**SAS Enterprise Miner**

**Data Import to SAS Enterprise Miner**

A computer screen shot of a computer screen

Description automatically generated A screenshot of a computer

Description automatically generated

In the initial phase, begin by creating a diagram in SAS Enterprise Miner. Then, choose the 'File Import' component from the sample options. Following this, navigate to the side tab, search for 'Import File', and select the path of the pre-processed CSV file already exported from Talend Data Preparation.

**Specify Variable Roles**

A screenshot of a computer

Description automatically generated

In the SAS Enterprise Miner process, specifying the variable roles is a key step to ensure the correct analysis of data. As shown in the variables list, each attribute has been assigned a specific role and level appropriate to its nature. For instance, 'Age', 'TotalSpent', and 'WebsiteVisits' are marked as 'Input' with an 'Interval' level, suitable for continuous data. 'Churn' is set as the 'Target' variable with a 'Nominal' level, as it is a categorical outcome we aim to predict. 'CustomerID' and 'LastPurchaseDate' are labeled as 'ID' and 'Time ID' respectively, recognizing their use as identifiers rather than variables to be analyzed. Other attributes like 'Gender', 'Location', and 'MembershipLevel' are categorized as 'Input' with a 'Nominal' or 'Ordinal' level, indicating their role in the model as categorical predictors. This designation ensures that each variable is treated correctly in the modeling process, facilitating accurate data analysis.

1. **Tasks 2**

***Decision Tree Analysis:*** *Create a decision tree model in SAS Enterprise Miner to analyse customer behaviour.*

A screenshot of a computer

Description automatically generated

As shown in the diagram, establish a decision tree model by linking three components which are file import, data partitioning, and the decision tree. The specific purpose of the component and property configuration data partitioning and Decision Tree will be discussed in the next section.

**Data Partitioning**

A screenshot of a computer

Description automatically generated

In SAS Enterprise Miner, data partitioning is used to split a dataset into separate subsets for model building and validation, allowing for unbiased assessment of model performance. For this model, I use a 70% portion for training and a 30% portion for validation.

**Decision Tree**

A screenshot of a computer

Description automatically generated

When building a decision tree model, the splitting rule based on the nature of our target variable. For interval target variables, the criterion often used is Variance, which minimizes the variance within each split. In cases with nominal target variables, the criterion of choice is typically Entropy, which measures the disorder or impurity of data at each split. Similarly, for ordinal target variables, the Entropy criterion is also favoured. By adapting the splitting rule to your specific target variable, you can optimize the performance and interpretability of your decision tree model.

A screenshot of a computer

Description automatically generated

The figure provides an overall result generated by this model. It encompasses various visual elements, including a score ranking overlay, a tree diagram, fit statistics, a treemap, and more.

Here's the complete tree diagram for the model predicting observed target values as shown in below: -

A computer screen shot of a diagram

Description automatically generated

Tree diagram with 14 splitting rules for the input column “TotalPurchases”.

A close-up of a number

Description automatically generated

According to the Variable Importance table the most important predictors is TotalPurchases.

A screenshot of a computer

Description automatically generated

The decision tree analysis, focusing on 'TotalPurchases', reveals distinct customer segments based on their purchasing behavior. Initial splits in the tree indicate that the number of purchases is a significant predictor of the behavior under study, likely customer churn. The detailed node statistics suggest that as the total number of purchases varies, so does the likelihood of a customer falling into one of two categories, which could signify churned versus active customers.

The further Nodes 4 to 7 suggests more nuanced thresholds of purchase behavior that are influential in predicting outcomes. For instance, certain nodes with higher purchase counts may correlate with a greater likelihood of customer retention, while others with fewer purchases might indicate a risk of churning.

Overall, the decision tree provides actionable insights, suggesting that the frequency of purchases is a key behavioral indicator. This information can be leveraged to tailor customer engagement strategies, such as targeted marketing to increase purchase frequency among customers showing signs of potential churn.

A screenshot of a computer

Description automatically generated

From this Fit Statistics Table, it seems that the misclassification rates maximum error are in acceptable range. The ASE and RASE also suggest that the model's predictions are close to the actual values on average.

The customer behavior analysis, using decision tree modeling, indicates that the frequency of purchases ('Total Purchases') is the most critical factor in determining customer churn, with other variables such as 'Occupation' and 'Age' also playing significant roles. 'Total Purchases' is the principal variable used to split decisions in the predictive tree, suggesting thresholds in purchasing behavior are key indicators of churn risk. The variable importance scores reveal 'Age' as a more significant predictor in the validation set than in training, suggesting demographic factors may influence churn differently across datasets.

The fit statistics exhibit a moderate misclassification rate, with the model being more accurate on training data compared to validation data, hinting at potential overfitting. The Average Squared Error (ASE) and Root Average Squared Error (RASE) point to a moderate error level in predictions, with these errors slightly inflated in the validation set, indicating room for improvement in the model's predictive accuracy.

Overall, the analysis underscores the need to consider how various factors like purchasing frequency, occupation, and age interplay in influencing customer retention. There is also a suggestion of the need to address overfitting and enhance the model's generalization capabilities to better predict churn across different customer segments.

1. **Tasks 3**

**Ensemble Methods: *Apply Bagging and Boosting, using the Random Forest algorithm as a Bagging example.***

A screenshot of a computer

Description automatically generated

In this SAS Enterprise Miner workflow, I've constructed an ensemble methods approach to enhance predictive performance, building upon the foundation laid by the decision tree model established in the earlier task. The ensemble methodology is incorporating two components which are the HP Forest and Gradient Boosting. The HP Forest is for the bagging technique using the Random Forest algorithm, which aggregates multiple decision trees to reduce variance and improve stability. On the other hand, the Gradient Boosting component implements boosting, a method that sequentially builds models with each one focusing on the errors of the previous model to reduce bias. To systematically assess the efficacy of these models, the Model Comparison component is employed to leverages the collective strengths of bagging and boosting to potentially outperform the individual predictive capabilities of the models involved.

A screenshot of a computer

Description automatically generated

The figure provides an overall result generated by the model comparison component. It encompasses of a score ranking overlay, fit statistics, and output file.

A screenshot of a computer

Description automatically generated

The chart displays the performance of three models which are Decision Tree, HP Forest (Bagging), and Gradient Boosting across different complexities, as indicated by depth. For the training data, Gradient Boosting consistently outperforms the other models, maintaining a higher cumulative lift across all depths. In contrast, the Decision Tree and HP Forest show a decreasing lift with increasing depth. On the validation set, the Decision Tree's performance drops as complexity grows, while the HP Forest and Gradient Boosting display more stable trends. Overall, Gradient Boosting seems to offer the best performance, particularly at higher depths, indicating its effectiveness in both training and validation scenarios.

A close-up of a document

Description automatically generated

For the training set, the Decision Tree model has a misclassification rate of 0.41926 and an average squared error of 0.23952. The Gradient Boosting model shows a similar misclassification rate of 0.45399 but has a slightly better average squared error of 0.2417. The HP Forest model presents the highest misclassification rate among the three at 0.48957 with an average squared error of 0.24968. These statistics suggest that while the Decision Tree and Gradient Boosting models have a closer performance in terms of error, the HP Forest model is less accurate on the training data.

**APPENDIX**

**Task 1**

A computer screen shot of a computer screen

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Task 2**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a diagram

Description automatically generated

A close-up of a number

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Task 3**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A close-up of a document

Description automatically generated