ALTERNATIVE ASSESSMENT 1 (50 marks) - WEEK 12

Liu Ziwei, S2160012

E-Commerce Customer Behavior Analysis

Background:

In the fast-paced world of e-commerce, understanding and mitigating product returns is critical for maintaining profitability and customer satisfaction. As businesses expand their online presence, the challenge of managing returns has grown exponentially. Therefore, analyzing customer return patterns offers a strategic advantage, allowing companies to identify and address the underlying factors contributing to returns. This not only improves the customer experience but also significantly reduces costs and resource wastage associated with the return process.

This study employs **SAS Enterprise Miner** to further analyze customer return patterns, which is advantageous for refining predictive models and enhancing decision-making strategies in the realm of e-commerce.

Objectives：

1. Design and build a decision tree model and ensemble method to predict customer behavior in SAS Enterprise Miner
2. Interpret the branch basis and results of the decision tree and extract insights into customer behaviors.

Dataset:

This study utilizes a comprehensive dataset titled "E-commerce Customer Behavior and Purchase Dataset," sourced from Kaggle. This synthetic dataset, generated using the Faker Python library, is designed to mirror the multifaceted nature of customer interactions and purchasing patterns in an e-commerce setting. It includes details such as customer demographics, transactional data, and purchase behaviors, making it ideal for a wide array of analyses like customer churn prediction, market basket analysis, and trend forecasting.

The dataset comprises various columns, providing valuable insights in consumer behavior, product preferences and purchasing dynamics. Its structured composition facilitates in-depth data analysis and predictive modeling, offering a rich resource in the e-commerce domain.

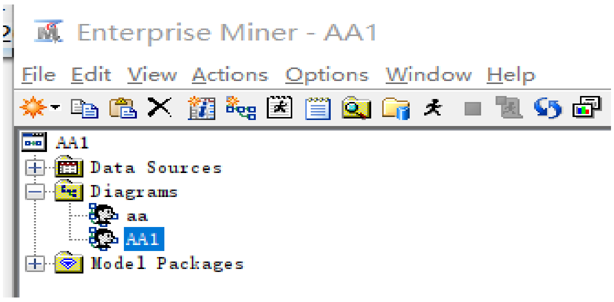
Table 1: E-commerce Customer Behavior and Purchase Dataset

Column Name	Description
Customer ID	A unique identifier for each customer.
Customer Name	The name of the customer (generated by Faker).
Customer Age	The age of the customer (generated by Faker).
Gender	The gender of the customer (generated by Faker).
Purchase Date	The date of each purchase made by the customer.
Product Category	The category or type of the purchased product.
Product Price	The price of the purchased product.
Quantity	The quantity of the product purchased.
Total Purchase Amount	The total amount spent by the customer in each transaction.
Payment Method	The method of payment used by the customer (e.g., credit card, PayPal).
Returns	Whether the customer returned any products from the order (binary: 0 for no return, 1 for return).
Churn	A binary column indicating whether the customer has churned (0 for retained, 1 for churned).

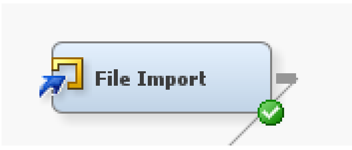
Results

1. Data Import and preprocessing

a. Create a new diagram named AA1



b. Download the dataset and import in the SAS Enterprise Miner



Then, the file is loaded from the local document.

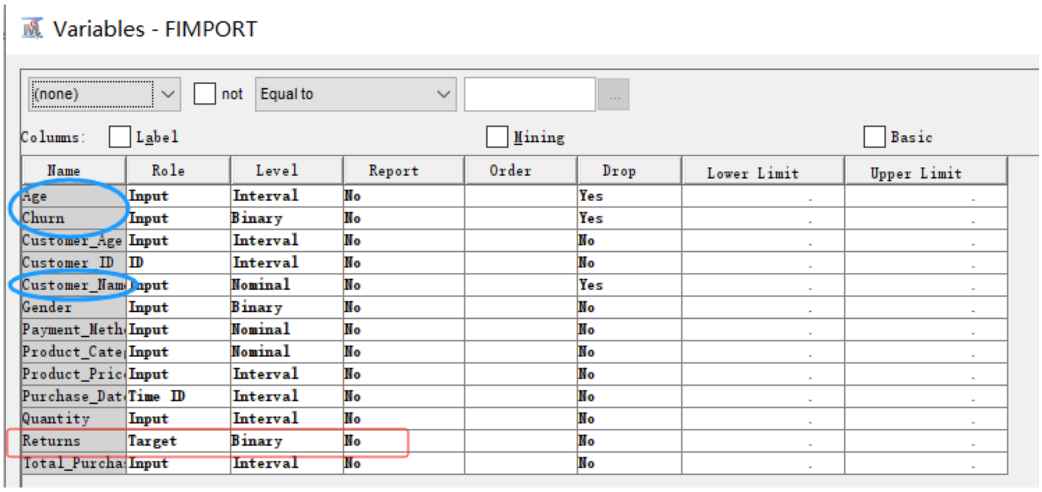
c. Define the features.

a) Target: Returns status

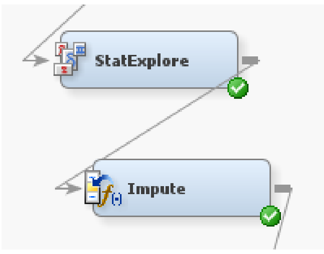
b) Inputs: Age, Gender, Payment Method, Product Category, Product Price, Quantity, Total Purchase Amount,

c) Repeated variables: Age

d) Others: Churn, Customer Name

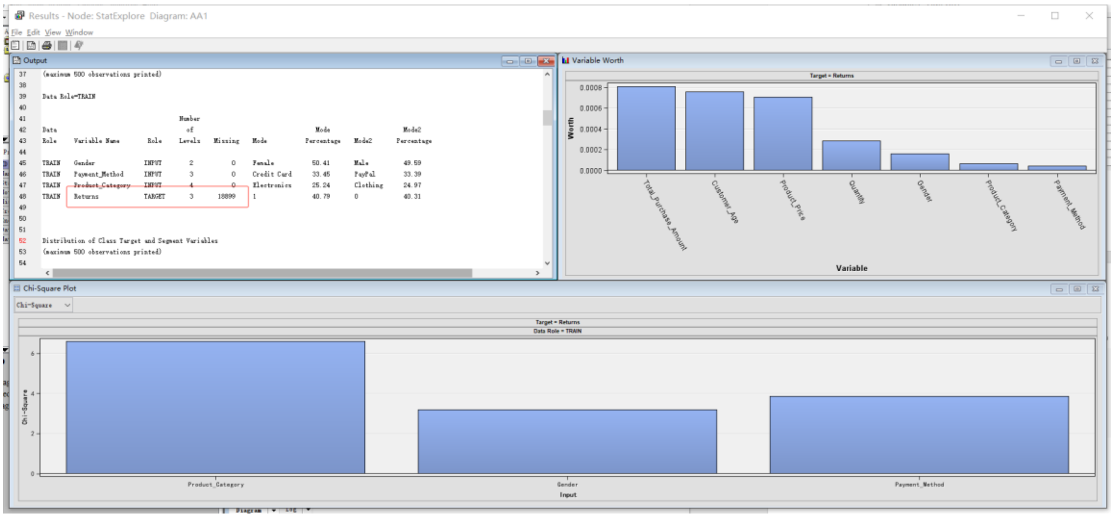


d. Explore the dataset and deal with the missing values



a) Explore the dataset:

We find that there are 18899 missing values in the objective columns ("Returns"), we need to deal with this.



b) Maximum the missing values of "Returns" status

Replacement Editor-WORK.OUTCLASS

Variable	Formatted Value	Replacement Value	Frequency Count	Type	Character Unformatted Value	Numeric Value
Returns	_UNKNOWN_	_DEFAULT_	.	N		.
Returns	.	1	47382	N		.
Returns	0		101142	N		0
Returns	1		101476	N		1
Product_Category	_UNKNOWN_	_DEFAULT_	.	C		.
Product_Category	Books		62247	C	Books	.
Product_Category	Home		62542	C	Home	.
Product_Category	Clothing		62581	C	Clothing	.
Product_Category	Electronics		62630	C	Electronics	.
Payment_Method	_UNKNOWN_	_DEFAULT_	.	C		.
Payment_Method	Cash		83012	C	Cash	.
Payment_Method	PayPal		83441	C	PayPal	.
Payment_Method	Credit Card		83547	C	Credit Card	.
Gender	_UNKNOWN_	_DEFAULT_	.	C		.
Gender	Female		124324	C	Female	.
Gender	Male		125676	C	Male	.

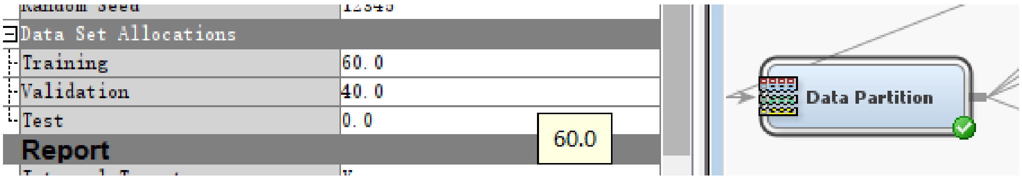
c) Check again the data distribution

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	REP_Gender	INPUT	2	0	Female	50.41	Male	49.59
TRAIN	REP_Payment_Method	INPUT	3	0	Credit Card	33.45	PayPal	33.39
TRAIN	REP_Product_Category	INPUT	4	0	Electronics	25.24	Clothing	24.97
TRAIN	REP_Returns	TARGET	2	0	1	59.69	0	40.31

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
Customer_Age	INPUT	43.72837	15.34341	100000	0	18	44	70	0.024525	-1.21108
Product_Price	INPUT	254.5278	141.6591	100000	0	10	254	500	0.004207	-1.19718
Quantity	INPUT	3.00694	1.414211	100000	0	1	3	5	-0.00754	-1.29992
Total_Purchase_Amount	INPUT	2721.013	1441.776	100000	0	101	2721	5349	-0.00185	-1.1934

e. Data Partition

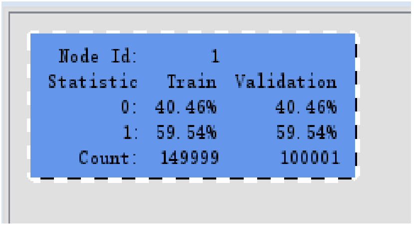
Edit the property for training and validation for 60% and 40%, separately.



2. Decision Tree Analysis

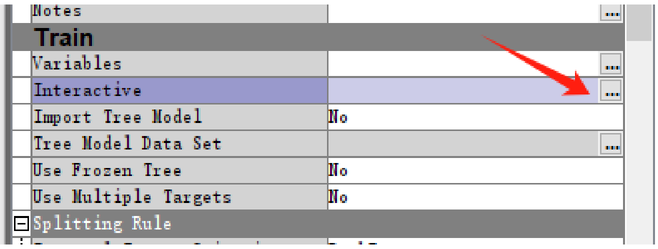
a. Run the classifier.

Click run and see the results. The results show only one node in the tree-based classifier, which cases a low quality in performance. Therefore, we need to set the property.

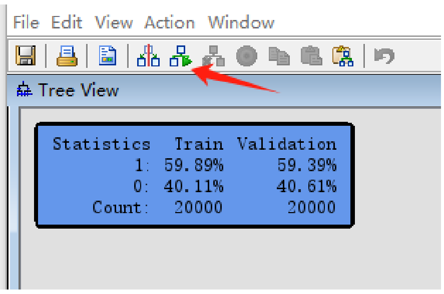


b. Interact with Training in Decision Tree

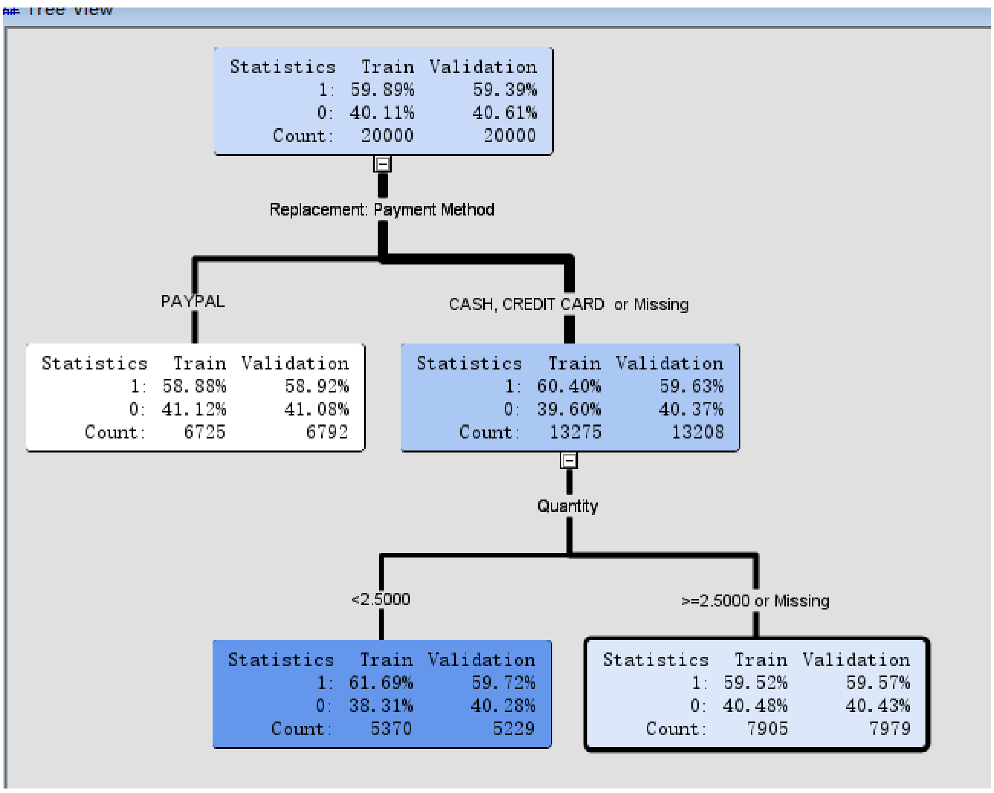
a) Click the decision Tree and move to the property. Click the Interactive in the Train module



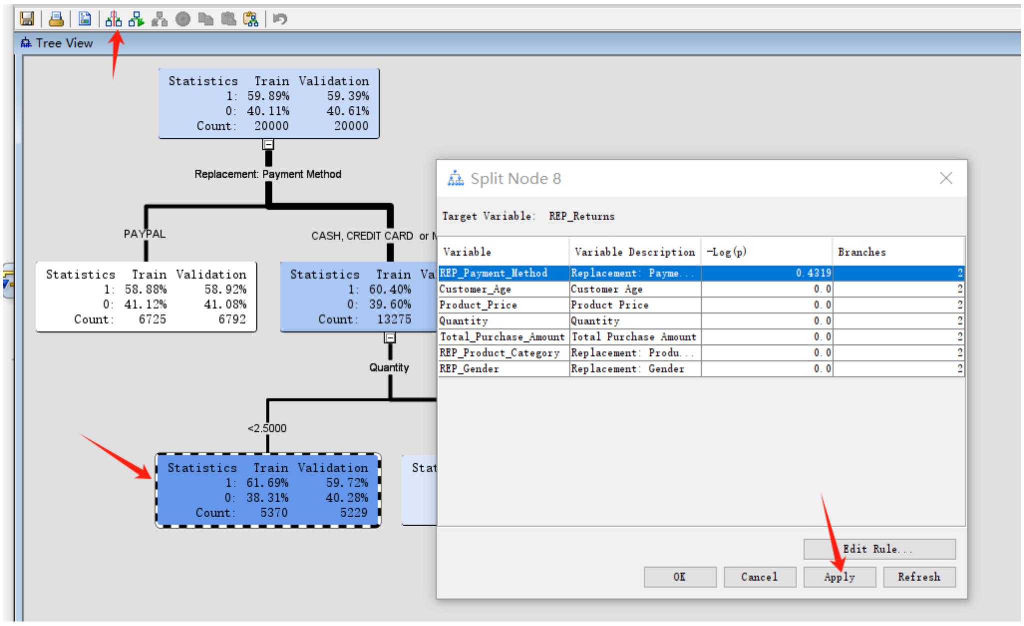
b) Click the first node, then train this node.



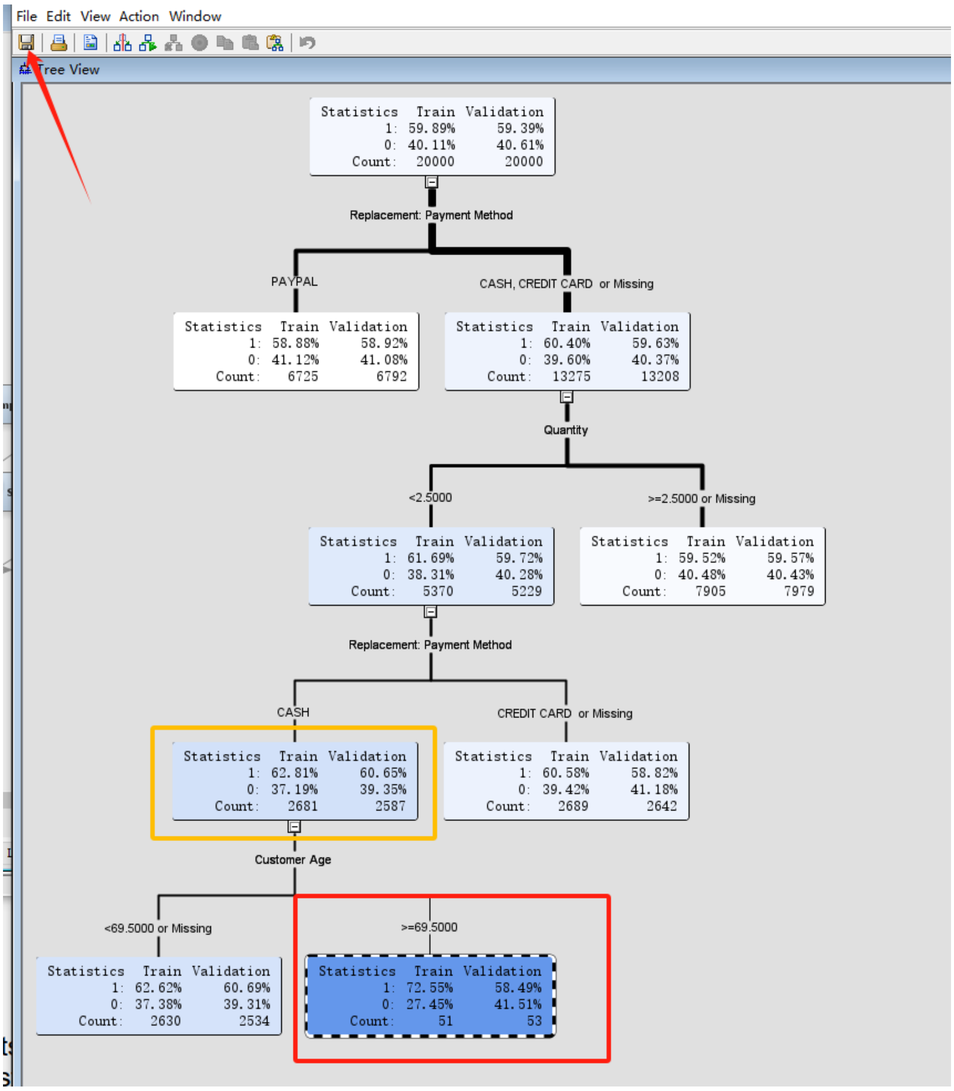
c) Then, we can see the new node by classifier.



d) Click the Split Node in the bar on the top for the new node with the darkest color, which is the best model so far, it will show another new nodes.



e) So far, we could see the result for this tree. Click the save for this tree, or else it cannot save in the later works.



c. Analyze the result

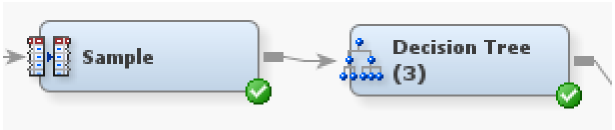
From the tree node, we can see there are 9 nodes and 4 depths. The performance for Train and validation dataset, with 72.55% and 58.49% in return variable, separately. b) When the quantity is less than or equal to 25000, the '1' outcomes are higher (61.69% train, 59.72% validation) than the parent node, indicating a higher likelihood of the target outcome for smaller transactions.

3. Ensemble Methods for Forest Random

Ensemble Methods includes boosting and bagging methods. The bagging technique combines multiple models trained on different subsets of data, whereas boosting trains the model sequentially, focusing on the error made by the previous model.

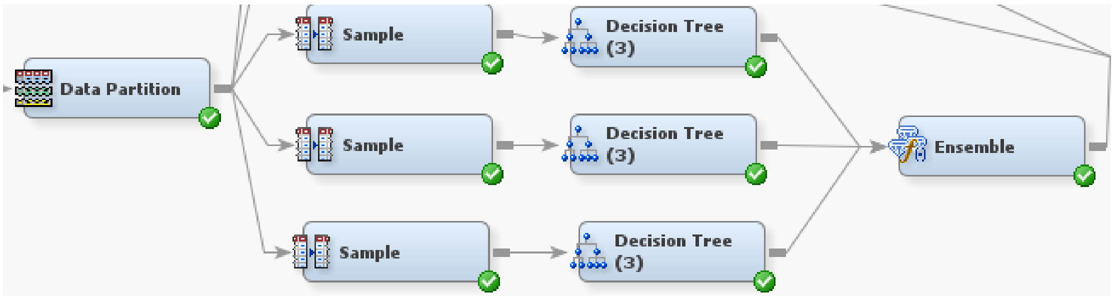
There is a simple method to generate a Forest Random classifier.

a. Define different decision trees as one of the ensemble methods.

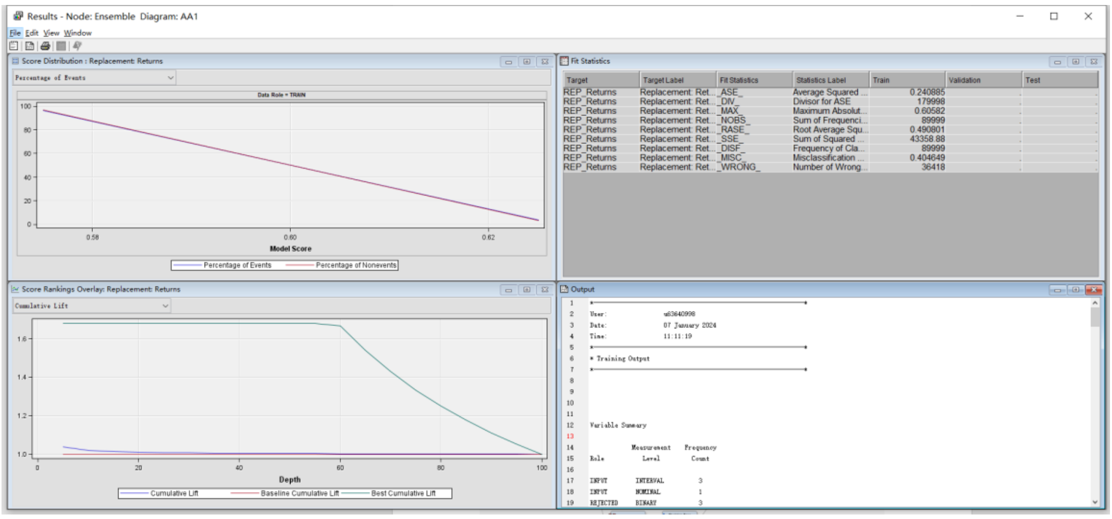


b. Define 3 kinds of Sample method and decision tree. For each sample, there are 60%, 70%, and 80% for training, separately.

c. Connect these classifiers together, into an ensemble.

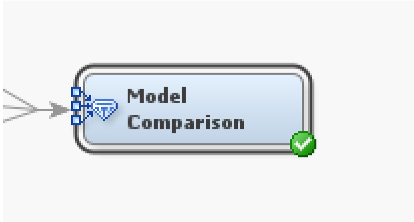


d. The result of Ensemble model

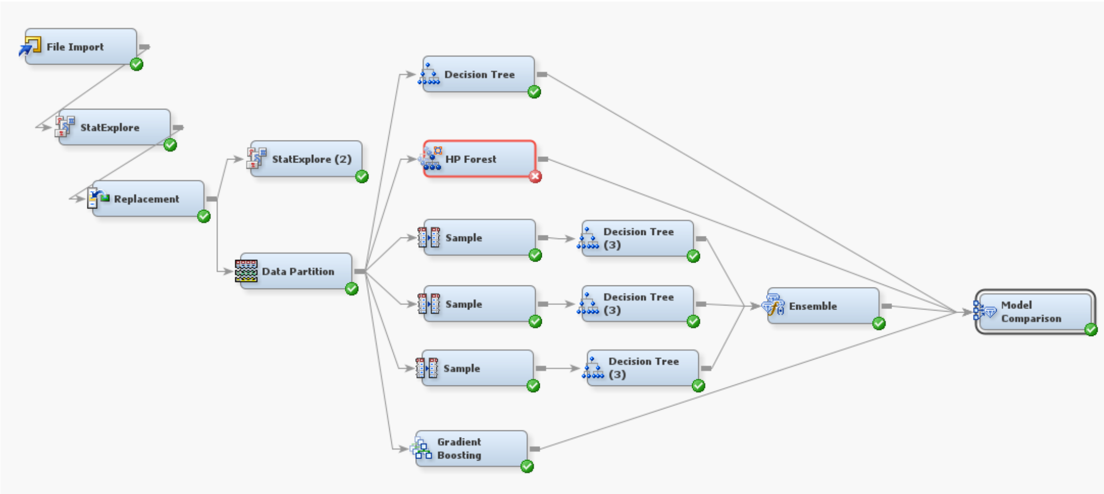


4. Model comparison

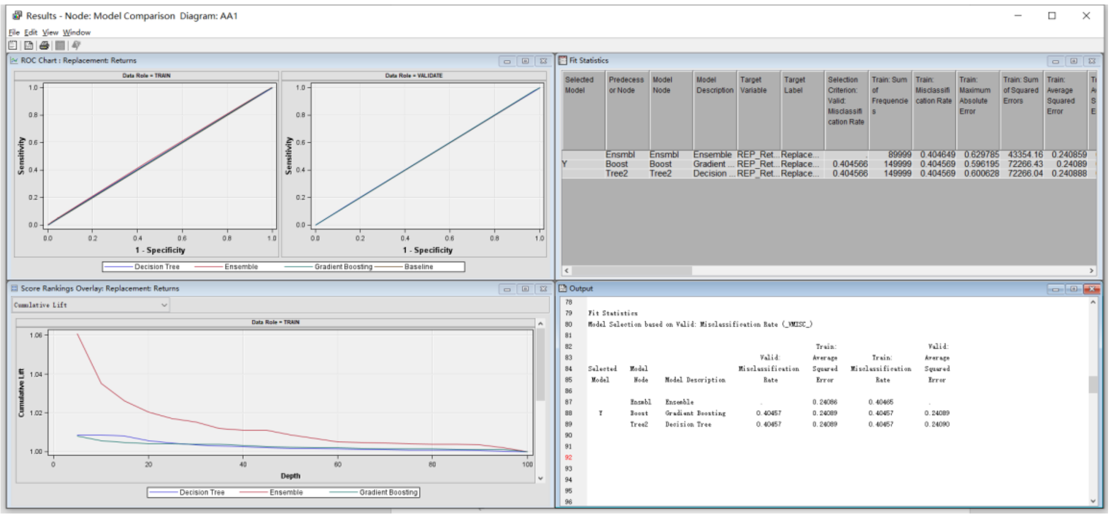
a. Load the module



b. Combine all the classifier; we can have a result for this module.



c. Results:



Fit Statistics
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
	Ensmbl	Ensemble	.	0.24086	0.40465	.
Y	Boost	Gradient Boosting	0.40457	0.24089	0.40457	0.24089
	Tree2	Decision Tree	0.40457	0.24089	0.40457	0.24090

d. Analysis

This ensemble modeling output, utilizing gradient boosting and decision trees in SAS Enterprise Miner, shows the model's performance for predicting 'Replacement: Returns'.

a) Both Area Under the ROC Curve (AUR) values are close to 0.5, suggesting the model barely performs better than random chance.

b) The Gini coefficient is nearly zero, indicating no significant discriminatory power. The misclassification rate is approximately 40% across training and validation, which is quite high.

c) The ensemble's cumulative lift over the baseline is slight, shown by a lift value near 1.

e. Insight and suggestion for customer behavior

Insights:

a) The close-to-random performance of both the decision tree and ensemble methods may suggest that customer return behavior is complex and possibly influenced by a multitude of factors that are not captured by the current model features.

b) When the quantity is less than or equal to 25000, the '1' outcomes are higher (61.69% train, 59.72% validation) than the parent node, indicating a higher likelihood of the target outcome for smaller transactions.

c) From the tree model, we can see Customers using cash, credit card, or with missing payment data have slightly higher '1' outcomes (60.40% train, 59.63% validation) than the overall average, indicating these payment methods might be associated with a higher likelihood of the target outcome.

d) For customers aged over 69,500 or with missing age data, there is a significant decrease in '1' outcomes (72.55% train, 81.58% validation), which is notably higher than any other group analyzed, indicating a strong likelihood of the target outcome occurring in this group.

Suggestion:

a) Consider incentivizing non-cash payment methods if the target outcome is undesirable (e.g., returns), as cash transactions are associated with a higher likelihood of this outcome.

b) Transactions under 5000 units need closer monitoring, especially if paid with cash, due to their higher association with the target outcome.

c) The significant difference in outcomes for the group aged over 69,500 or with missing data suggests a need for a deeper understanding of this segment. They could represent an outlier group with specific behaviors or a data quality issue.

Reflections or Learning Outcomes

1. In this experience, I encountered some issues when run the SAS, and fortunately, some of them are solved. However, there is a problem that can be solved. When I set the HP forest (random forest), I still can not run the process even if I set the property correctly. Therefore, the best solution should be rechecking the process and variables again and make sure the dependent and independent variables are right.
2. In addition, high-performance models like HP Forests have many hyperparameters that can be tuned. An error might suggest that the hyperparameters are not set correctly for the dataset at hand, leading to overfitting, underfitting, or other issues. Therefore, it is vital to make sure all the set is correct.
3. From this study, I understand deeply for random forest and decision tree. We can combine many decision trees together to make a random forest, as an ensemble method. If we use only a random forest model, it would be a bagging method, while connecting many decision trees or other models, it would be a boosting method. The bagging technique combines multiple models trained on different subsets of data, whereas boosting trains the model sequentially.
4. What's more, I learnt how to explain the node meanings for decision tree. The tree starts splitting to subtle, which means that there are some values in the features are cutting off values. There value is the key to classify the object, or even inflect the results in the marketing.

Appendix:

Result for other three decision trees in ensemble model: