

WQD7005 Data Mining

Alternative Assessment 1 (G2)

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

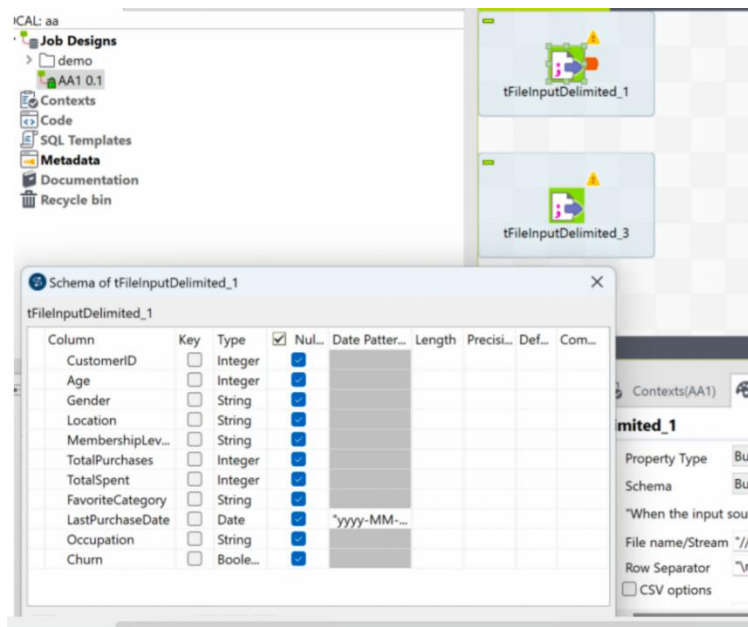
Table of Contents

1	Data Import and Preprocessing	3
1.1	Merge datasets using Talend Data Integration	3
1.2	Data processing using Talend Data Preparation	4
1.3	Import dataset into SAS	6
1.4	Setting specify variable roles	7
1.5	Handle missing values	8
2	Decision Tree Analysis	10
2.1	Data Partition	10
2.2	Decision Tree Model	11
2.3	Comparison of multiple Decision tree	12
2.4	Analyse customer behaviour	14
3	Ensemble Methods	15
3.1	Add random forest & gradient boosting model	15
3.2	Comparison	16

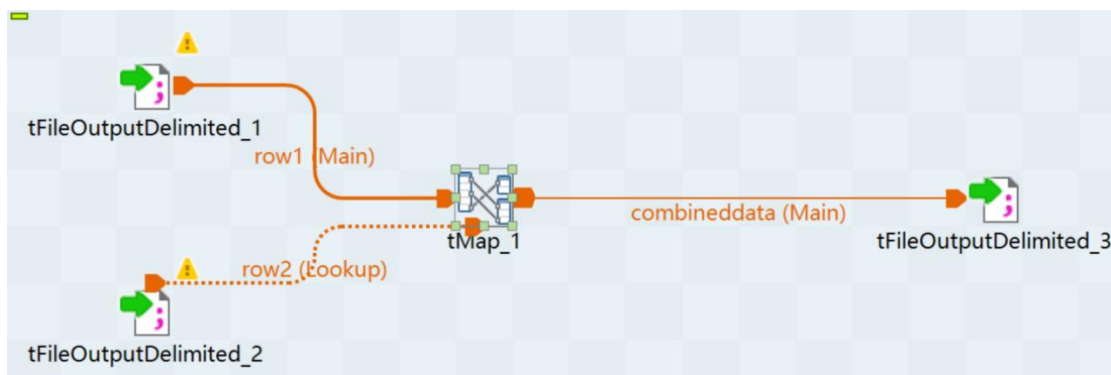
1 Data Import and Preprocessing

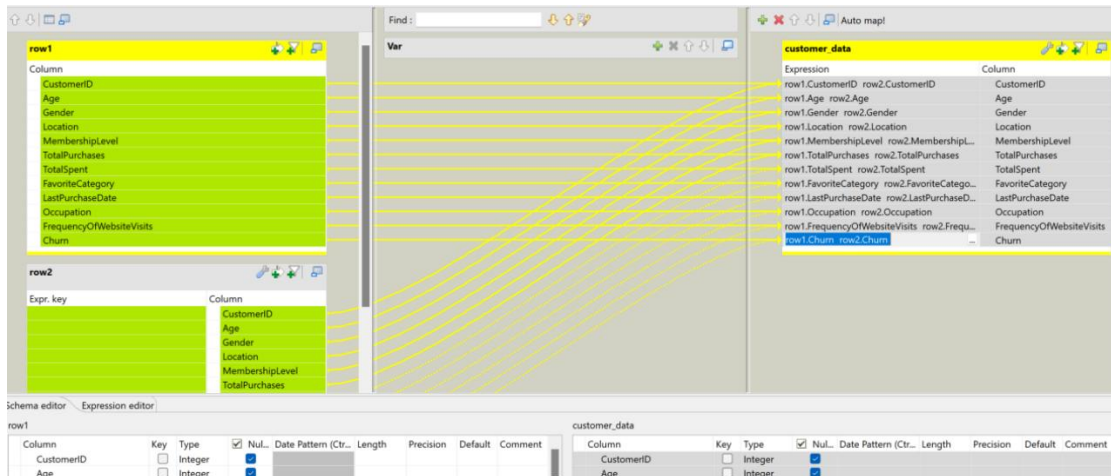
1.1 Merge datasets using Talend Data Integration

Import two CSV datasets in Talend Integration, and click on the "Edit schema" button for further customization.



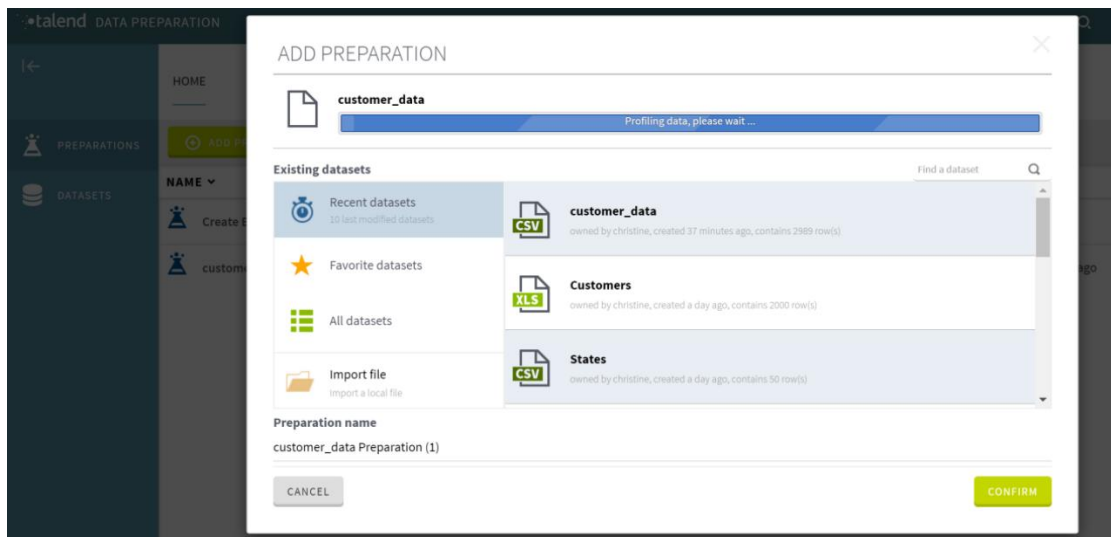
Add a tMap node to merge two datasets and generate the output.



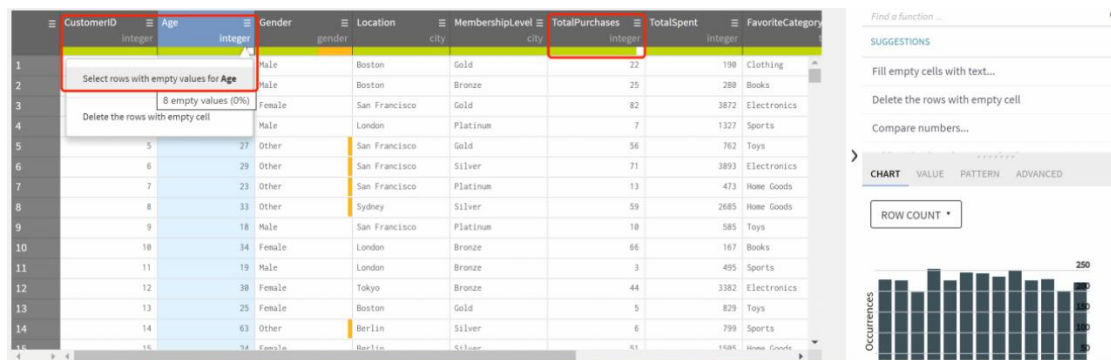


1.2 Data processing using Talend Data Preparation

Import the merged dataset in Talend Data Preparation.



Filter records with null values and replace them using the mean.



Filters Age: rows with empty values 8/2989

Processing...

CustomerID	Age	Gender	Location	MembershipLevel	TotalPurchases	TotalSpent	FavoriteCategory
integer	integer	gender	city	city	integer	integer	
2		Male	Boston	Bronze	25	288	Books
63		Female	San Francisco	Gold	85	1574	Electronics
279		Other	London	Platinum	15	938	Electronics
312		Other	San Francisco	Gold	65	1688	Home Goods
318		Female	Tokyo	Bronze	89	345	Toys
334		Other	San Francisco	Bronze	51	4185	Electronics
341		Male	San Francisco	Platinum	52	58	Toys
348		Female	Tokyo	Silver	28	757	Home Goods

Age

COLUMN ROW

Find a function ...

Value: 43 mean value

Apply changes to: ☐ All rows ☒ Filtered rows

CHART VALUE PATTERN ADVANCED

ROW COUNT *

The replacement of missing values with the mean was successful, and now there are no null values in this column.

Filters Age: rows with empty values 0/2989

Add a filter ...

CustomerID	Age	Gender	Location	MembershipLevel
integer	integer	gender	city	city

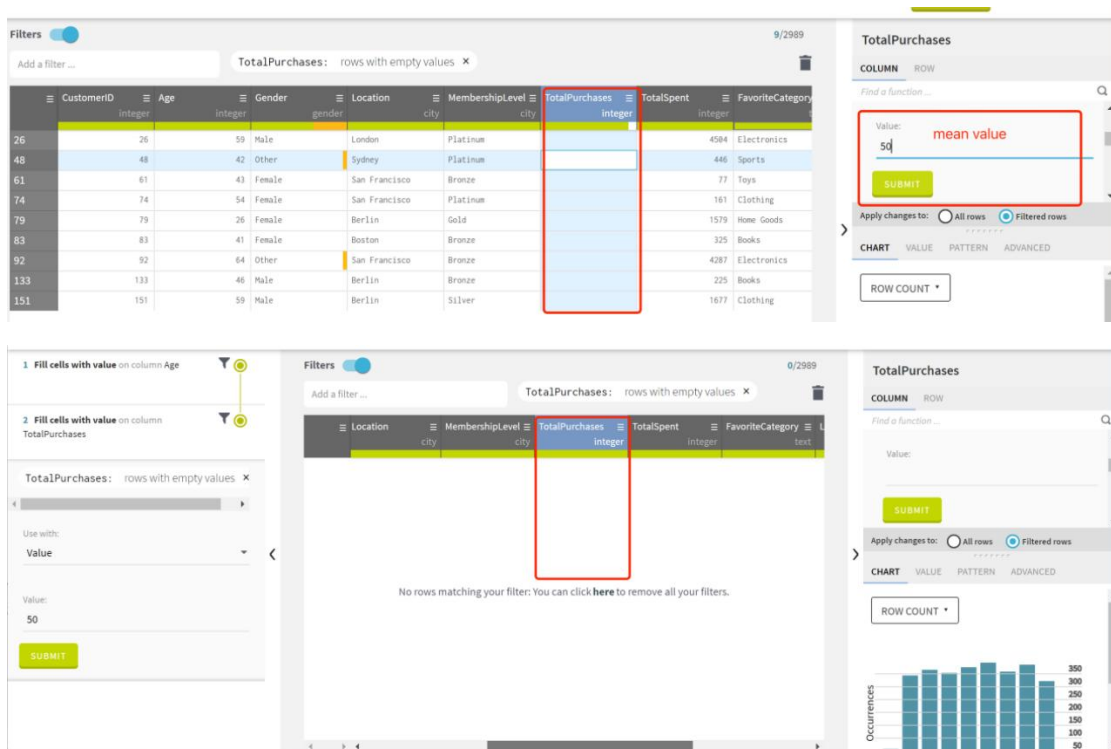
Apply the same method to handle the "totalpurchase" column.

Add a filter ...

CustomerID	Age	Gender	Location	MembershipLevel	TotalPurchases	TotalSpent	FavoriteCategory
integer	integer	gender	city	city	integer	integer	
1	1	55 Male	Boston			190	Clothing
2	2	43 Male	Boston			280	Books
3	3	30 Female	San Francisco			3872	Electronics
4	4	26 Male	London			1327	Sports
5	5	27 Other	San Francisco	Gold	56	762	Toys
6	6	29 Other	San Francisco	Silver	71	3893	Electronics
7	7	23 Other	San Francisco	Platinum	13	473	Home Goods
8	8	33 Other	Sydney	Silver	59	2685	Home Goods
9	9	18 Male	San Francisco	Platinum	10	585	Toys
10	10	34 Female	London	Bronze	66	167	Books
11	11	19 Male	London	Bronze	3	495	Sports
12	12	30 Female	Tokyo	Bronze	44	3382	Electronics
13	13	25 Female	Boston	Gold	5	829	Toys
14	14	63 Other	Berlin	Silver	6	799	Sports
15	15	24 Female	Berlin	Silver	61	1585	Home Goods

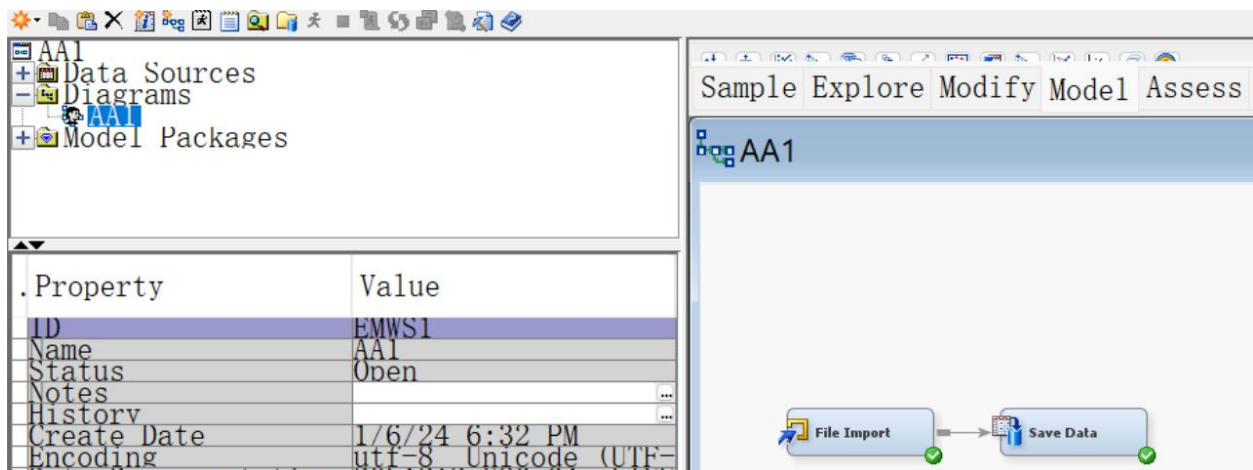
Select rows with empty values for TotalPurchases

Delete the rows with empty cell



1.3 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.

1.4 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

1.5 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 configuration panel is open, displaying various settings. The 'Interval Variables' checkbox is highlighted with a red rectangle. On the right, the workflow diagram shows a 'File Import' node connected to a 'Save Data' node, and an 'Ids' node connected to the 'StatExplore' node.

Property	Value
General	
Node ID	Stat
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Data	
Number of Observations	100000
Validation	No
Test	No
Standard Reports	
Interval Distribution	Yes
Class Distributions	Yes
Level Summary	Yes
Use Segment Variable	No
Cross-tabulation	...
Variable Selection	
Hide Rejected Variables	Yes
Number of Selected Variables	1000
Chi-Square Statistic	
Chi-Square	Yes
Interval Variables	Yes
Number of bins	5
Correlation Statistic	
Correlations	Yes
Pearson Correlations	Yes
Spearman Correlation	No
Status	
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.

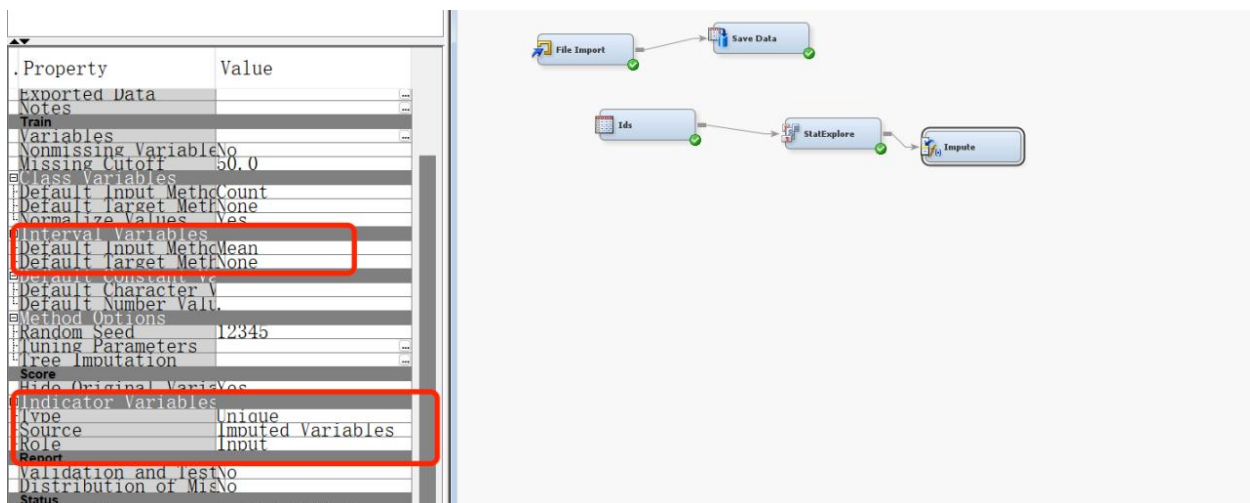
49	
50	
51	Interval Variable Summary Statistics
52	(maximum 500 observations printed)
53	
54	Data Role=TRAIN
55	
56	
57	Variable Role Mean Standard Non Missing Minimum Median Maximum Skewness Kurtosis
58	
59	Age INPUT 43.38108 14.76881 2981 8 18 43 69 0.007942 -1.1401
60	LastPurchaseDate INPUT 23193.5 106.4118 2989 0 23011 23194 23375 -0.00015 -1.23545
61	TotalPurchases INPUT 50.10168 28.13889 2980 9 1 51 99 -0.01793 -1.16665
62	TotalSpent INPUT 1182.889 1086.448 2989 0 50 823 4990 1.489987 1.763251
63	Churn TARGET 0.512546 0.499926 2989 0 0 1 1 -0.05023 -1.99882
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	Standard Deviation	Minimum	Maximum	Number of Missing
Age	43.38108	10.10168	18.0	65.0	0
TotalPurchases	50.10168	10.10168	10.0	100.0	0

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	Missing					
59	Variable	Role	Mean	Deviation	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
General	
Node ID	Part
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Output Type	Data
Partitioning Method	Default
Random Seed	12345
Data Set Allocations	
Training	70.0
Validation	30.0
Test	0.0
Report	
Interval targets	Yes
Class targets	Yes
Status	
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

```

graph LR
    A[File Import] --> B[Save Data]
    C[Ids] --> D[StatExplore]
    D --> E[Impute]
    E --> F[StatExplore (2)]
    F --> G[Data Partition]

```

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
Minimum 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
Subsample	
Method	Assessment
Number of Leaves	1
Assessment Measure	Decision
Assessment Fraction	0.25
Cross Validation	
Perform Cross Valid	No
Number of Subsets	10
Number of Repeats	1
Seed	12345
Observation Based	
Observation Based	No
Number Single Var	5
Multiple Adjustment	
Bonferroni Adjustment	Yes
Time of Bonferroni	Before
Inputs	No
Number of Inputs	1
Depth Adjustment	Yes
Output Variables	
Leaf Variable	Yes
Minimum Leaf Sample	
Create Sample	Default
Sample Method	Random
Sample Size	10000
Sample Seed	12345
Performance	Disk
Score	
Variable Selection	Yes
Leaf Role	Segment
Report	
Tree Precision	4
Class Target Node (Percent Correctly C	
Interval Target Node Coverage	
Node Text	
Status	
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

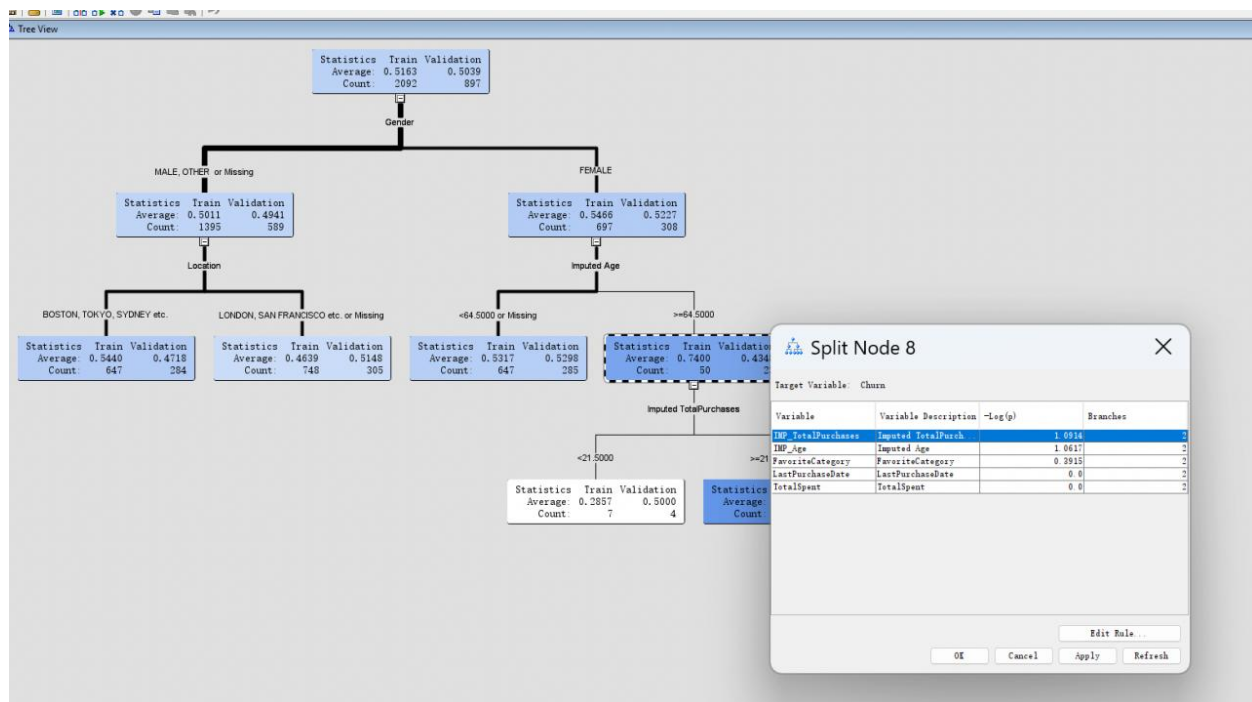
```

graph LR
    A[File Import] --> B[Save Data]
    B --> C[ID]
    C --> D[StatExplore]
    D --> E[Input]
    E --> F[StatExplore (2)]
    F --> G[Data Partition]
    G --> H[Decision Tree]
  
```

Run completed

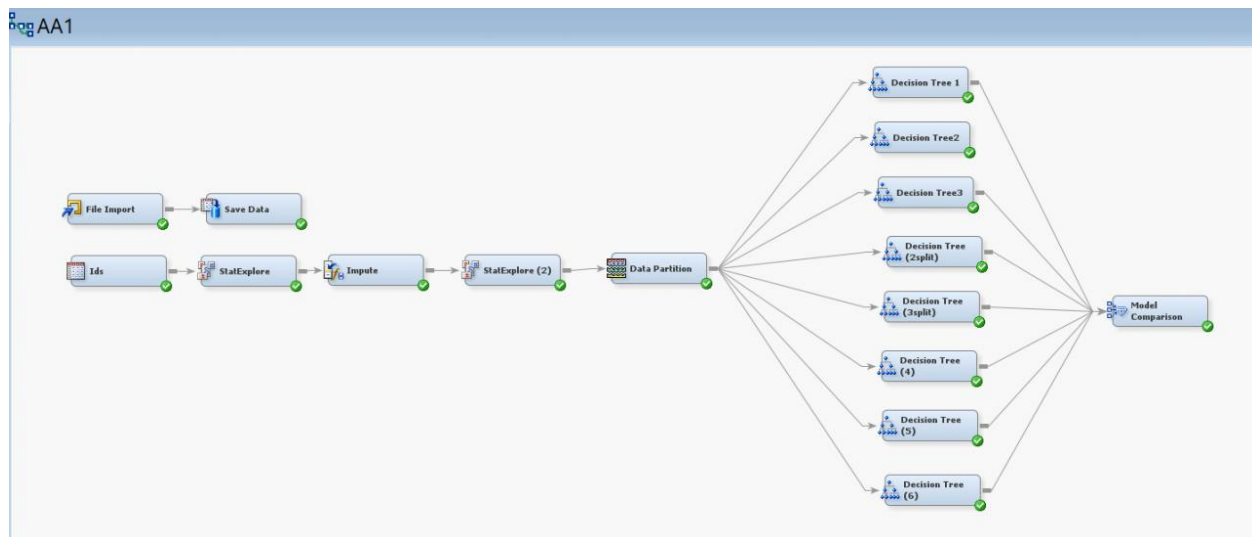
© 22060214@siswa.um.edu.my as u63466161 Connected to SASApp - Logical Workspace Server

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	Decision...	Churn			0.252596	2092	0.576842	518.8746	0.248028	0.498024	2092	2092	89
tree4	tree4	Decision...	Churn			0.253348	2092	0.833933	518.8779	0.248299	0.498299	2092	2092	89
tree6	tree6	Decision...	Churn			0.254483	2092	0.772727	511.8774	0.244683	0.494655	2092	2092	89
tree3	tree3	Decision...	Churn			0.254988	2092	0.701031	510.0159	0.243793	0.493754	2092	2092	89
tree7	tree7	Decision...	Churn			0.255256	2092	0.813953	515.0617	0.246444	0.496431	2092	2092	89
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

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

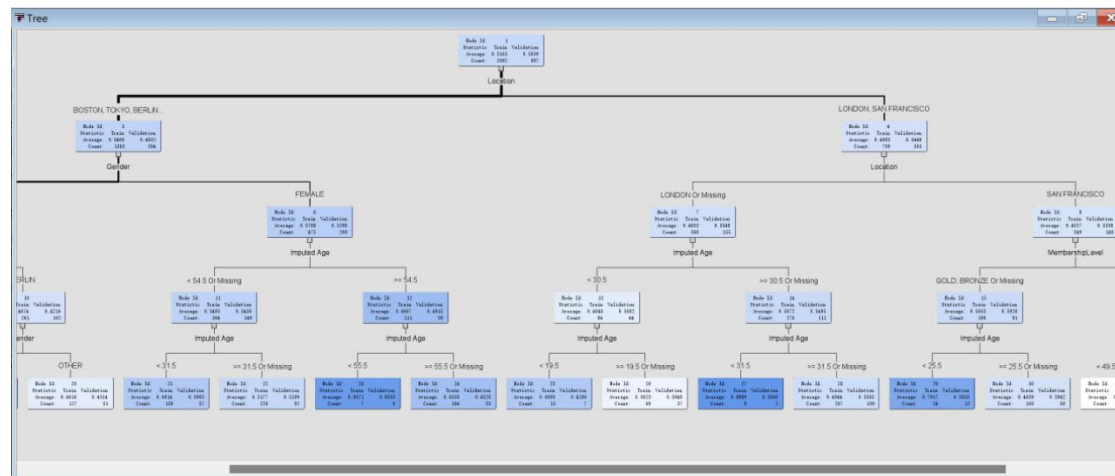
Mean Predicted

Mean Predicted

Mean Predicted

Mean Predicted

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.

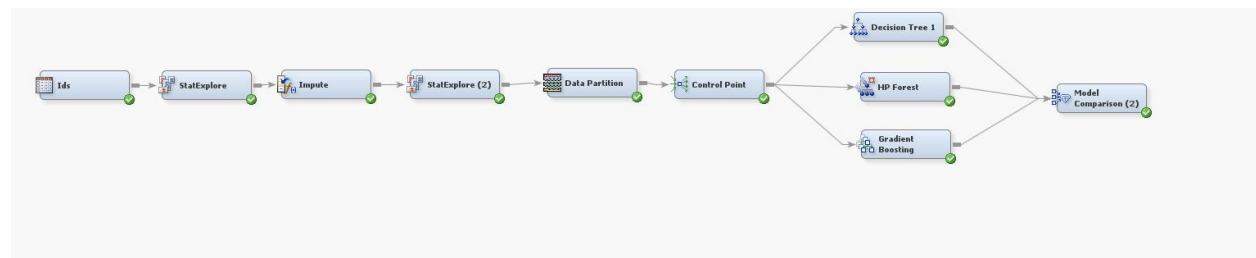
Risk Management: Identify high-risk customer groups and devise specific retention strategies for them.

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.

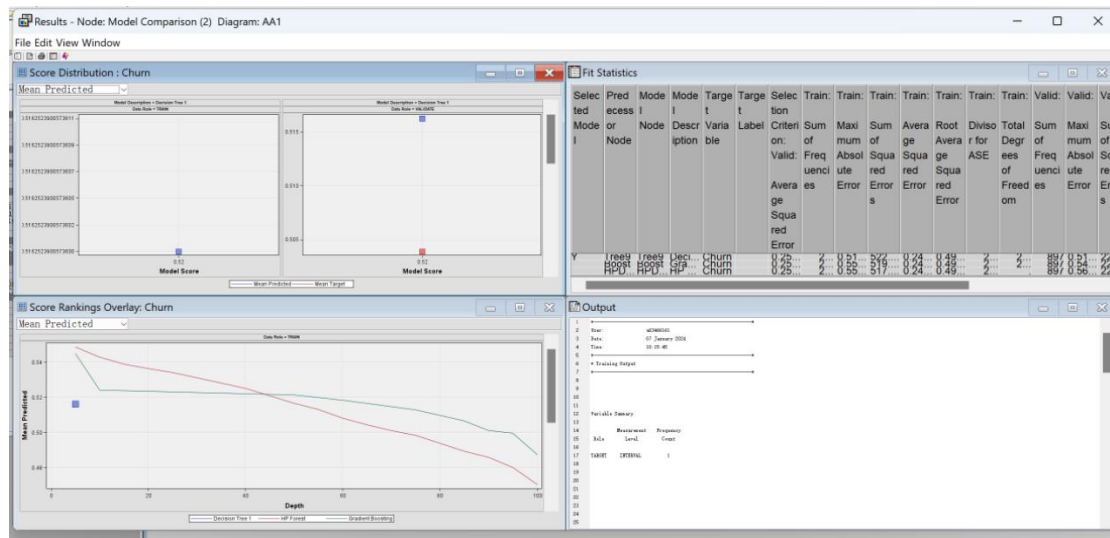
3 Ensemble Methods

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	Predictor 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	Valid: Sum of Case Weights
Y	Tree9	Tree9	Decision Tree 1	Churn	Churn	0.25014	2092	0.516252	522.44	0.249736	0.498437	2092	2092	897	0.516252	224.37	0.250137	0.500137	897	2092	897
	Boost	Boost	Gradient Boosting	Churn	Churn	0.25042	2092	0.52709	519.547	0.24834	0.498437	2092	2092	897	0.545234	224.8281	0.250427	0.500427	897	2092	897
	HPDM	HPDM	HP Forest	Churn	Churn	0.250918	2092	0.533628	517.8115	0.24752	0.497914	2092	2092	897	0.561697	225.0733	0.250918	0.500917	897	2092	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

Log AA1

