

## **WQD7005 Data Mining**

### **Alternative Assessment 1 (G2)\_SAS\_Step**

<b>Matric ID</b>	<b>Name</b>
22060214	Mei Zhu
Git Hub link	<a href="https://github.com/ChristineLzy/WQD7005_AA1">https://github.com/ChristineLzy/WQD7005_AA1</a>

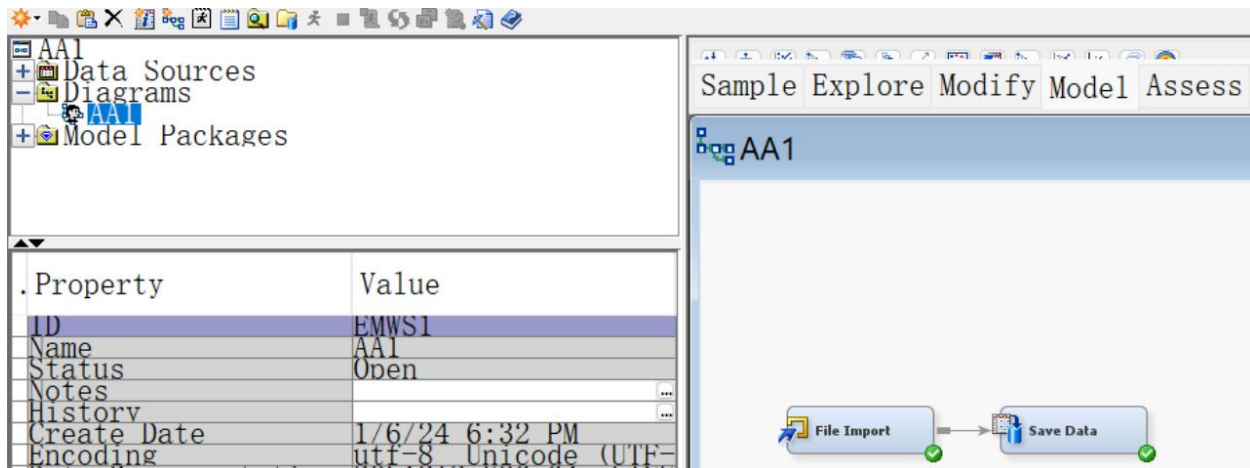
## Table of Contents

1 Data Import and Preprocessing .....	3
1.1 Import dataset into SAS .....	3
1.2 Setting specify variable roles .....	4
1.3 Handle missing values .....	5
2 Decision Tree Analysis .....	7
2.1 Data Partition .....	7
2.2 Decision Tree Model .....	8
2.3 Comparison of multiple Decision tree .....	9
2.4 Analyse customer behaviour .....	11
3 Ensemble Methods .....	12
3.1 Add random forest & gradient boosting model .....	12
3.2 Comparison .....	13

## 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.

**Data Source Wizard -- Step 5 of 8 Column Metadata**

(none) ☐ not Equal to ☐ ...

Columns: ☐ Label ☐ Mining ☐ Basic ☐ Statistics

Name	Role	Level	Report	Order	Drop	Lower Limit
Age	Input	Interval	No		No	.
Churn	Target	Interval	No		No	.
Customer ID	Input	Interval	No		No	.
Favorite	Input	Nominal	No		No	.
Gender	Input	Nominal	No		No	.
LastPurc	Input	Interval	No		No	.
Location	Input	Nominal	No		No	.
Membersh	Input	Nominal	No		No	.
Occupati	Input	Nominal	No		No	.
TotalPur	Input	Interval	No		No	.
TotalSpe	Input	Interval	No		No	.

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

**Data Source Wizard -- Step 8 of 8 Summary**

Metadata Completed.

Library: SEMMA  
Data Source: EM\_SAVE\_TRAIN  
Role: Raw

Role	Level	Count
ID	Interval	1
Input	Interval	4
Input	Nominal	5
target	Interval	1

### 1.3 Handle missing values

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

The screenshot shows the Orange3 interface. On the left, the 'StatExplore' node's properties are displayed in a table. The 'Interval Variables' property is highlighted with a red box. On the right, a workflow diagram shows the 'File Import' node connected to 'Save Data', and the 'Ids' node connected to 'StatExplore'.

Property	Value
<b>General</b>	
Node ID	Stat
Imported Data	...
Exported Data	...
Notes	...
<b>Train</b>	
Variables	...
<b>Data</b>	
Number of Observations	100000
Validation	No
Test	No
<b>Standard Reports</b>	
Interval Distributions	Yes
Class Distributions	Yes
Level Summary	Yes
Use Segment Variable	No
Cross-tabulation	...
<b>Variable Selection</b>	
Hide Rejected Variables	Yes
Number of Selected Variables	1000
Chi-Square Statistic	...
Chi-Square	Yes
Interval Variables	Yes
Number of Bins	...
<b>Correlation Statistics</b>	
Correlations	Yes
Pearson Correlations	Yes
Spearman Correlation	No
<b>Status</b>	
Create Time	1/7/24 1:34 AM
Run ID	...
Last Error	...
Last Status	...
Last Run Time	...
Run Duration	...
Grid Host	...
User-Added Node	No

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

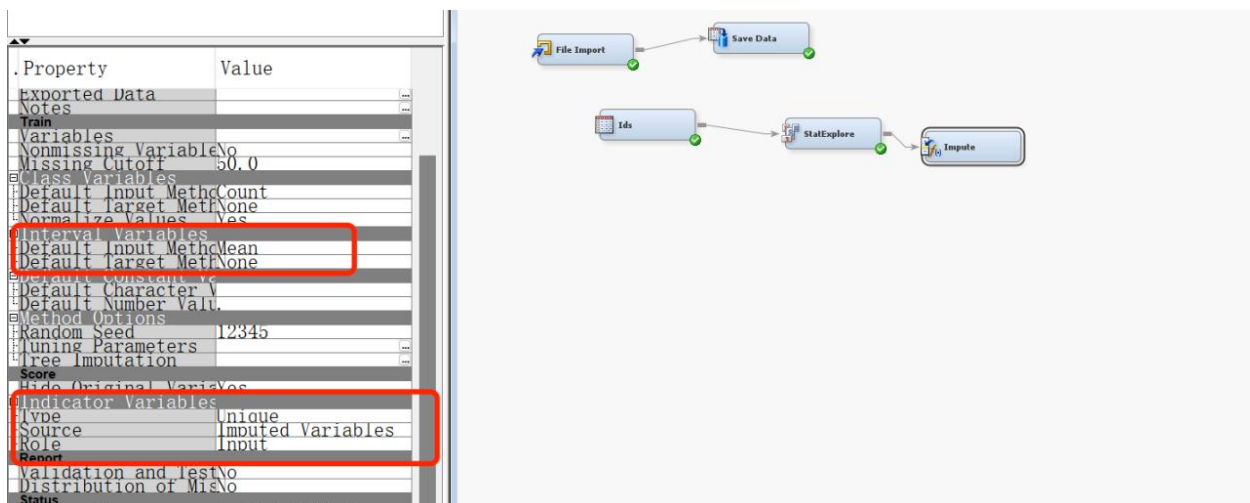
Output										
49										
50										
51	Interval Variable Summary Statistics									
52	(maximum 500 observations printed)									
53										
54	Data Role=TRAIN									
55										
56	Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness
57										Kurtosis
58										
59	Age	INPUT	43.38108	14.76881	2981	8	18	43	69	0.007942
60	LastPurchaseDate	INPUT	23193.5	106.4118	2989	0	23011	23194	23375	-0.00015
61	TotalPurchases	INPUT	50.10168	28.13889	2980	9	1	51	99	-0.01793
62	TotalSpent	INPUT	1182.889	1086.448	2989	0	50	823	4990	1.489987
63	Churn	TARGET	0.512546	0.499926	2989	0	0	1	1	-0.05023
64										
65										
66										

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:

Results - Node: Impute Diagram: AA1

File Edit View Window

Imputation Summary

Variable Name	Impute Method	Imputed Variable	Indicator Variable	Impute Value	Role	Measurement Level	Label	Number of Missing for TRAIN
Age	MEAN	IMP_Age	M_Age	43.38108	INPUT	INTERVAL		0
TotalPurchases	MEAN	IMP_TotalPurchases	M_TotalPurchases	50.10168	INPUT	INTERVAL		0

Output

Variable	Mean	Std. Dev.	N	Min	Max	Q1	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	Q23	Q24	Q25	Q26	Q27	Q28	Q29	Q30	Q31	Q32	Q33	Q34	Q35	Q36	Q37	Q38	Q39	Q40	Q41	Q42	Q43	Q44	Q45	Q46	Q47	Q48	Q49	Q50	Q51	Q52	Q53	Q54	Q55	Q56	Q57	Q58	Q59	Q60	Q61	Q62	Q63	Q64	Q65	Q66	Q67	Q68	Q69	Q70	Q71	Q72	Q73	Q74	Q75	Q76	Q77	Q78	Q79	Q80	Q81	Q82	Q83	Q84	Q85	Q86	Q87	Q88	Q89	Q90	Q91	Q92	Q93	Q94	Q95	Q96	Q97	Q98	Q99	Q100
Age	43.38108	10.10168	1000	18	65	25	35	45	55	65	75	85	95	105	115	125	135	145	155	165	175	185	195	205	215	225	235	245	255	265	275	285	295	305	315	325	335	345	355	365	375	385	395	405	415	425	435	445	455	465	475	485	495	505	515	525	535	545	555	565	575	585	595	605	615	625	635	645	655	665	675	685	695	705	715	725	735	745	755	765	775	785	795	805	815	825	835	845	855	865	875	885	895	905	915	925	935	945	955	965	975	985	995	1000

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

53	Interval Variable Summary Statistics										
54	(maximum 500 observations printed)										
55											
56	Data Role=TRAIN										
57											
58				Standard	Non						
59	Variable	Role	Mean	Deviation	Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
60											
61											
62	IMP_Age	INPUT	43.38108	14.74903	2989	0	18	43	69	0.007953	-1.1351
63	IMP_TotalPurchases	INPUT	50.10168	28.09648	2989	0	1	50.10168	99	-0.01795	-1.16111
64	LastPurchaseDate	INPUT	23193.5	106.4118	2989	0	23011	23194	23375	-0.00015	-1.23545
65	TotalSpent	INPUT	1182.889	1086.448	2989	0	50	823	4990	1.489987	1.763251
66	Churn	TARGET	0.512546	0.499926	2989	0	0	1	1	-0.05023	-1.99882
67											
68											

## 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%.

Property	Value
<b>General</b>	
Node ID	Part
Imported Data	...
Exported Data	...
Notes	...
<b>Train</b>	
Variables	...
Output Type	Data
Partitioning Method	Default
Random Seed	12345
<b>Data Set Allocations</b>	
Training	70.0
Validation	30.0
Test	0.0
<b>Report</b>	
Interval targets	Yes
Class targets	Yes
<b>Status</b>	
Create time	1/7/24 2:18 AM
Run ID	
Last Error	
Last Status	
Last Run time	
Run Duration	
Grid Host	
User-Added Node	No

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		
Type	Data Set	Number of 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.

Property	Value
Missing Values	Use in search
Use Input Once	No
Maximum Branch	5
Maximum Depth	6
Minimum Categoricals	5
Leaf Size	5
Number of Rules	5
Number of Surrogate	0
Split Size	1
Use Decisions	No
Use Priors	No
Exhaustive	3000
Node Sample	20000
Method	Assessment
Number of Leaves	1
Assessment Measure	Decision
Assessment Fraction	0.25
Perform Cross Valid	No
Number of Subsets	10
Number of Repeats	1
Seed	12345
Observation Based	No
Number Single Var	5
Bonferroni Adjustment	Yes
Time of Bonferroni	Before
Inputs	No
Number of Inputs	1
Depth Adjustment	Yes
Output Variables	Yes
Leaf Variable	Yes
Create Sample	Default
Sample Method	Random
Sample Size	10000
Sample Seed	12345
Performance	Disk
Variable Selection	Yes
Leaf Role	Segment
Precision	4
Tree Precision	4
Class Target Node (Percent Correctly C	
Interval Target Node (Average	
Node Text	
Create time	1/7/24 9:30 AM
Run ID	6926e551-d44a-3b47
Last Error	
Last Status	Complete
Last Run time	1/7/24 9:32 AM
Run Duration	0 Hr. 0 Min. 3.84 s
Grid Host	
User-Added Node	No

```

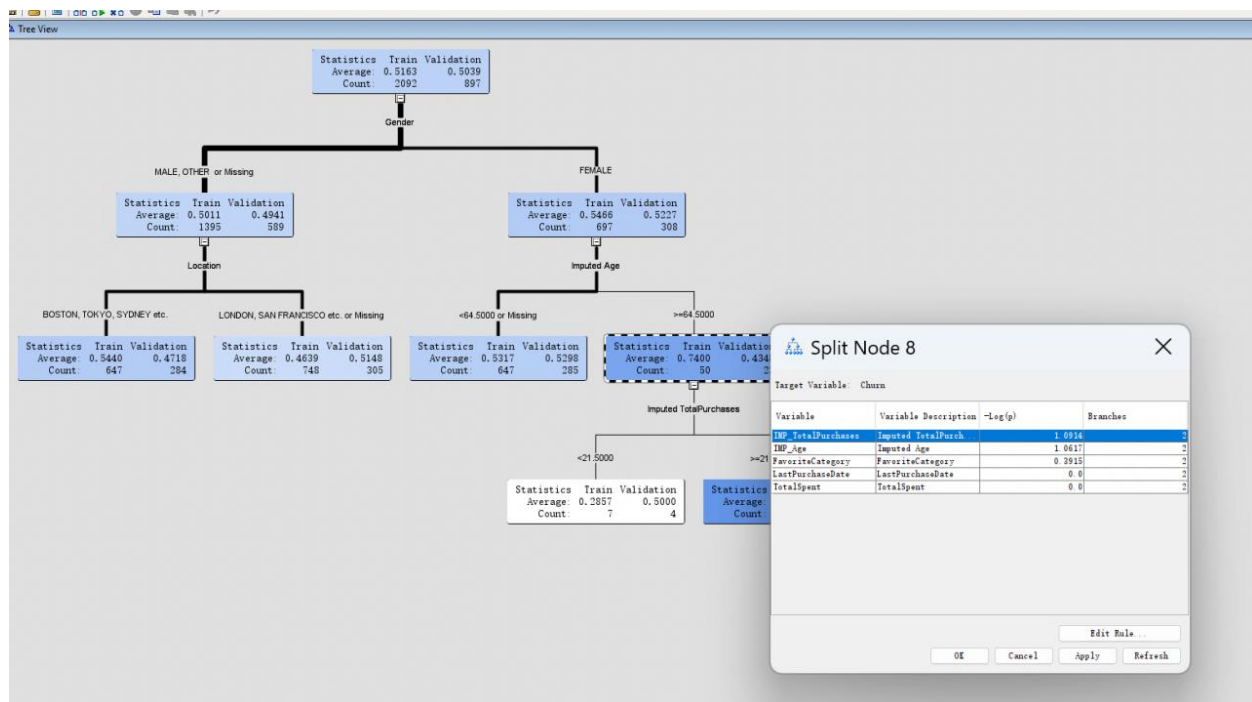
graph LR
    FileImport[File Import] --> SaveData[Save Data]
    Data[Data] --> StatExplore[StatExplore]
    SaveData --> Input[Input]
    StatExplore --> StatExplore2[StatExplore (2)]
    Input --> DataPartition[Data Partition]
    StatExplore2 --> DecisionTree[Decision Tree]
  
```

Run completed

© 22060214@siswa.um.edu.my as u63466161 Connected to SASApp - Logical Workspace Server

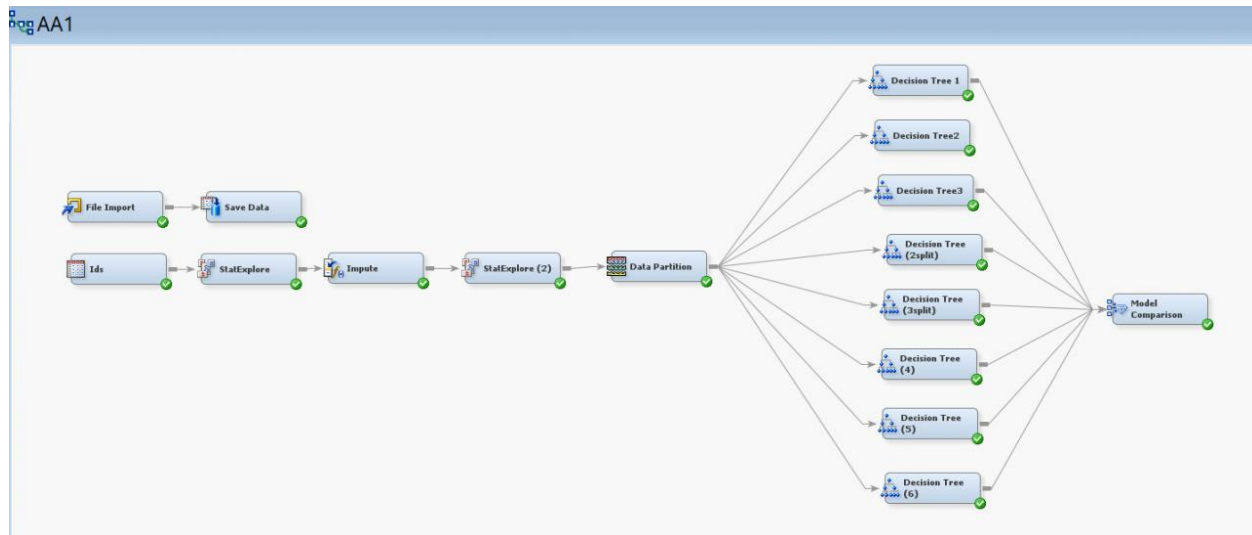
Add a "Split" node to the decision tree to partition the dataset into different subsets based on conditions of the input variables. This aids the model in learning patterns and trends within the data.





## 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

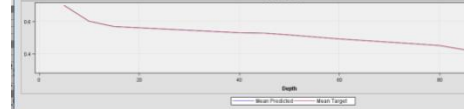
Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Average Squared Error	Train: Sum of Frequencies	Train: Maximum Absolute Error	Train: Sum of Squared Errors	Train: Average Squared Error	Train: Root Average Squared Error	Train: Divisor for ASE	Train: Total Degrees of Freedom	Valid: Sum of Frequencies
tree5	tree5	tree5	Decision... Churn			0.252596	2092	0.576842	518.8746	0.248028	0.498024	2092	2092	89
tree4	tree4	tree4	Decision... Churn			0.253348	2092	0.833933	518.8779	0.248299	0.498289	2092	2092	89
tree6	tree6	tree6	Decision... Churn			0.254483	2092	0.772727	511.8774	0.244683	0.494655	2092	2092	89
tree3	tree3	tree3	Decision... Churn			0.254988	2092	0.701031	510.0159	0.243793	0.493754	2092	2092	89
tree7	tree7	tree7	Decision... Churn			0.255256	2092	0.813953	519.0617	0.246444	0.496431	2092	2092	89
tree	tree	tree	Decision... Churn			0.258561	2092	0.888889	505.0135	0.241402	0.491327	2092	2092	89

## Results - Node: Decision Tree random Diagram: AA1

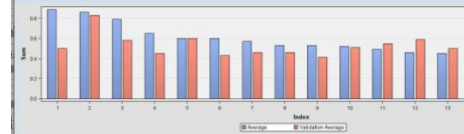
File Edit View Window

Score Rankings Matrix: Churn

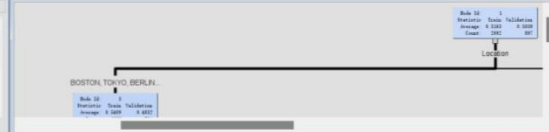
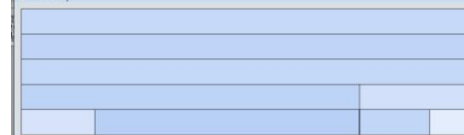
Mean Predicted



Leaf Statistics

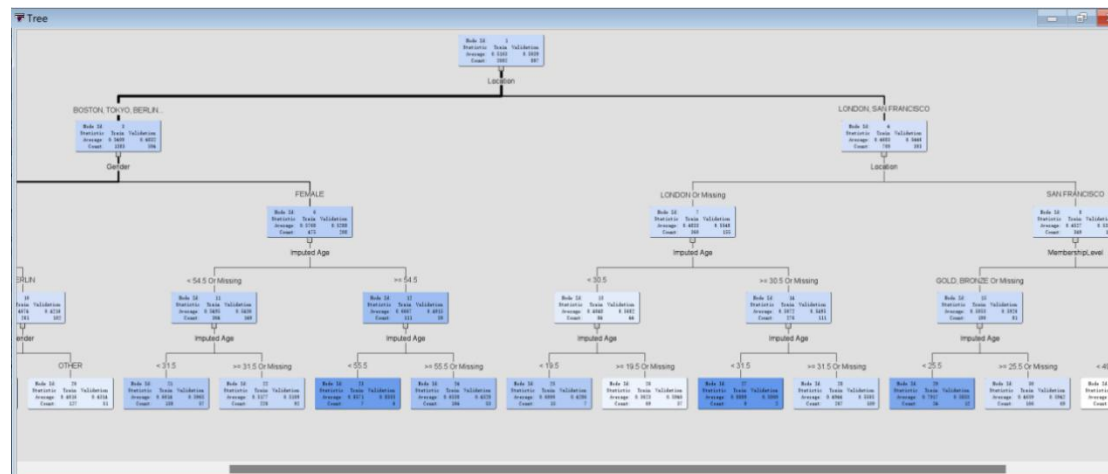


Tree



Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
Churn	Churn	N/A	Sum of Frequencies	2092	89	89
Churn	Churn	MAX	Sum of Squared Errors	0.888889	0.888889	0.888889
Churn	Churn	AVERAGE	Average Squared Error	0.252596	0.252596	0.252596
Churn	Churn	ROOT AVERAGE	Root Average Squared Error	0.491327	0.491327	0.491327
Churn	Churn	DI	Divisor for ASE	2092	89	89
Churn	Churn		Total Degrees of Freedom	2092	89	89

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.

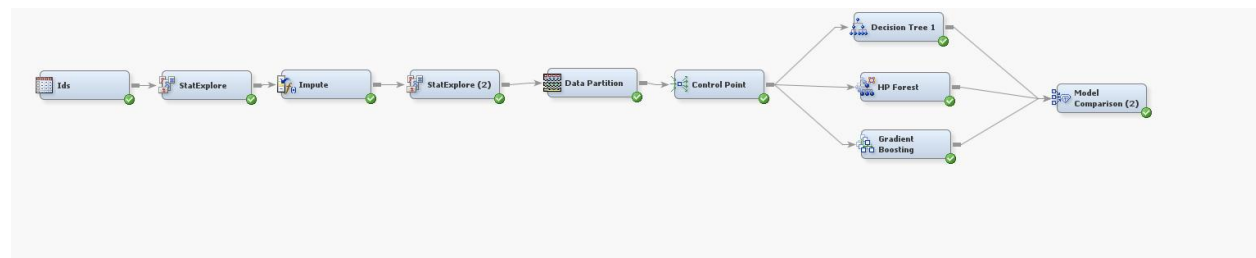
**Risk Management:** Identify high-risk customer groups and devise specific retention strategies for them.

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.

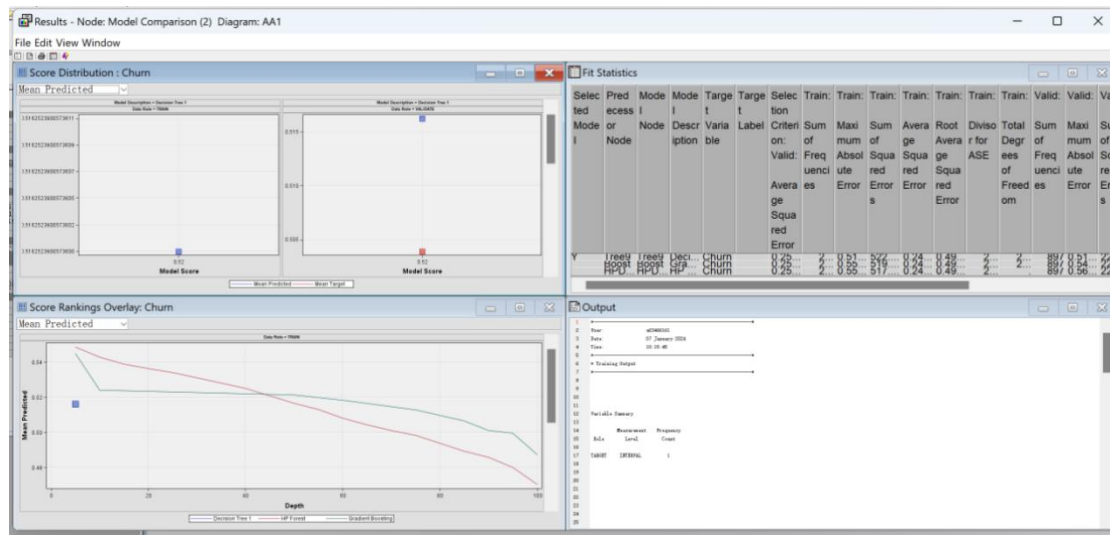
### 3 Ensemble Methods

#### 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																						
Select d Model	Precede sor Node	Model Node	Model Descript ion	Target Variable	Target Label	Selection Criterion: Valid: Average Squared Error		Train: Sum of Fre quencies	Train: Maximum Absolute Error	Train: Sum of Square d Errors	Train: Average Square d Error	Train: Root Average Square d Error	Train: Divisor for ASE	Train: Total Degrees of Freedom	Valid: Sum of Fre quencies	Valid: Maximum Absolute Error	Valid: Sum of Square d Errors	Valid: Average Square d Error	Valid: Root Average Square d Error	Valid: Divisor for VASE	Train: Sum of Case Weights Times Freq	Valid: Sum of Case Weights Times Freq
Y	Tree9 Boost HPDM	Tree9 Boost HPDM	Decision Tree Gradient Boosting HP Forest	Churn	Churn			0.25013 0.25042 0.25092	2092 2092 2092	0.16525 0.16270 0.15326	523.4474 519.947 517.818	0.24973 0.24834 0.24752	2092 2092 2092	2092 2092 2092	891 891 891	0.15622 0.14922 0.14616	224.3732 224.6628 225.0733	0.25014 0.25042 0.25092	0.25014 0.25042 0.25092	891 891 891	2092	891

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

Log AA1

