ALTERNATIVE ASSESSMENT 1

Shijie Zeng 23051016

GitHub: https://github.com/firesaku/WQD7005

Answer the question below based on the given scenario.

Submit your answer within ONE (1) DAY after the question is given in SPECTRUM. Answers should be submitted and saved with the student's name followed by matric number as the file name in the format of .pdf (e.g. Ali_s123456.pdf).

Case Study: E-Commerce Customer Behaviour Analysis

Background:

You will work with a dataset of customer transactions from an e-commerce website, encompassing various customer attributes and purchase history over the last year. The structure provided below is a guideline. Feel free to enhance this dataset by adding relevant attributes that you believe will enrich your analysis. Use the structure as a foundation to create your own sample dataset that reflects realistic customer behaviour.

Dataset Structure:

CustomerID: Unique identifier for each customer.

Age: Age of the customer.

Gender: Gender of the customer.

Location: Geographic location of the customer.

MembershipLevel: Indicates the membership level (e.g., Bronze, Silver, Gold, Platinum).

TotalPurchases: Total number of purchases made by the customer.

TotalSpent: Total amount spent by the customer.

FavoriteCategory: The category in which the customer most frequently shops (e.g., Electronics, Clothing, Home Goods).

LastPurchaseDate: The date of the last purchase.

[Additional Attributes]: Consider adding more attributes like customer's occupation, frequency of website visits, etc.

Churn: Indicates whether the customer has stopped purchasing (1 for churned, 0 for active).

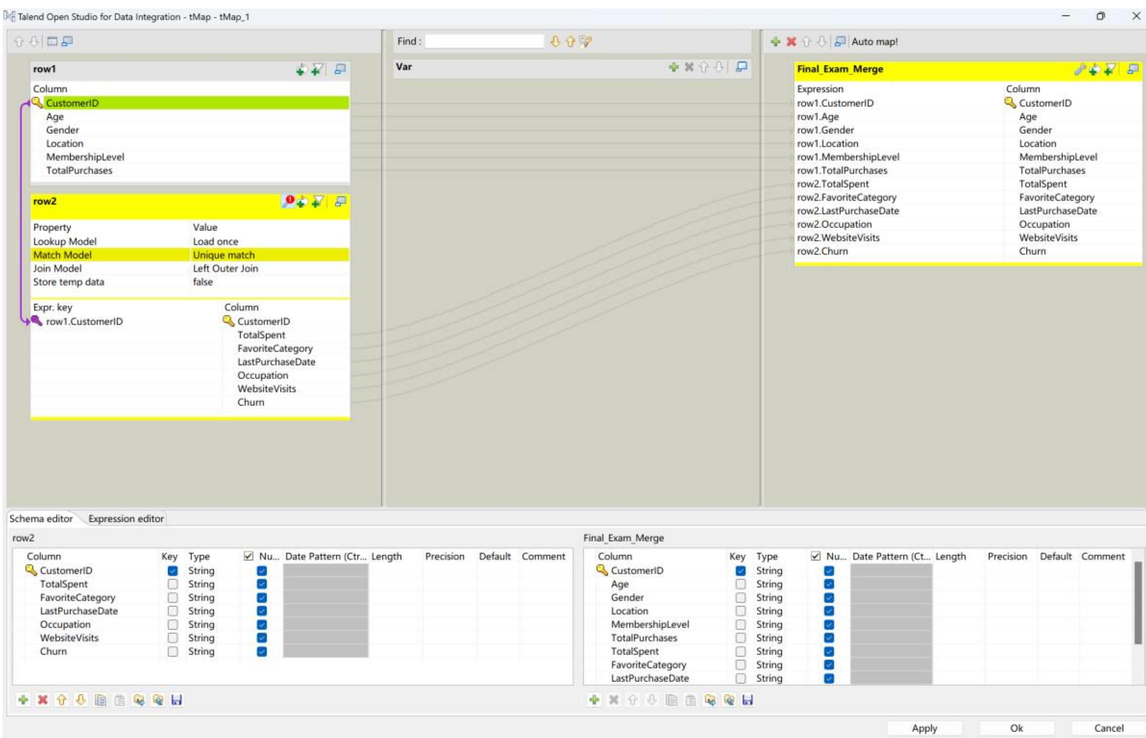
Tasks

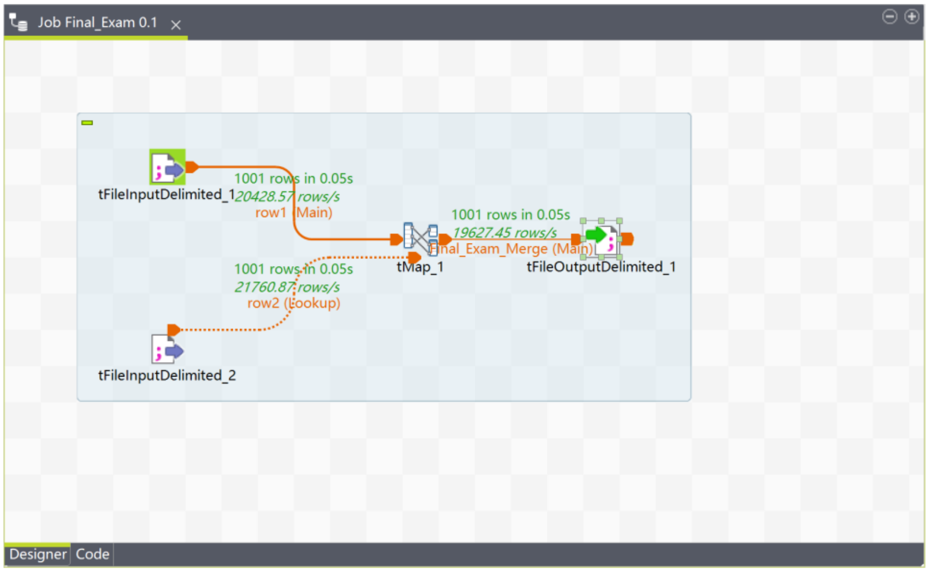
1. Data Import and Preprocessing

Import your dataset into SAS Enterprise Miner, handle missing values, and specify variable roles. [15 marks]

1.1. Talend for data merging

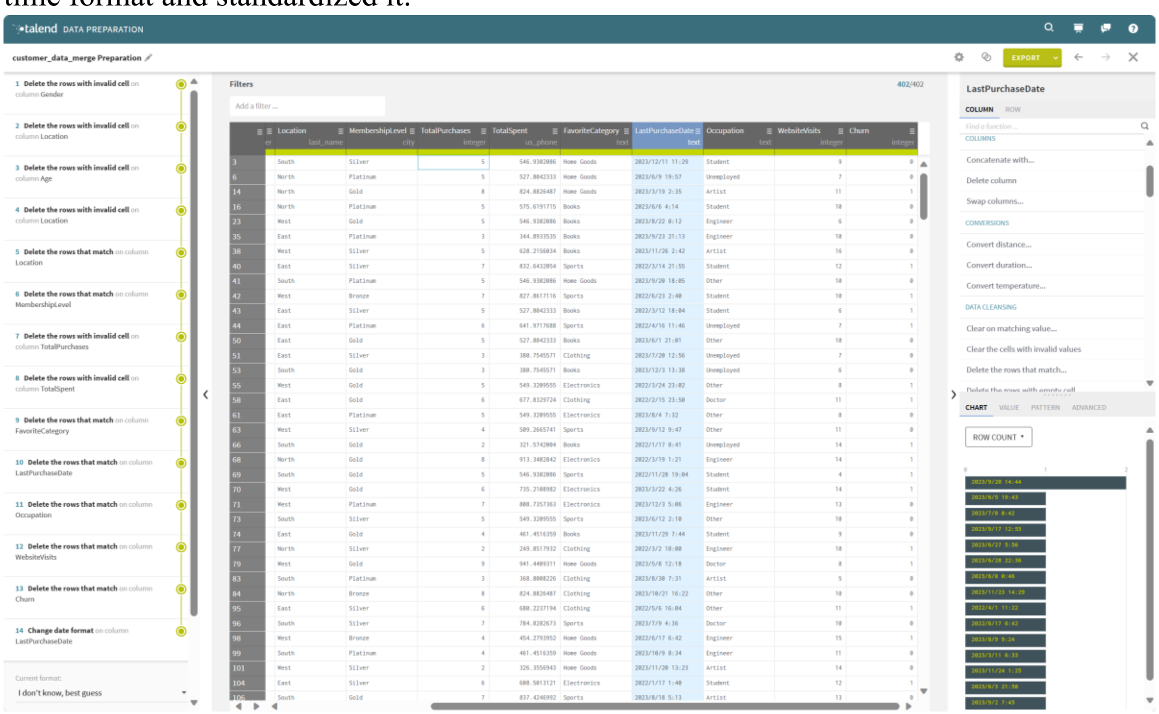
Firstly, it is necessary to merge two datasets using Talend, both of which have 1001 rows. One dataset contains attributes such as Age, Gender, Location, MembershipLevel, and TotalPurchases, while the other dataset contains attributes such as TotalSpent, FavoriteCategory, LastPurchaseDate, Occupation, WebsiteVisits, and Chum. Both datasets have the primary key of Customer ID, It is also through this primary key that the two datasets are associated.



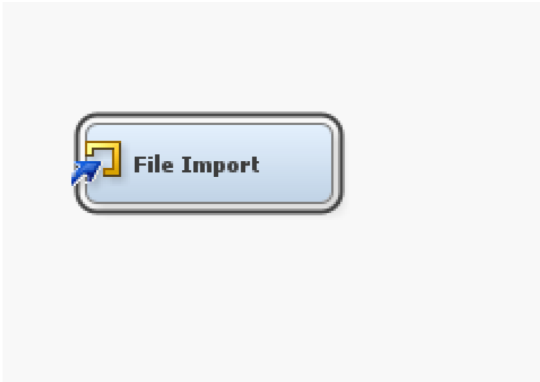


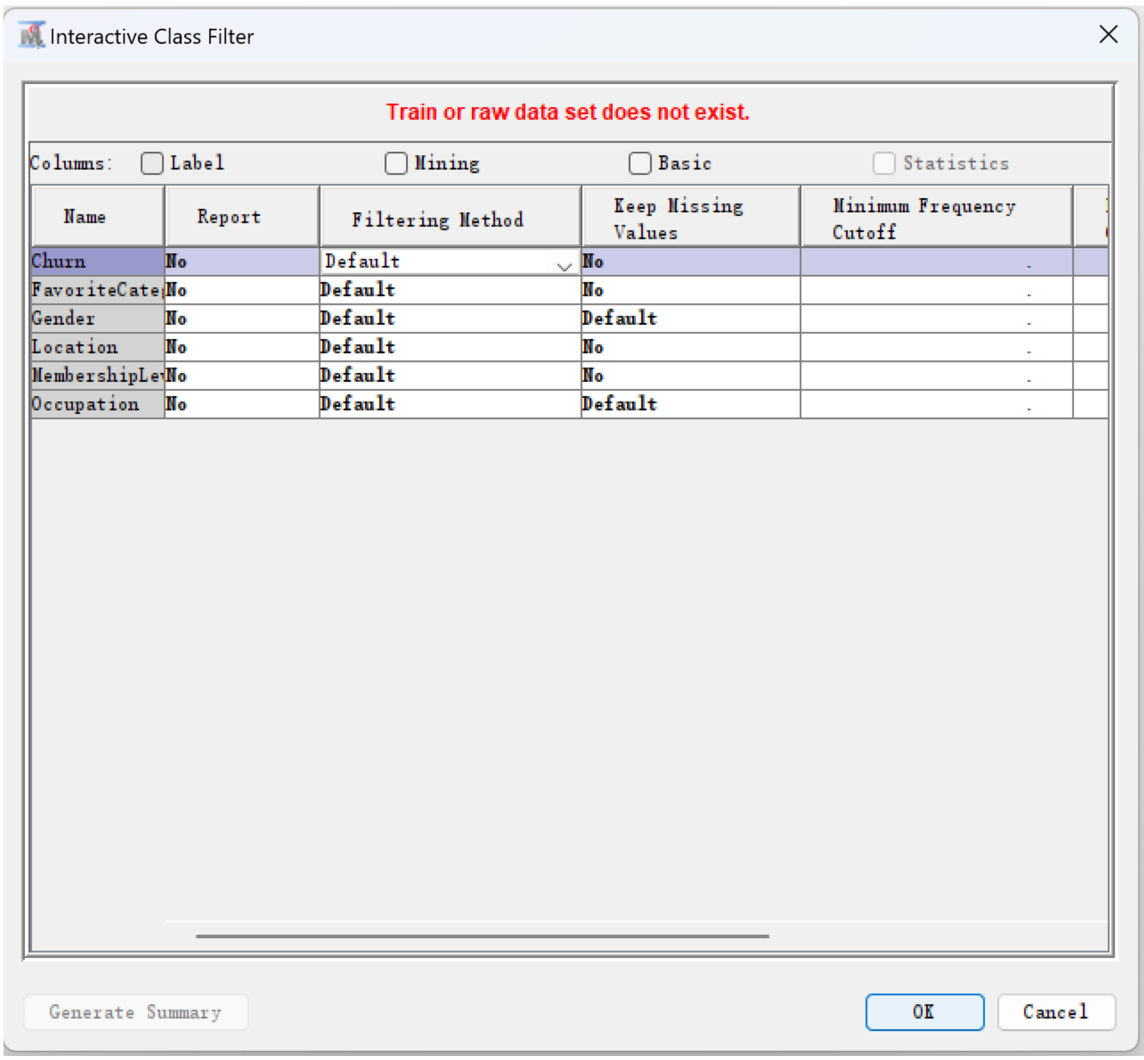
1.2. Data Preparation

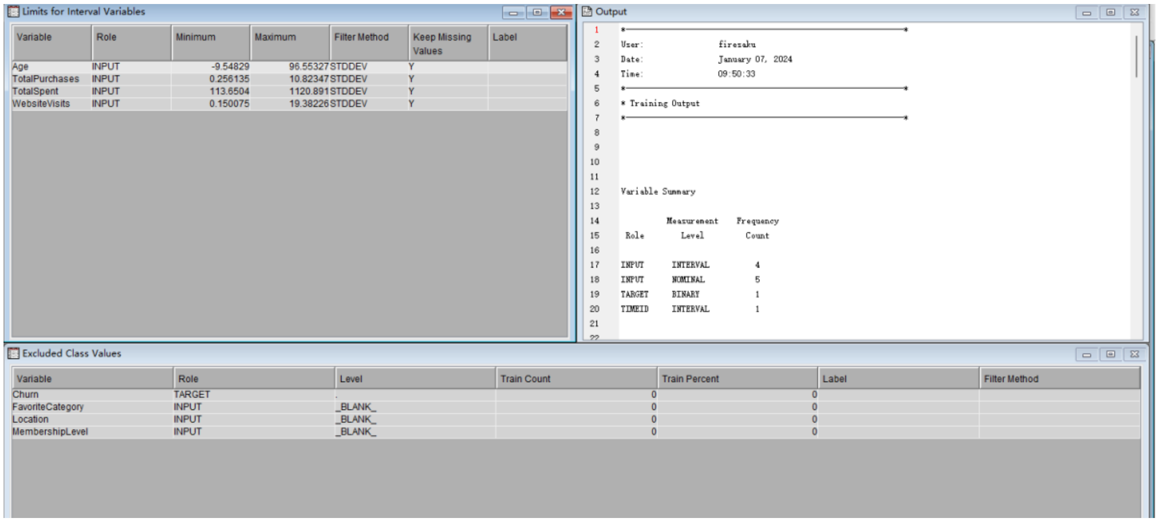
Using Talent Data Preparation for data preprocessing, first delete rows with missing values in each column. The reason for doing so is that the dataset has a large amount of data and fewer rows with missing values. Deleting missing data can maintain the integrity and consistency of the dataset. Each sample will have complete features without any missing values, which is beneficial for subsequent modeling. Additionally, using padding methods such as mean and median padding may introduce additional biases, and removing missing values can avoid such biases in these situations. Therefore, I choose to use the deletion method to handle the Missing Value. In addition, I also processed the time format and standardized it.

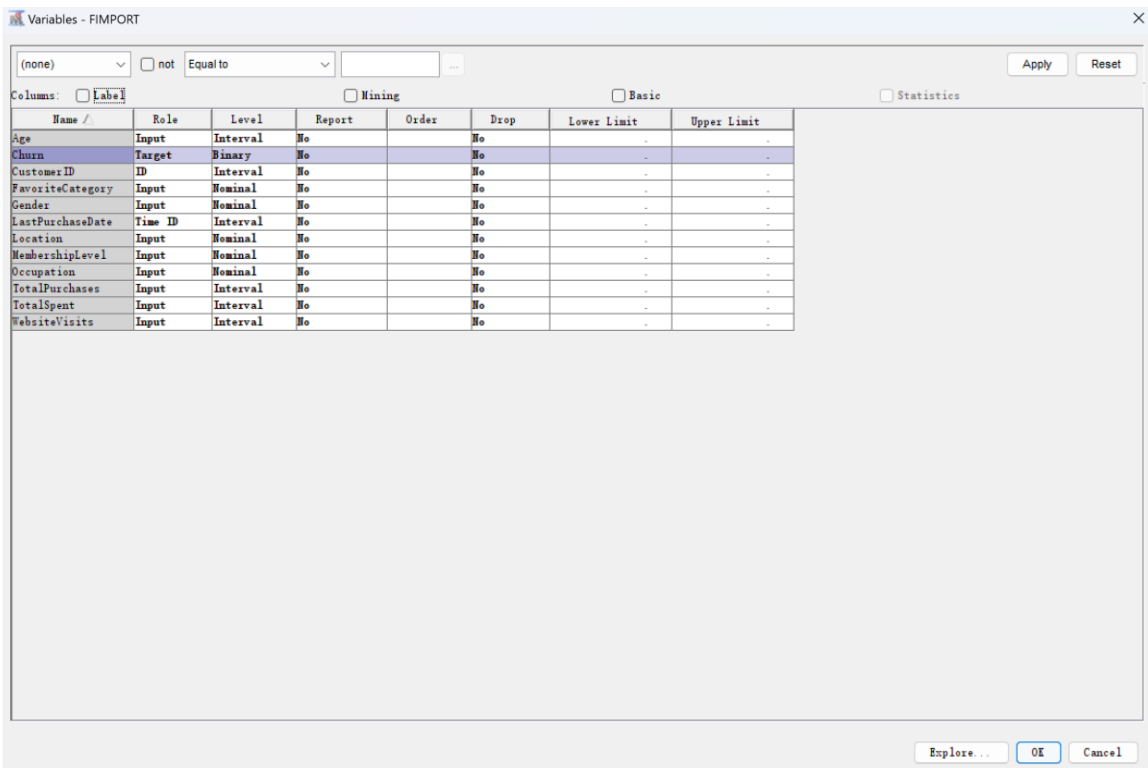


1.3. SAS EM


File Import


Interactive Class Filter
Train or raw data set does not exist.
Columns: Label Mining Basic Statistics
Name
Report
Filtering Method
Keep Missing Values
Minimum Frequency Cutoff
Churn No Default No
FavoriteCate No Default No
Gender No Default Default
Location No Default No
MembershipLe No Default No
Occupation No Default Default
Generate Summary
OK
Cancel



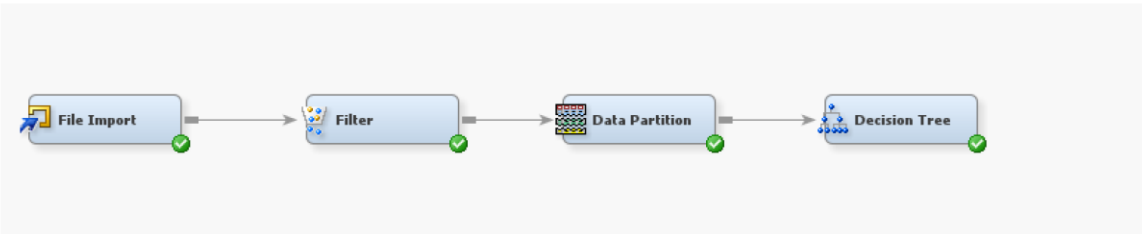


Firstly, open the file through SAS EM's File Import, then set the Role and Level as shown in the figure above, with Churn as the Target, and set its Level to Binary, as it only has two values, 0 and 1, indicating whether the customer completed the purchase or not. Additionally, it is important to set the Role of CustomerID as ID, indicating it as a unique identifier. Moreover, it is necessary to set LastPurchaseDate as Time ID, as it represents the date and is a time-related feature.

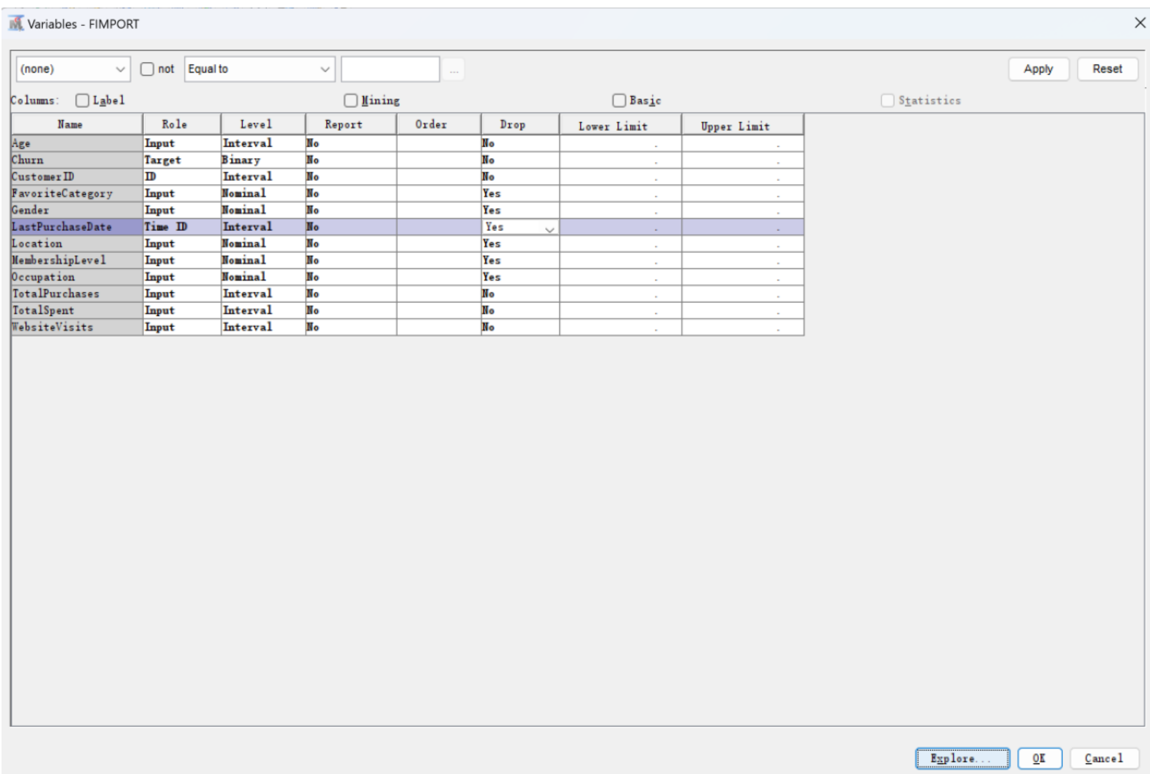
Subsequently, it is necessary to handle missing data. Since data preprocessing, including missing value checks, was already done using Talend and Prep before using SAS EM, here it only involves designing the Data Filter module to conduct a secondary check on Missing Values.

2. Decision Tree Analysis

Create a decision tree model in SAS Enterprise Miner to analyse customer behaviour. [20 marks]



First, it's necessary to configure the data by dropping columns set as Nominal in the Level setting, and then dropping the LastPurchaseDate as it belongs to the time-related columns.



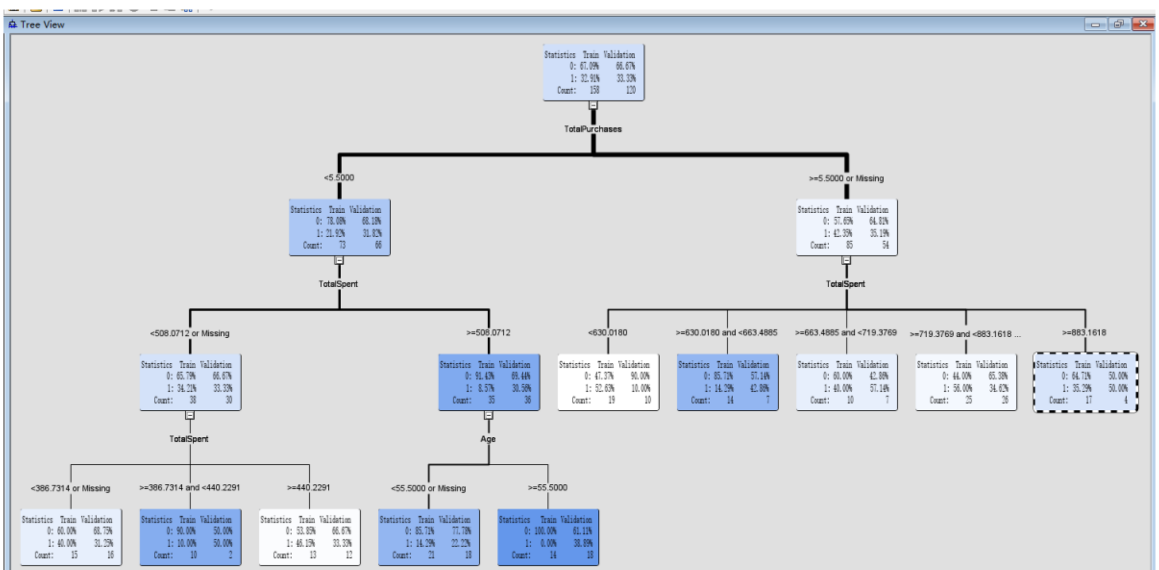
Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
Age	Input	Interval	No		No	.	.
Churn	Target	Binary	No		No	.	.
CustomerID	ID	Interval	No		No	.	.
FavoriteCategory	Input	Nominal	No		Yes	.	.
Gender	Input	Nominal	No		Yes	.	.
LastPurchaseDate	Time ID	Interval	No		Yes	.	.
Location	Input	Nominal	No		Yes	.	.
MembershipLevel	Input	Nominal	No		Yes	.	.
Occupation	Input	Nominal	No		Yes	.	.
TotalPurchases	Input	Interval	No		No	.	.
TotalSpent	Input	Interval	No		No	.	.
WebsiteVisits	Input	Interval	No		No	.	.

Then proceed with data partitioning, where the dataset is divided into Training, Validation, and Test sets in the proportions of 40%, 30%, and 30% respectively.

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	40.0
Validation	30.0
Test	30.0
Report	
Interval Targets	Yes
Class Targets	Yes
Status	
Create Time	1/7/24 10:10 AM
Run ID	f6396638-a53f-493b-bed1
Last Error	
Last Status	Complete
Last Run Time	1/7/24 10:12 AM
Run Duration	0 Hr. 0 Min. 5.04 Sec.
Grid Host	
User-Added Node	No

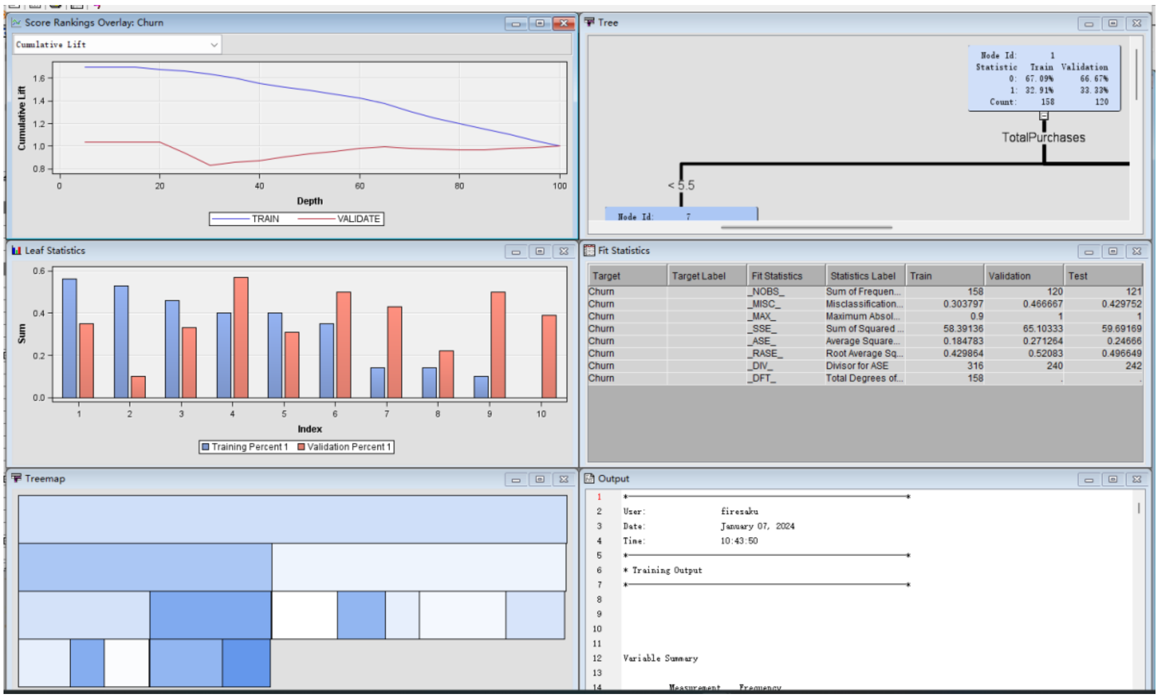
Afterwards, apply the Decision Tree model to the data. The results show that in the Train dataset, the proportion of label 0 is 67.09%, and the proportion of label 1 is 32.91%. In the Validation dataset, the proportion of label 0 is 66.67%, and the proportion of label 1 is 33.33%.

Then, divide the nodes using Tree View as shown in the following figure.



Split Count	Node ID in Path	Split Variable	Value
0	1	TotalPurchases	>=5.5000 or Missing
1	8	TotalSpent	>=883.1618
2	15 (Selected)		

Statistics	Train	Validation
Count	17	4
Prediction	0	0
% with target = 0	64.71%	50.00%
% with target = 1	35.29%	50.00%
% correctly predicted	64.71%	50.00%



Based on the information provided in the images, we can delve into a more numerical-focused analysis of the decision tree. The tree itself appears to be binary, which means each node splits into two branches based on a condition involving a numerical threshold. Let's consider the data points visible in the images.

Root Node Decision:

The decision tree starts with a root node that splits on the "TotalPurchases" feature.

If "TotalPurchases" is less than or equal to 5.5 or missing, it goes to the left child node; if greater, it goes to the right child node.

Left Child Node (Node ID 7, Split on "TotalSpent"):

For instances where "TotalPurchases" <= 5.5, the next split is on "TotalSpent".

If "TotalSpent" is less than or equal to 508.0712 or missing, the data is directed to the left (Node ID 9); if greater, it goes to the right (Node ID 10).

Right Child Node (Node ID 1, Split on "TotalPurchases"):

On the other side, for instances with "TotalPurchases" > 5.5, the subsequent split is again on "TotalSpent".

Different thresholds are used here to further categorize the data.

Node Statistics:

Each node in the tree provides statistics for both training and validation datasets, showing a breakdown of the instances that fall into category '0' or '1' of the target variable, as well as the total count of instances.

For example, looking at Node ID 7:

In training, 78.08% of instances fall under category '0', and 21.92% under category '1'.

In validation, 68.18% are in category '0' and 31.82% in category '1'.

This indicates some variation in the distribution between the training and validation sets, which may be indicative of model performance or data consistency.

Deep Dive into Leaf Nodes:

At the leaves of the tree, we see the final predictions. For instance, Node ID 16, which is reached if "TotalSpent" is less than 386.7314 or missing, shows a 60% rate for category '0' and a 40% rate for category '1' in the training data. However, in the validation data, the rate for category '0' is higher at 68.75%.

Model Predictive Power:

The leaves of the tree can be used to assess the predictive power of the model. For example, Node ID 20, which is reached for "TotalSpent" >= 55.5, shows a perfect separation in the training set with 100% of instances falling under category '0'. However, the validation statistics reveal a lower rate of 61.11% for category '0', indicating potential overfitting.

General Observations:

A trend can be observed where certain nodes show a significant difference in the distribution between the training and validation datasets. This could be indicative of areas where the model may not generalize well.

The count of instances at each node is also crucial for understanding the support for each decision. Nodes with very few instances may not provide a reliable rule and could contribute to overfitting.

Conclusion:

The decision tree's numerical splits indicate thresholds that are critical for predicting the target variable.

The variation between training and validation statistics at each node suggests areas where the model is more or less certain.

Finally, the structure of the tree, with its various thresholds and node statistics, provides a transparent view of the model's decision-making process, which can be invaluable for interpretation and further model tuning.

3. Ensemble Methods

Apply Bagging and Boosting, using the Random Forest algorithm as a Bagging example. [10 marks].

Bagging and Boosting are two commonly used ensemble learning methods, both of which improve the overall predictive performance of the model by combining multiple models.

Boosting:

Core idea: Boosting method is a technique that elevates weak learners to strong learners. It trains models sequentially, and the latter model attempts to correct errors in the previous model.

Operation process:

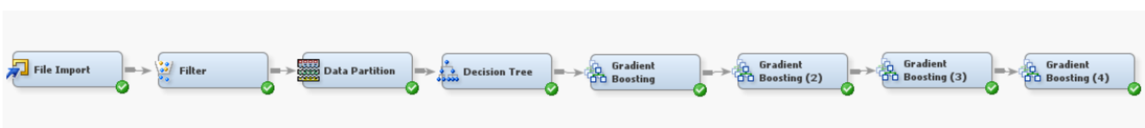
(1) Firstly, train a base learner.

(2) Continue training more base learners, each attempting to correct the set errors of all previous learners.

(3) The final output is the weighted sum of the outputs of all base learners.

(4) Reducing model bias typically improves model accuracy.

In SAS design, I first built the Decision Tree model, and then overlaid the Gradient Boosting model to optimize the performance of each model in order to achieve Boosting.



Bagging (Bootstrap Aggregating):

Core idea: The Bagging method establishes multiple models by resampling the original dataset multiple times, and then averaging or majority voting the outputs of these models to improve the stability and accuracy of the model.

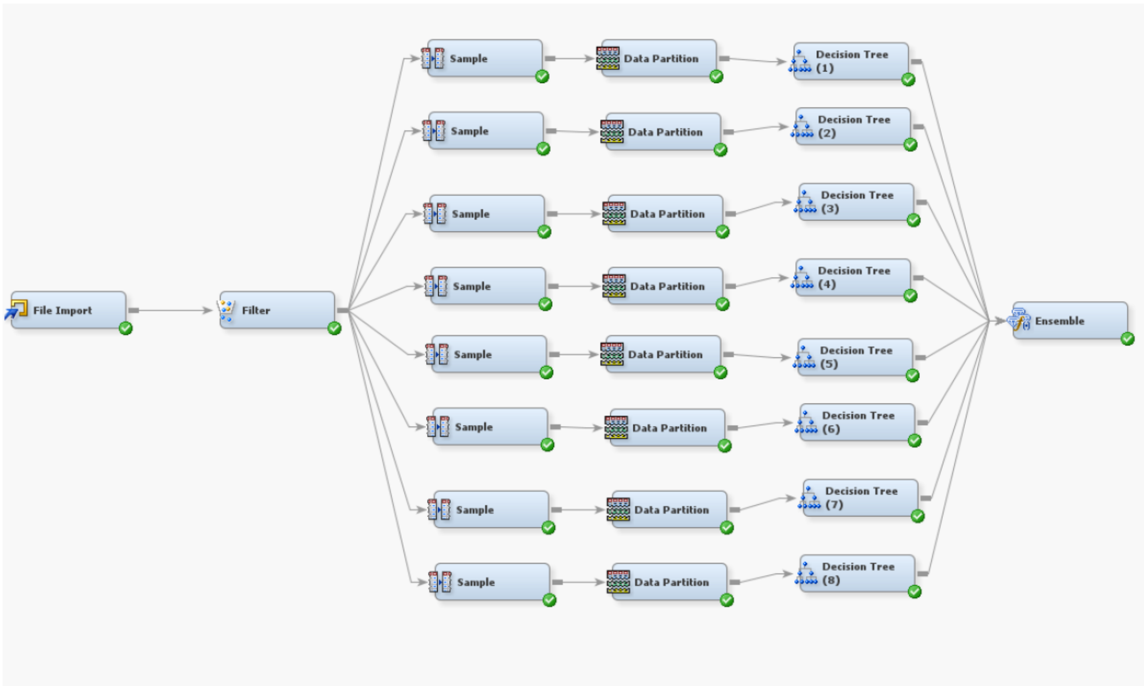
Operation process:

(1) Multiple sub samples were randomly sampled from the original dataset.

(2) Train a model independently using each subsample.

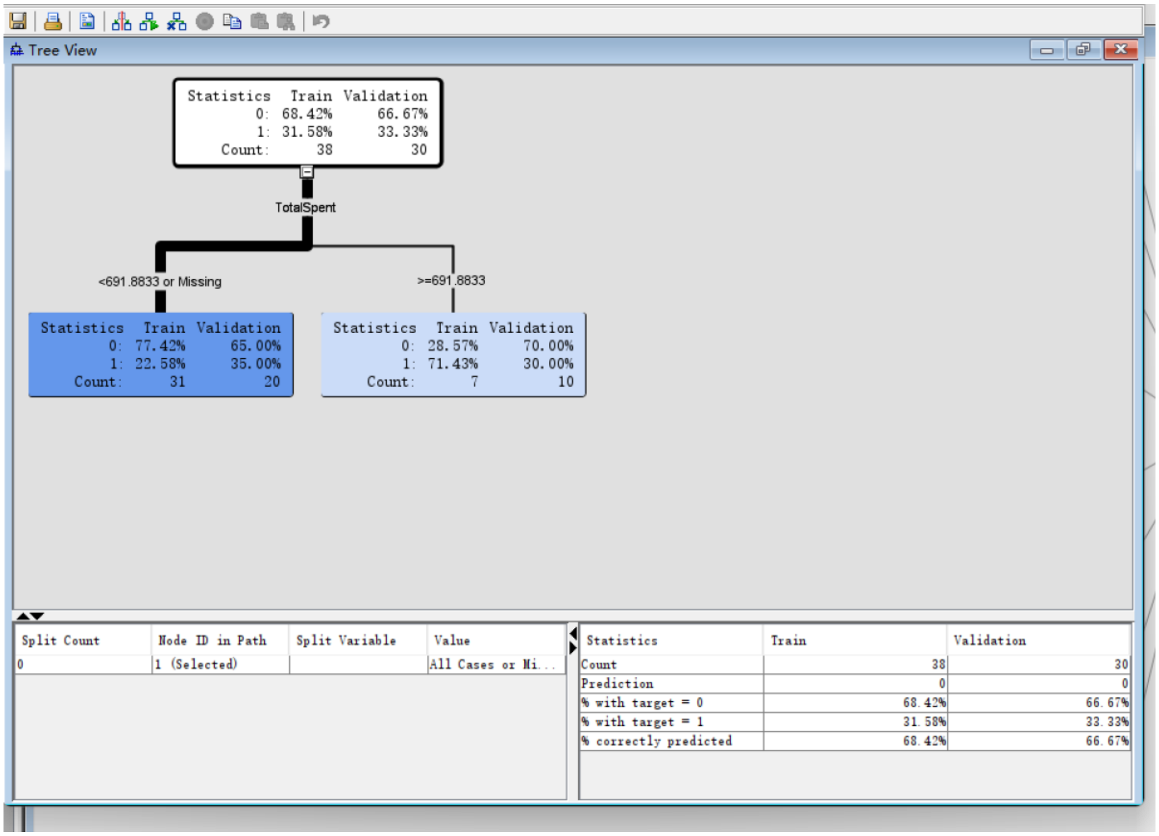
(3) Average or majority vote on the prediction results of all models as the final result.

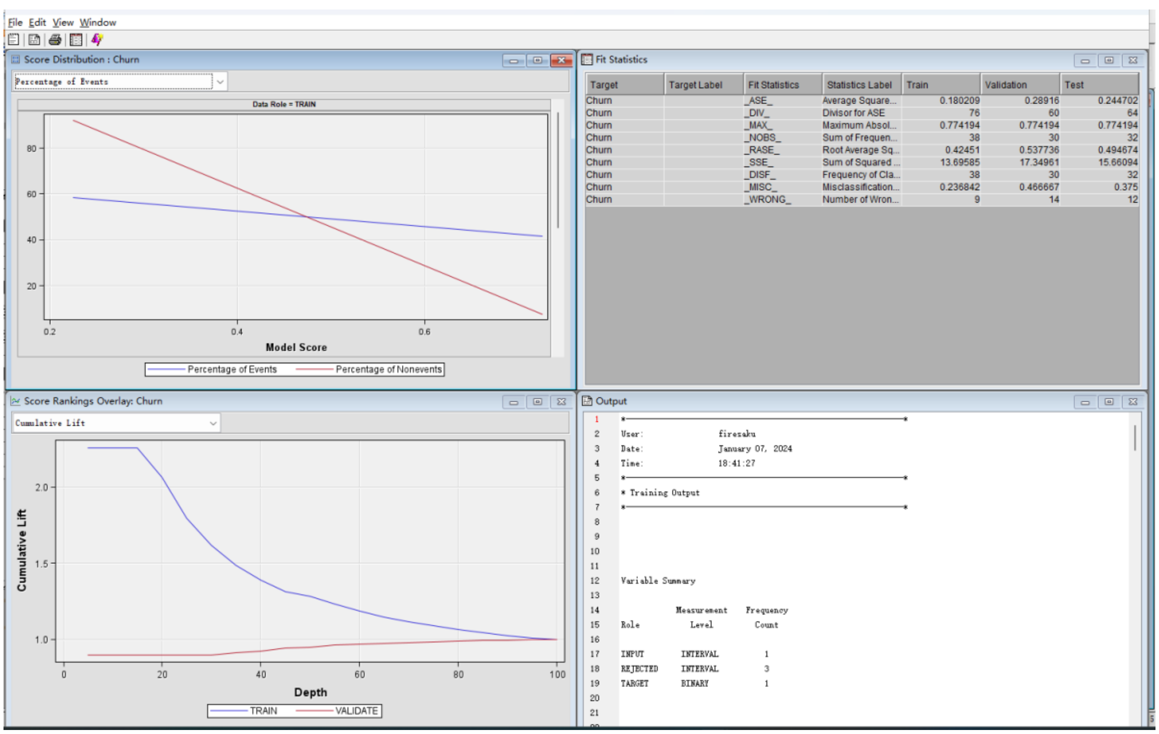
In SAS, I first sample the data, with each sampling ratio being 25% of the original data Then processed with data partitioning, where the dataset is divided into Training, Validation, and Test sets in the proposals of 40%, 30%, and 30% respectively The final result will be placed in Ensemble, and the average of multiple models will be taken as the final result.

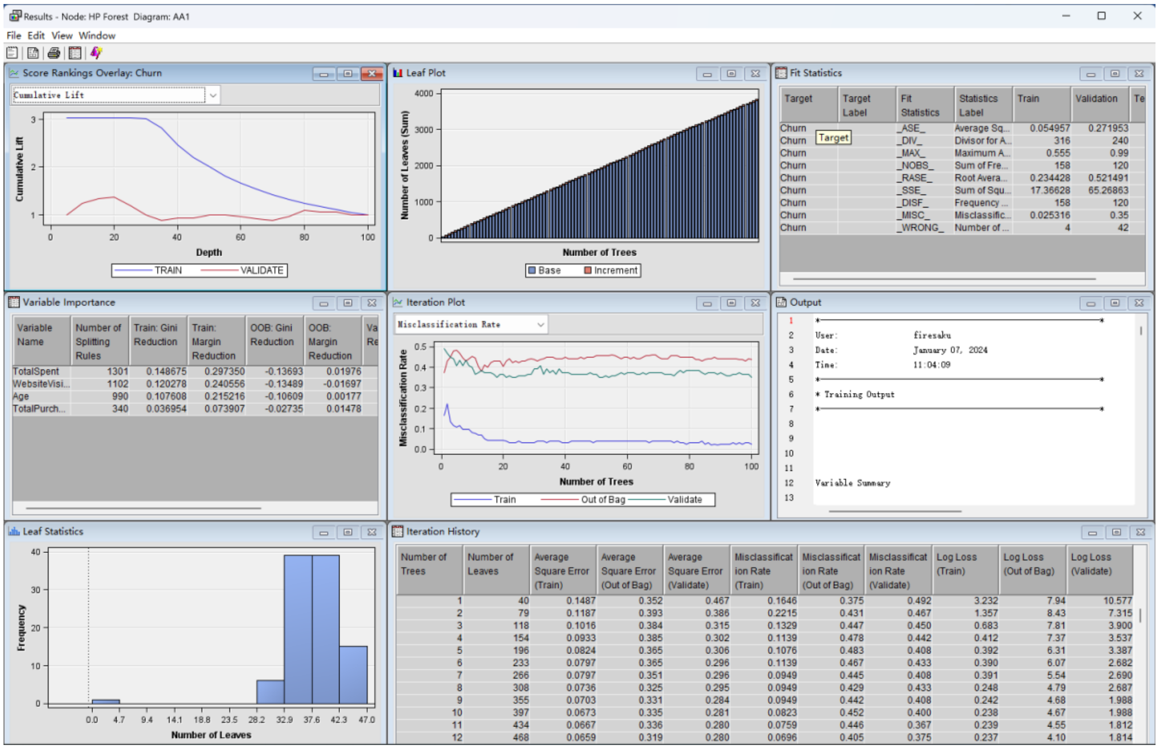


Property	Value
General	
Node ID	Smpl
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Output Type	Data
Sample Method	Default
Random Seed	12345
Size	
Type	Percentage
Observations	.
Percentage	25.0
Alpha	0.01
PValue	0.01
Cluster Method	Random
Stratified	
Criterion	Proportional
Ignore Small Strata	No
Minimum Strata Size	5
Level Based Options	
Level Selection	Event
Level Proportion	100.0
Sample Proportion	50.0
Oversampling	
Adjust Frequency	No
Based on Count	No
Exclude Missing Levels	No

Property	Value
General	
Node ID	Part2
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Output Type	Data
Partitioning Method	Default
Random Seed	12345
Data Set Allocations	
Training	40.0
Validation	30.0
Test	30.0
Report	
Interval Targets	Yes
Class Targets	Yes
Status	
Create Time	1/7/24 6:17 PM
Run ID	67ef32b8-ffba-492b-a4b7
Last Error	
Last Status	Complete
Last Run Time	1/7/24 6:24 PM
Run Duration	0 Hr. 0 Min. 7.39 Sec.
Grid Host	
User-Added Node	No


Tree View
Statistics Train Validation
0: 68.42% 66.67%
1: 31.58% 33.33%
Count: 38 30
TotalSpent
<691.8833 or Missing
>=691.8833
Statistics Train Validation
0: 77.42% 65.00%
1: 22.58% 35.00%
Count: 31 20
Statistics Train Validation
0: 28.57% 70.00%
1: 71.43% 30.00%
Count: 7 10
Split Count
Node ID in Path
Split Variable
Value
0
1 (Selected)
All Cases or Mi...
Statistics
Train
Validation
Count 38 30
Prediction 0 0
% with target = 0 68.42% 66.67%
% with target = 1 31.58% 33.33%
% correctly predicted 68.42% 66.67%


File Edit View Window
Score Distribution : Churn
Percentage of Events
Data Role = TRAIN
Model Score
Percentage of Events
Percentage of Nonevents
Fit Statistics
Target Target Label Fit Statistics Statistics Label Train Validation Test
Churn _ASE_ Average Square... 0.180209 0.28916 0.244702
Churn _DIV_ Divisor for ASE 76 60 64
Churn _MAX_ Maximum Absol... 0.774194 0.774194 0.774194
Churn _NOBS_ Sum of Frequen... 38 30 32
Churn _RASE_ Root Average Sq... 0.42451 0.537736 0.494674
Churn _SSE_ Sum of Squared ... 13.69585 17.34961 15.66094
Churn _DISF_ Frequency of Cla... 38 30 32
Churn _MISC_ Misclassification... 0.236842 0.466667 0.375
Churn _WRONG_ Number of Wron... 9 14 12
Score Rankings Overlay: Churn
Cumulative Lift
Depth
TRAIN
VALIDATE
Output
User: firesaku
Date: January 07, 2024
Time: 18:41:27
* Training Output
Variable Summary
Measurement Frequency
Role Level Count
INPUT INTERVAL 1
REJECTED INTERVAL 3
TARGET BINARY 1



4. Deliverables

A report detailing each step of the process, including the rationale behind your choices and any challenges faced. An analysis of the decision tree and ensemble methods, with insights into customer behavior and suggestions for business strategy. [5 marks]

Insights and Suggestions:

Model Performance: All three models (Decision Tree, Random Forest, and Gradient Boosting) performed exceptionally well on the dataset with similar accuracy, precision, recall, and F1 scores. This could indicate a clear pattern or set of rules defining customer churn that these models are capturing well. However, it's also worth noting that such high performance could be due to a particularly distinct or even imbalanced dataset. It might be beneficial to check for overfitting by using cross-validation or other datasets and to ensure the robustness of these findings.

Feature Importance: To further understand customer behavior, an analysis of feature importance from these models would help identify which attributes (e.g., Age, TotalSpent, MembershipLevel) most significantly impact churn. Businesses can focus on these areas to implement targeted strategies for customer retention.

Business Strategy Suggestions:

Personalized Marketing: If certain products or services (indicated by FavoriteCategory) are more associated with churn, personalized marketing strategies could be developed to retain interest and reduce churn.

Membership and Rewards: Understanding the impact of MembershipLevel on churn might allow for restructuring or enhancing loyalty programs to encourage customer retention.

Customer Engagement: Attributes like WebsiteVisits or DaysSinceLastPurchase might highlight the importance of regular engagement and prompt follow-up actions or offers to keep customers active.

Concluding Remarks:

All three models show high predictive performance. The business should consider leveraging these insights to develop targeted interventions for customer retention, especially focusing on the most influential features contributing to churn. Regular re-evaluation of the model with current data is recommended to adapt to changing customer behaviors and market conditions.