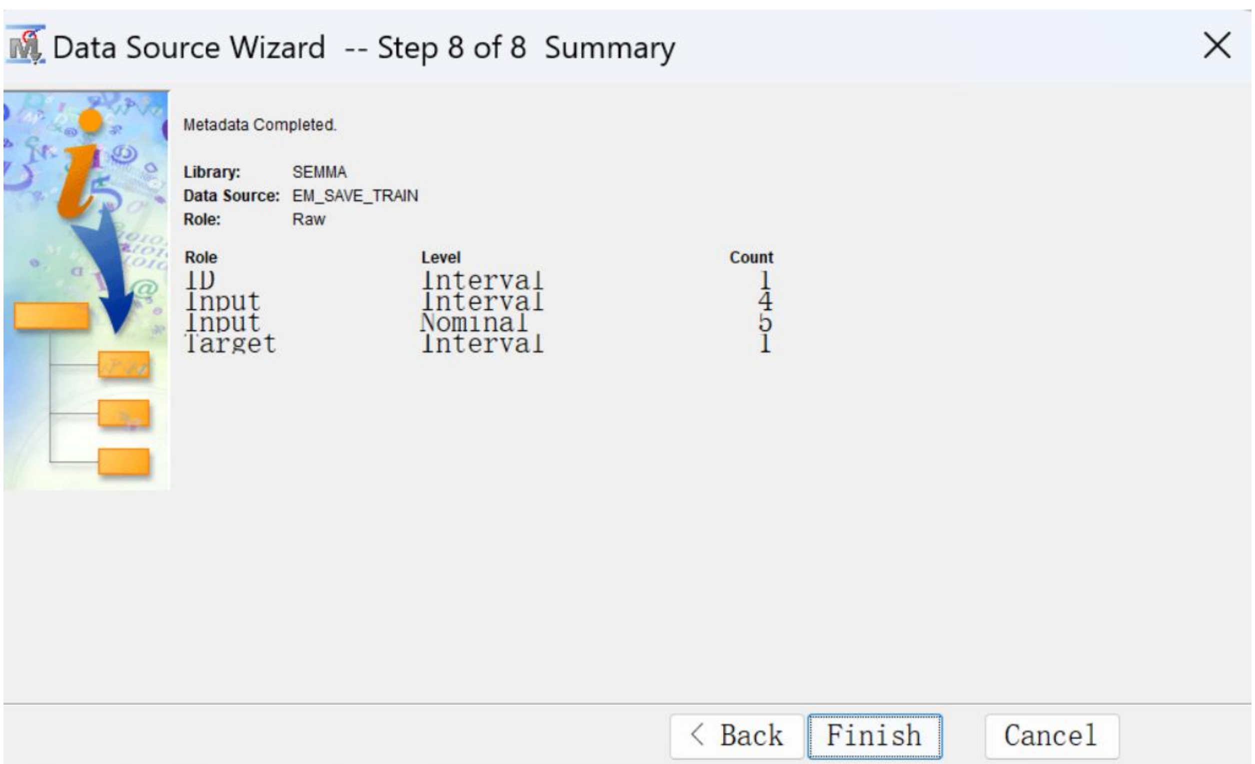
Create a new data source and configure variable roles.

In this dataset, designate "churn" as the target variable, set "customer" as the identifier (ID), and designate the remaining attributes as input, adjusting their levels accordingly to either interval or nominal.

1.2 Setting specify variable roles.

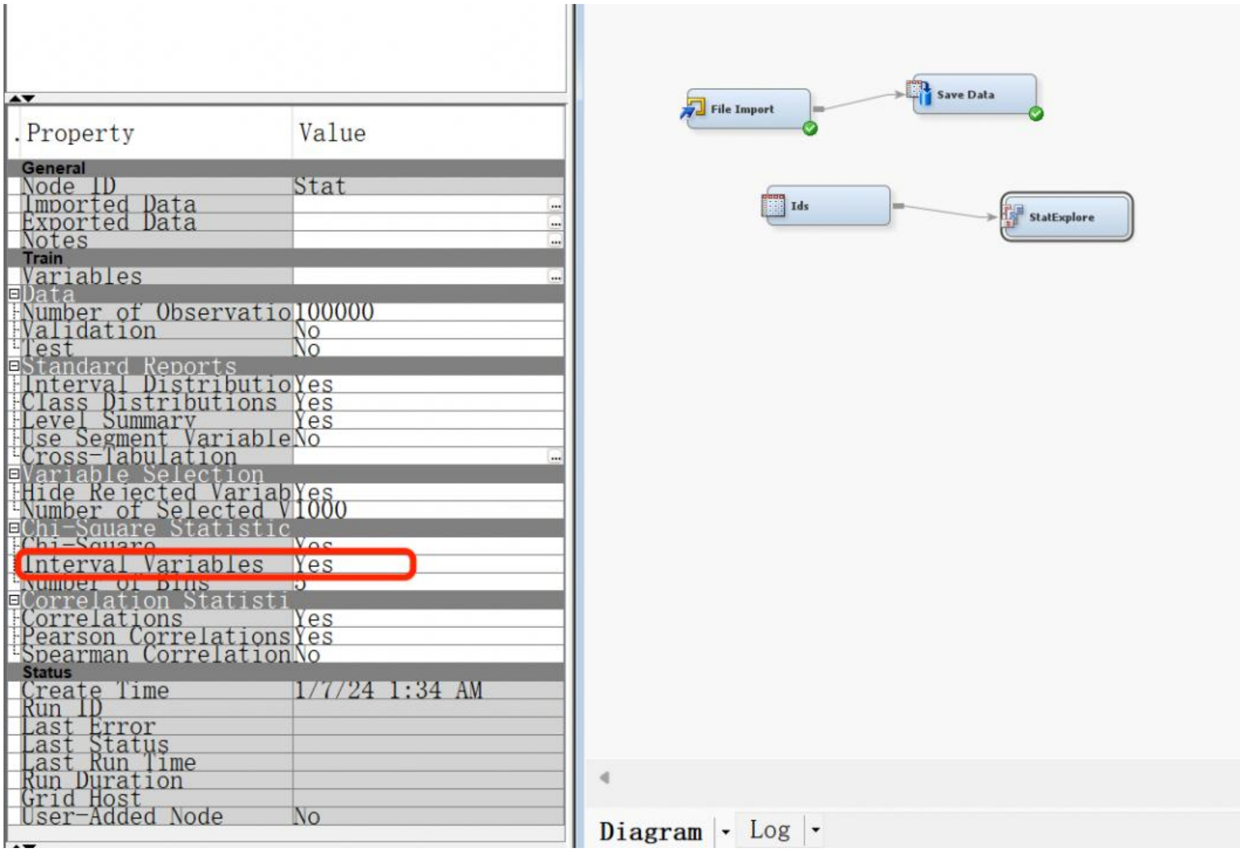


The configuration of variable roles in the "specify variable roles" is as follows:

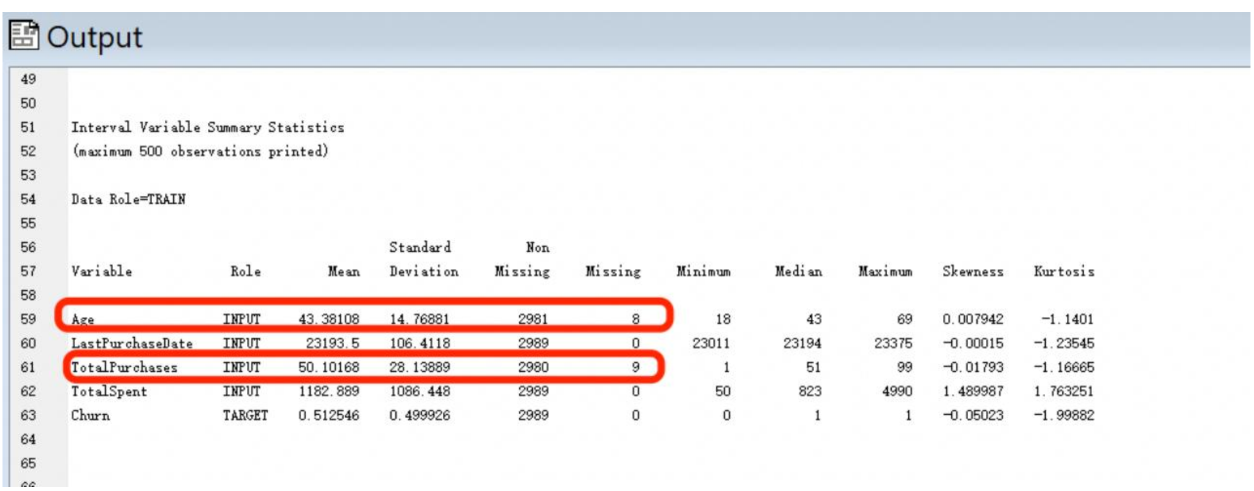


1.3 Handle missing values

After importing the data source, add a "StatExplore" node to examine the distribution of the data.



The results indicate that there are missing values in "age" and "totalpurchases," with 8 and 9 missing values respectively.

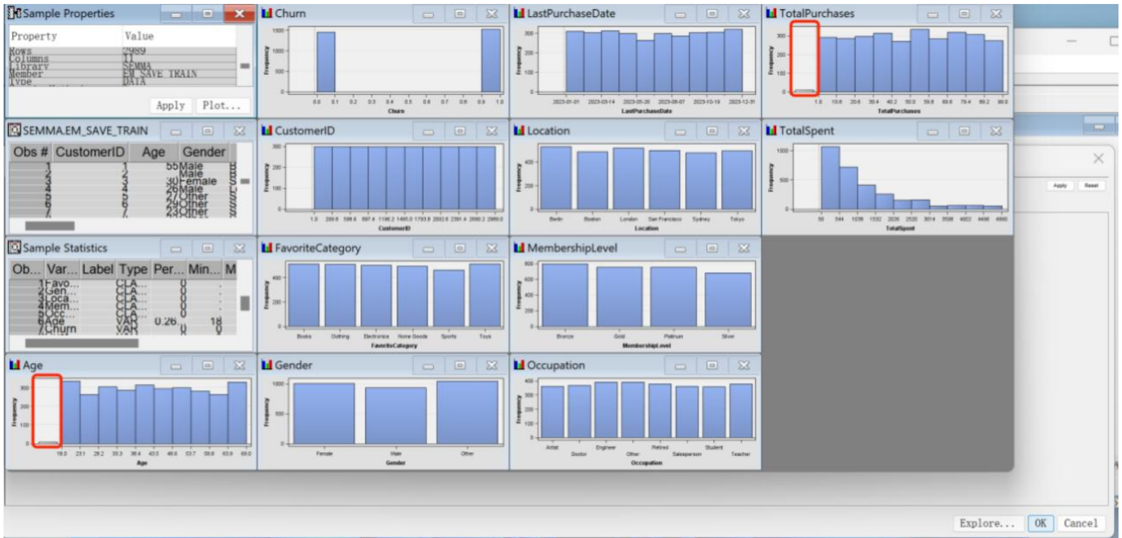

Output

Interval Variable Summary Statistics
(maximum 500 observations printed)

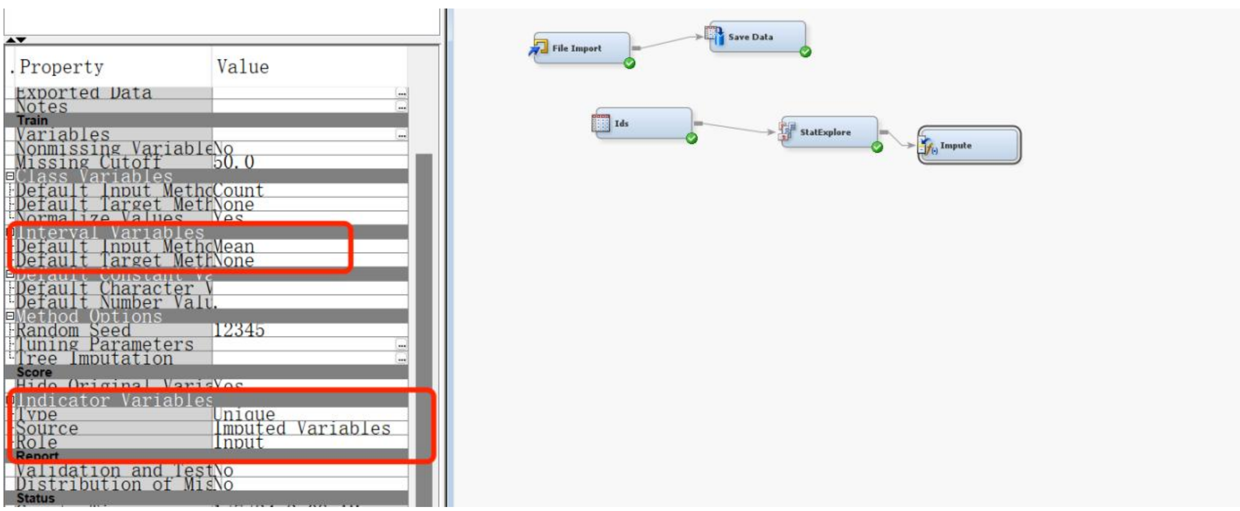
Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
Age	INPUT	43.38108	14.76881	2981	8	18	43	69	0.007942	-1.1401
LastPurchaseDate	INPUT	23193.5	106.4118	2989	0	23011	23194	23375	-0.00015	-1.23545
TotalPurchases	INPUT	50.10168	28.13889	2980	9	1	51	99	-0.01793	-1.16665
TotalSpent	INPUT	1182.889	1086.448	2989	0	50	823	4990	1.489987	1.763251
Churn	TARGET	0.512546	0.499926	2989	0	0	1	1	-0.05023	-1.99882

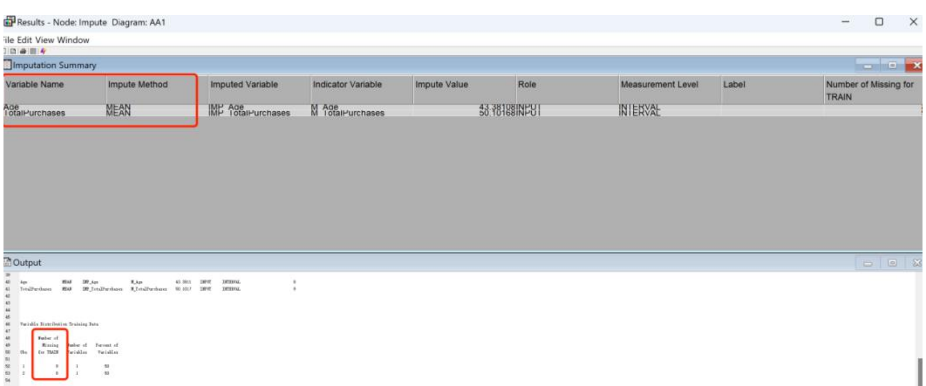
Right-click on the data source node, select "Edit Variable," and then click "Explore" to visualize the distribution of the data. Here, you can also intuitively observe the distribution of missing values. For instance, the gray portion in the histogram represents the missing values in "age" and "totalpurchases."



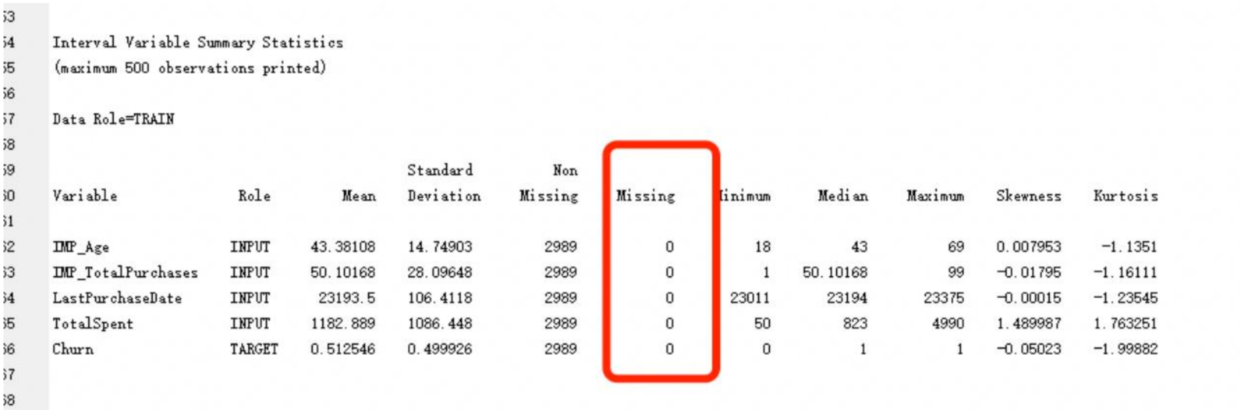
Add an "impute" node to replace missing values for interval-type data with the mean.



The results are as follows; only the "age" and "totalpurchases" columns were processed, and the number of missing values matches the previous records:



Recheck the situation of missing values. The results indicate that there are currently no missing values.


Interval Variable Summary Statistics
(maximum 500 observations printed)

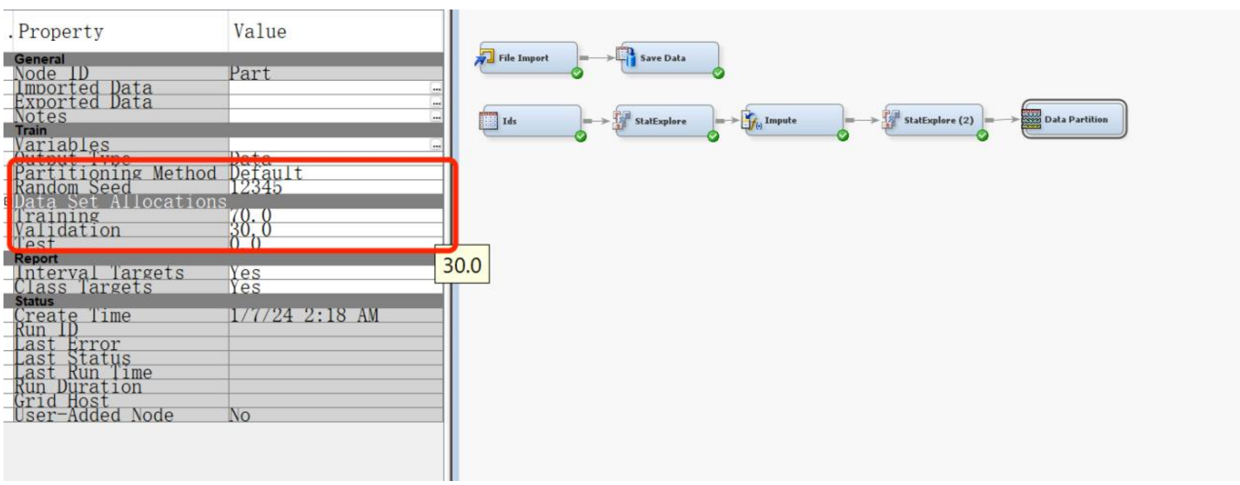
Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
IMP_Age	INPUT	43.38108	14.74903	2989	0	18	43	69	0.007953	-1.1351
IMP_TotalPurchases	INPUT	50.10168	28.09648	2989	0	1	50.10168	99	-0.01795	-1.16111
LastPurchaseDate	INPUT	23193.5	106.4118	2989	0	23011	23194	23375	-0.00015	-1.23545
TotalSpent	INPUT	1182.889	1086.448	2989	0	50	823	4990	1.489987	1.763251
Churn	TARGET	0.512546	0.499926	2989	0	0	1	1	-0.05023	-1.99882

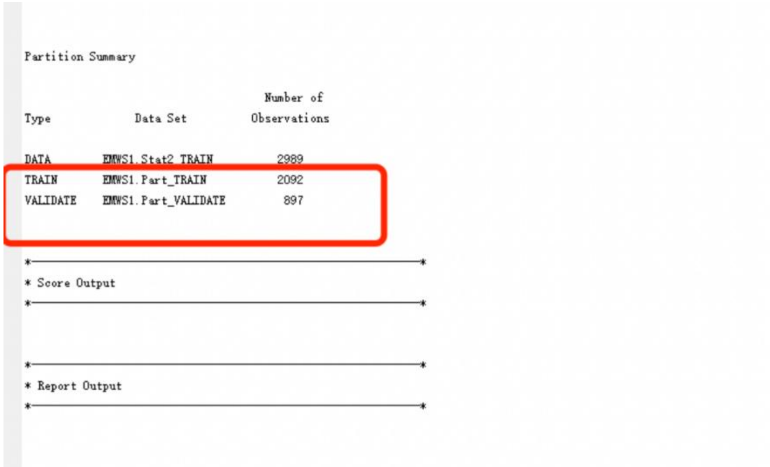
2 Decision Tree Analysis

2.1 Data Partition

Add a "data partition" node and set the training ratio to 70% and the validation ratio to 30%.



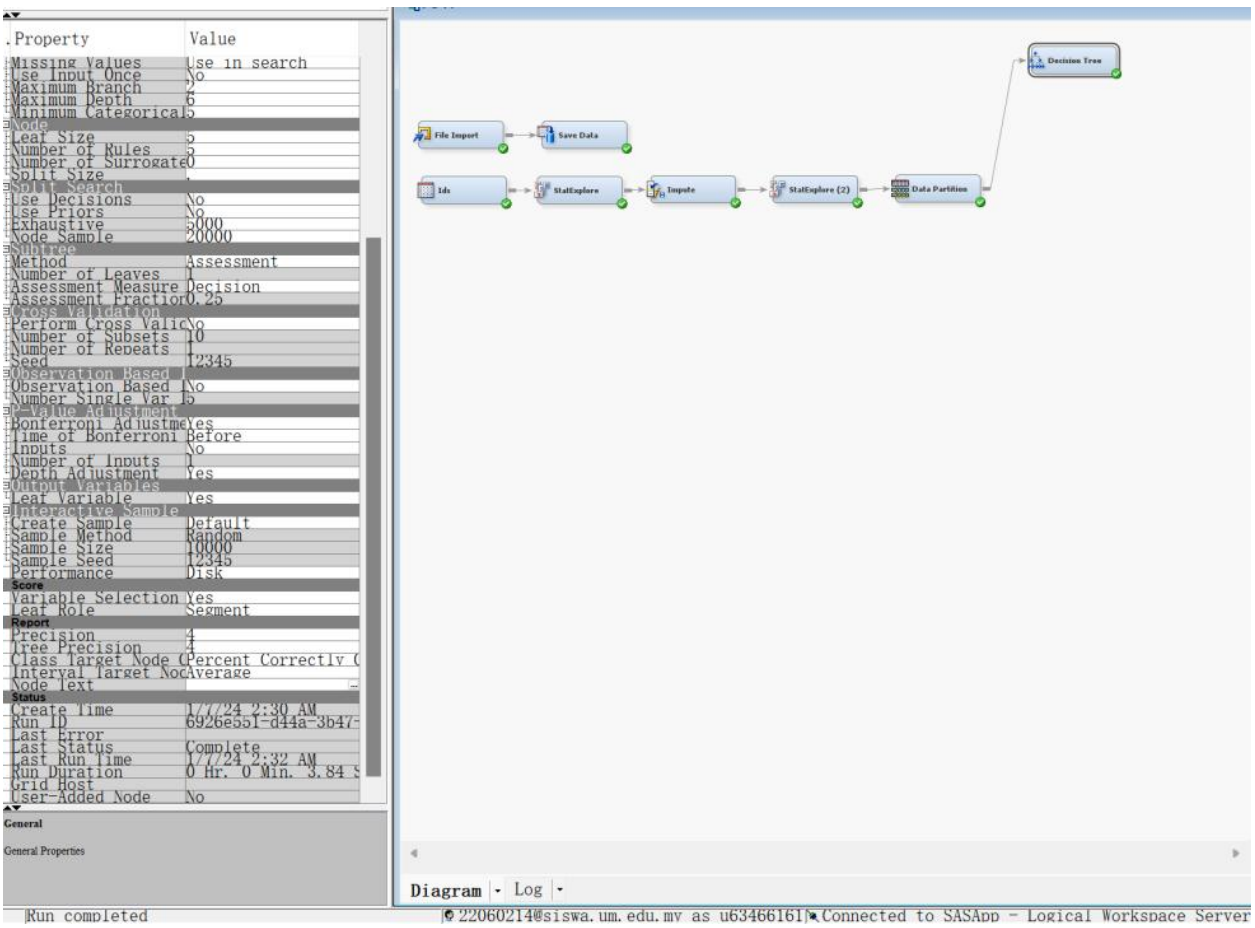
The execution results are as follows, indicating the distribution of the dataset into a 70:30 ratio for the training set and test set.



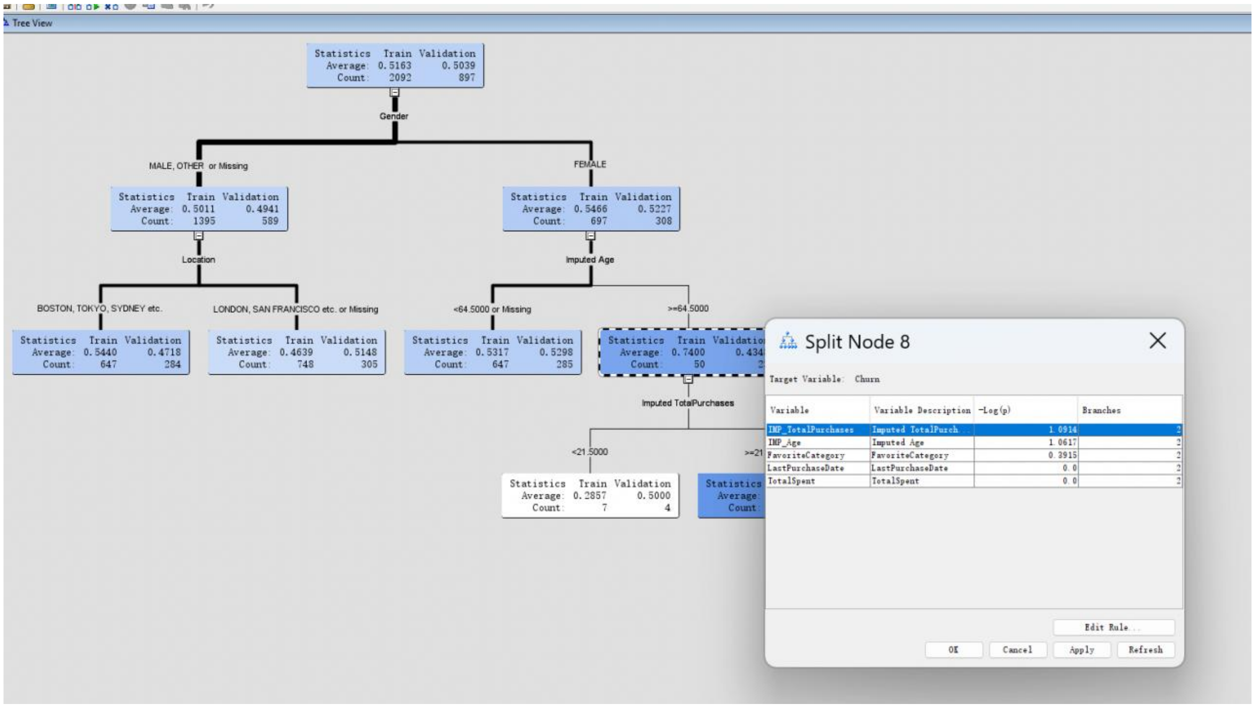
```
Partition Summary

                                  Number of
Type          Data Set          Observations

DATA       EMWS1.Stat2_TRAIN        2989
TRAIN      EMWS1.Part_TRAIN         2092
VALIDATE   EMWS1.Part_VALIDATE       897

*------------------------------------------------------------*
* Score Output
*------------------------------------------------------------*

*------------------------------------------------------------*
* Report Output
*------------------------------------------------------------*
```

2.2 Decision Tree Model

Add a node for the decision tree.

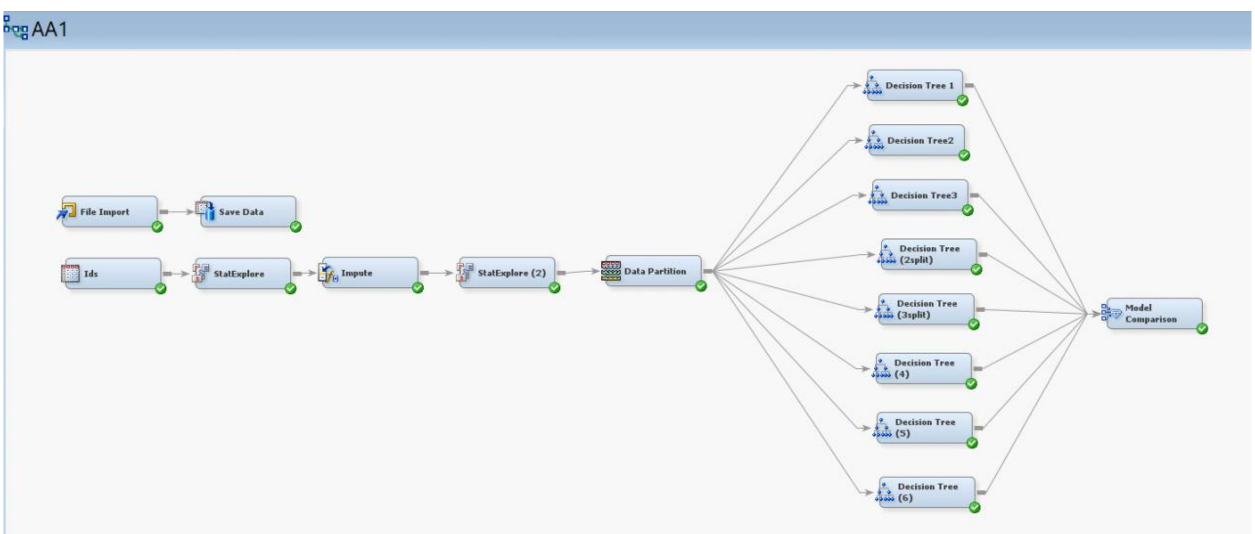


Add a "Split" node to the decision tree to partition the dataset into different subsets based on conditions of the input variables. This aids the model in learning patterns and trends within the data.

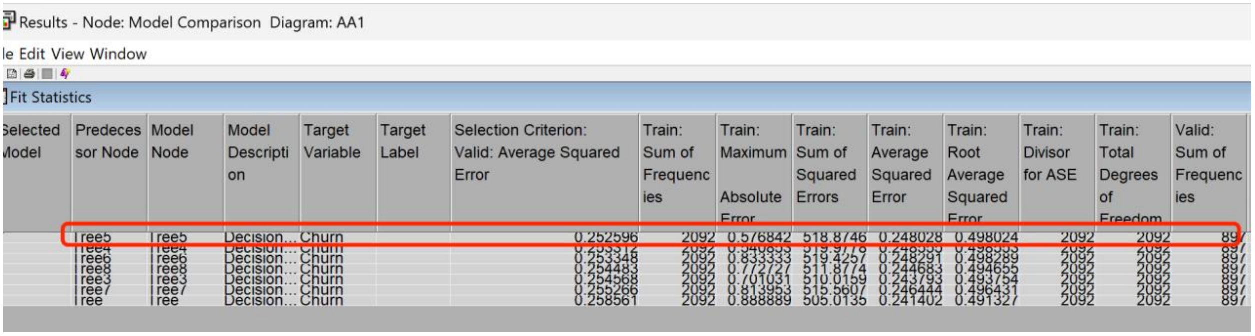


2.3 Comparison of multiple Decision tree

Add multiple decision tree models with different parameter values, then include a model comparison node to identify the decision tree model with the best predictive performance.



The results are as follows: The predictive performance of tree5 is the best.

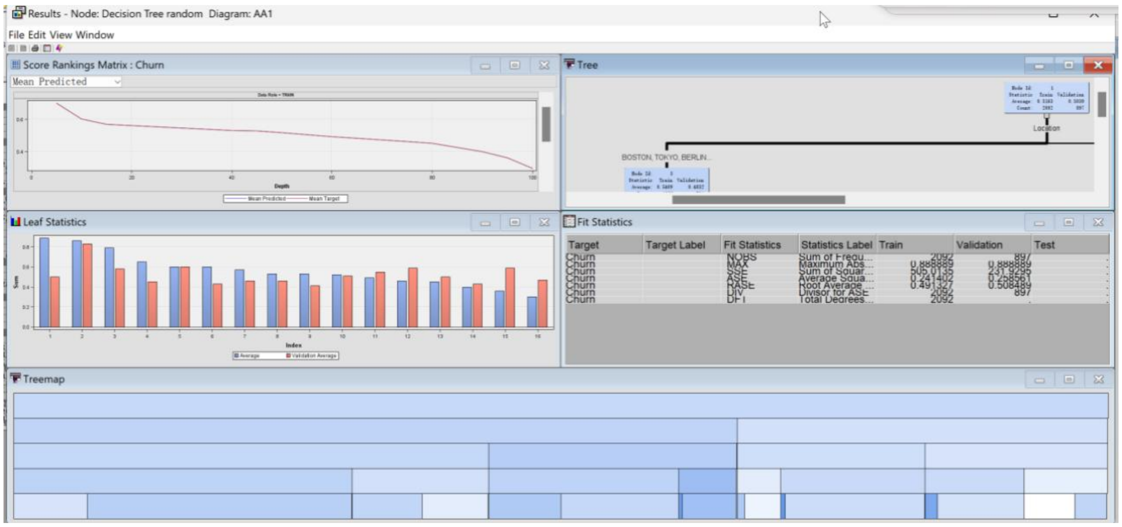


Results - Node: Model Comparison Diagram: AA1

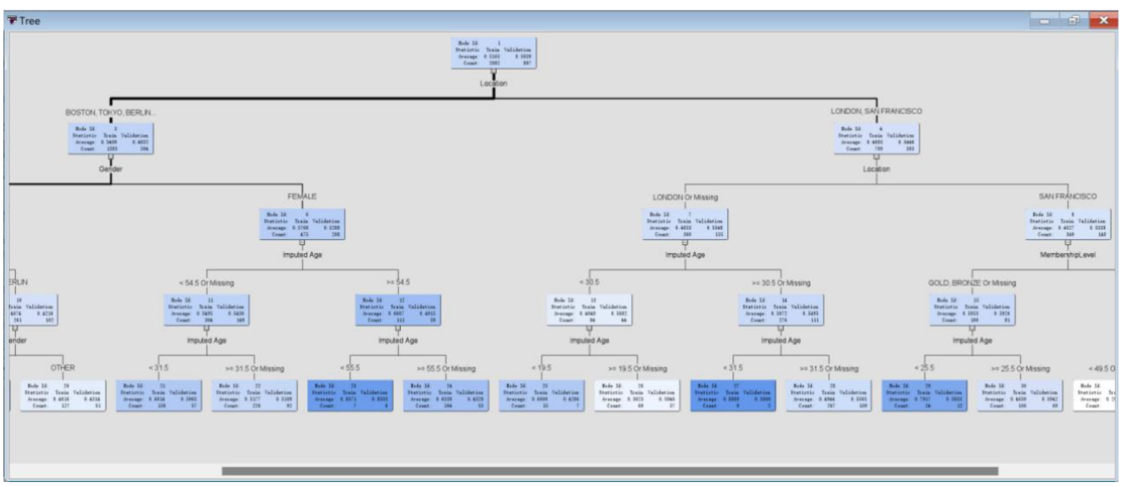
le Edit View Window

Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Average Squared Error	Train: Sum of Frequencies	Train: Maximum Absolute Error	Train: Sum of Squared Errors	Train: Average Squared Error	Train: Root Average Squared Error	Train: Divisor for ASE	Train: Total Degrees of Freedom	Valid: Sum of Frequencies
	Tree5	Tree5	Decision...	Churn		0.252596	2092	0.576842	518.8746	0.248028	0.498024	2092	2092	897
	Tree4	Tree4	Decision...	Churn		0.253312	2092	0.540833	519.9778	0.248555	0.498553	2092	2092	897
	Tree6	Tree6	Decision...	Churn		0.253348	2092	0.833333	519.4257	0.248291	0.498289	2092	2092	897
	Tree8	Tree8	Decision...	Churn		0.254483	2092	0.772727	511.8774	0.244683	0.494655	2092	2092	897
	Tree3	Tree3	Decision...	Churn		0.254568	2092	0.701031	510.0159	0.243793	0.493754	2092	2092	897
	Tree7	Tree7	Decision...	Churn		0.255266	2092	0.813953	515.5607	0.246444	0.496431	2092	2092	897
	Tree	Tree	Decision...	Churn		0.258561	2092	0.888889	505.0135	0.241402	0.491327	2092	2092	897



The decision tree with the best predictive performance is as follows:



The results for the decision tree with the best predictive performance are as follows, with an accuracy of 0.88889.

Data Role=VALIDATE Target Variable=Churn Target Label=' '

Range for Predicted	Mean Target	Mean Predicted	Number of Observations	Model Score
0.859 - 0.889	0.50000	0.88889	2	0.87414
0.830 - 0.859	0.83333	0.85714	6	0.84464
0.771 - 0.800	0.58333	0.79167	12	0.78565
0.653 - 0.682	0.45283	0.65385	53	0.66767
0.594 - 0.623	0.57813	0.60129	64	0.60868
0.564 - 0.594	0.46460	0.56917	226	0.57918
0.505 - 0.535	0.47305	0.52350	167	0.52019
0.476 - 0.505	0.55046	0.49438	109	0.49069
0.446 - 0.476	0.55118	0.45931	127	0.46120
0.387 - 0.417	0.43137	0.40157	51	0.40221
0.358 - 0.387	0.59459	0.36232	37	0.37271
0.299 - 0.328	0.46512	0.29897	43	0.31372

2.4 Analyse customer behaviour

Top-Level Node: This node displays that the entire dataset is initially split based on the "Location" variable, suggesting that "Location" might be a crucial predictive factor influencing the target variable. The top-level node divides the data into two or more subgroups, such as "BOSTON_TWO_OR_MORE" and "LYON."

Second-Level Nodes: These nodes further break down the data for each location based on gender ("FEMALE" or "MALE"). For instance, it can be observed that for the "BOSTON_TWO_OR_MORE" location, gender is a factor further dividing the data.

Third-Level Nodes and Below: Building upon gender, further segmentation is based on age, represented by the "impulse Age" variable. For example, age categories like "<=45" and ">55" may correspond to different user behavior patterns. This indicates that age is an influencing factor within specific gender and location combinations.

Leaf Nodes: These are the final nodes of the decision tree, representing the model's predictive outcomes. In the screenshot, each leaf node has an assessment of "Risk" and "Value," which could be probabilities or expected values derived from the model's learning on the training data.

From the analysis above, we can draw some preliminary conclusions about customer behavior:

Location Disparities: Location is a significant differentiating factor, suggesting potential significant differences in customer behavior across different regions.

Gender and Behavior: Gender is associated with certain customer behaviors, potentially impacting their purchasing decisions or service preferences.

Impact of Age: Age further refines differences in customer behavior, indicating that customers in different age groups may have distinct needs and preferences.

To derive meaningful business insights from these conclusions, consider the following action steps:

Customized Marketing: Design tailored marketing campaigns for customers based on different location and gender combinations.

Service Improvements: Adjust products or services to meet the diverse needs of customers in different age groups.

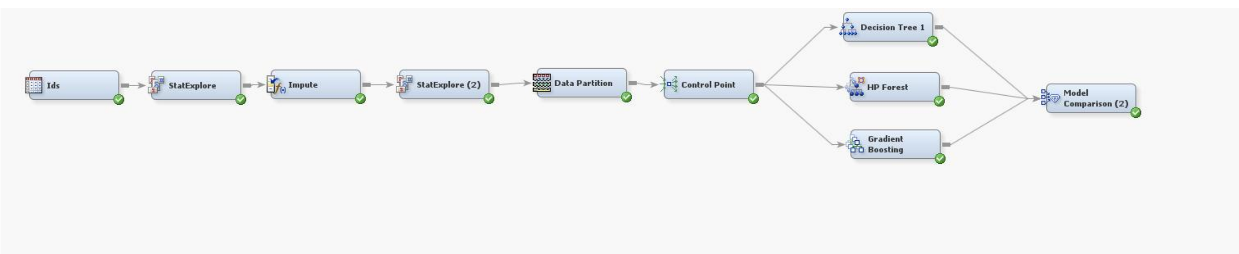
Risk Management: Identify high-risk customer groups and devise specific retention strategies for them.

Finally, applying integrated approaches such as random forests or gradient boosting can further enhance the model's performance and robustness, assisting businesses in making more accurate predictions and decisions based on complex datasets.

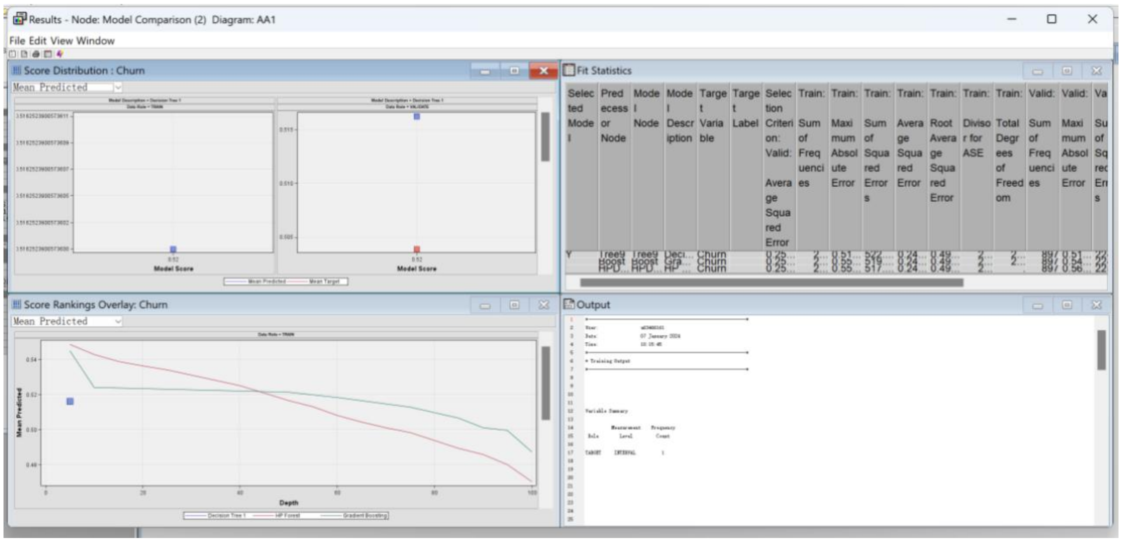
3 Ensemble Methods

3.1 Add random forest & gradient boosting model

Additionally, incorporate Random Forest as a bagging technique and Gradient Boosting as a boosting method. Subsequently, compare their performance with that of the decision tree, which demonstrates the best predictive capabilities.



3.2 Comparison



From the results, it appears that the decision tree outperforms others in terms of performance.

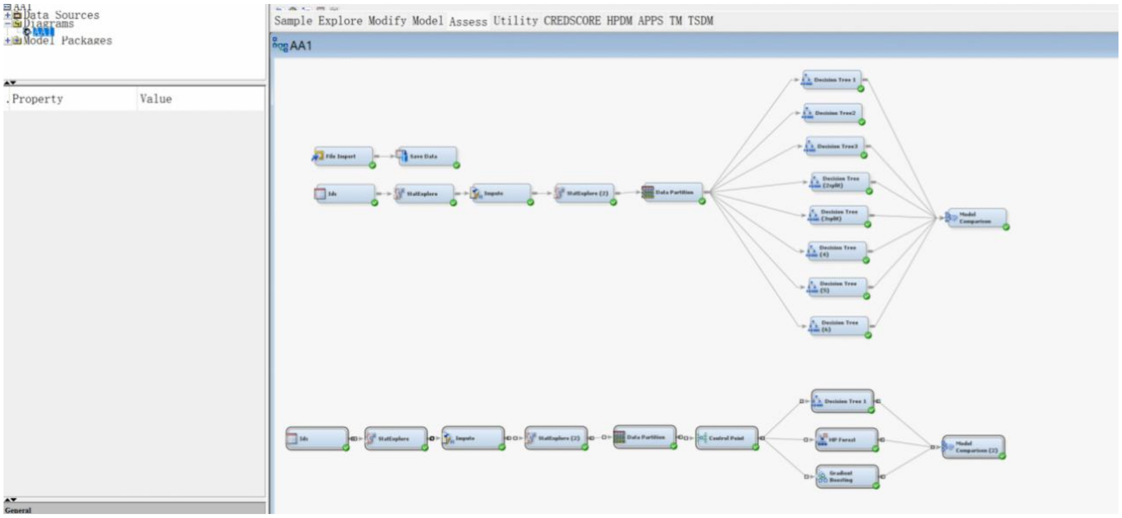
Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Average Squared Error	Train: Sum of Frequencies	Train: Maximum Absolute Error	Train: Sum of Squared Errors	Train: Average Squared Error	Train: Root Average Squared Error	Train: Divisor for ASE	Train: Total Degrees of Freedom	Valid: Sum of Frequencies	Valid: Maximum Absolute Error	Valid: Sum of Squared Errors	Valid: Average Squared Error	Valid: Root Average Squared Error	Valid: Divisor for VASE	Train: Sum of Case Weights Times Freq	Valid: Sum of Case Weights Times Freq
Y	Tree9	Tree9	Decisio...	Churn		0.250137	2092	0.516252	522.4474	0.249736	0.499736	2092	2092	897	0.516252	224.3732	0.250137	0.500137	897		
	Boost	Boost	Gradie...	Churn		0.250422	2092	0.552709	519.527	0.24834	0.498337	2092	2092	897	0.545244	224.6281	0.250422	0.500421	897	2092	897
	HPDM...	HPDM...	HP For...	Churn		0.250918	2092	0.553628	517.8115	0.24752	0.497514	2092	.	897	0.561657	225.0733	0.250918	0.500917	897	.	.

Fit Statistics
Model Selection based on Valid: Average Squared Error (_VASE_)

Selected Model	Model Node	Model Description	Valid: Average Squared Error	Train: Average Squared Error
Y	Tree9	Decision Tree 1	0.25014	0.24974
	Boost	Gradient Boosting	0.25042	0.24834
	HPDMForest	HP Forest	0.25092	0.24752

The overall status of the project is as follows:


AA1
Data Sources
Diagrams
Model Packages
Property
Value
General
Sample Explore Modify Model Assess Utility CREDSCORE HPDM APPS TM TSDM
AA1
File Import
Save Data
StatExplore
Impute
StatExplore (2)
Data Partition
Decision Tree 1
Model Comparison
Control Point
HP Forest
Gradient Boosting
Model Comparison (2)