UNIVERSITY OF MALAYA

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

MASTER OF DATA SCIENCE

WQD7005 DATA MINING

ALTERNATIVE ASSESSMENT 1

CASE STUDY: E-Commerce Customer Behaviour Analysis

Name	:	Wong Jia Hui
Matric ID	:	S2192852

GitHub Repository Link:
https://github.com/jhwong97/WQD7005-Case-Study-AA1.git

Table of Contents

1 Introduction

1.1 Case Study Overview

In this case study, a dataset related to e-commerce customer behaviour is collected. The dataset will be used to train models like decision tree, boosting and bagging model in predicting the customer churn. To accomplish this, a toolkit comprising Talend Data Integration, Talend Data Preparation, and SAS Enterprise Miner has been strategically selected. These tools, renowned for their proficiency in handling diverse datasets and facilitating advanced analytics, provide the foundation for a comprehensive exploration of E-commerce customer dynamics. The details of the work are explained in this report.

1.2 Selected Tools for Case Study

To execute data integration and preprocessing tasks, tools like Talend Data Integration, Talend Data Preparation, and SAS Enterprise Miner are being employed.

1.3 Objectives

- To obtain or create a dataset which satisfies or closely matches the Alternative Assessment 1 criteria and related to E-commerce customers behaviour analysis.
- To utilise existing tools like Talend Data Integration, Talend Data Preparation and SAS Enterprise Miner to perform data integration, data preprocessing, data analysis and modelling on the collected dataset.
- To create decision tree, bagging and boosting model to predict the customer churn and evaluate their model performance via SAS Enterprise Miner.
- To discuss the potential strategies to enhance the performance of the selected models.
- To suggest potential business strategy for customer retention.

2 Dataset Description

The dataset comprises a total of 24,999 data rows, encompassing diverse customer attributes and their purchase history throughout the year 2023. The description of the dataset attributes is outlined in *Table 2.1*.

Attributes	**Description**
CustomerID	A unique identifier for each customer
Age	Age of the customer
Gender	Gender of the customer (Female, Male)
Location	Geographic location of the customer
Occupation	The occupation of the customer
MembershipLevel	The membership level of the customer in that e-commerce platform
TotalPurchases	Total number of purchases or quantity made by the customer
TotalSpent	Total amount spent by the customer in Ringgit Malaysia (RM)
FavouriteCategory	The category in which the customer most frequently shops
LastPurchaseDate	The date of the last purchase by the customer
PaymentMethod	The payment method in which the customer most frequently uses (Cash, Credit Card, TnG)
Churn	Indicates whether the customer has stopped purchasing (1 for churned, 0 for active)

Table 2.1 Description for dataset attributes.

The datasets from the e-commerce website are divided into two main parts initially, and thus data integration between these two datasets are needed to create the final dataset for modelling purpose in SAS Enterprise Miner.

3 Data Integration using Talend Data Integration

There are a total of two datasets gathered. Therefore, Talend Data Integration is used to merge the two sets of data. The process of the data integration using Talend Data Integration is illustrated in *Figure 3.1*, where components like tFileInputDelimit, tUnite and tFileOutputDelimited are used.

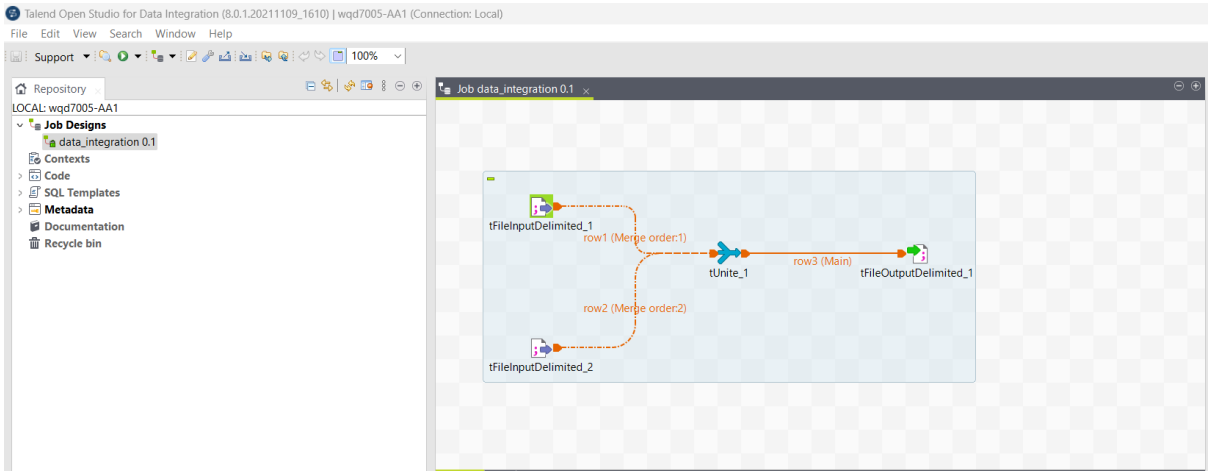

Figure 3.1 Data integration process using Talend Data Integration.

4 Basic Data Cleaning using Talend Data Preparation

Prior to importing the integrated dataset to SAS Enterprise Miner, Talend Data Preparation is used to provide a quick and easy way for data profiling of the dataset. It allows us to have a quick understanding of our dataset and identify if there are any invalid or missing values to be treated as shown in *Figure 4.1*.

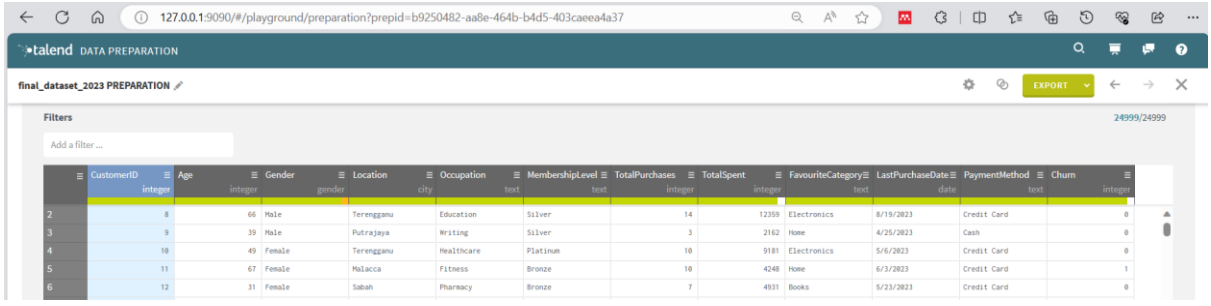

Figure 4.1 Data profiling and basic data cleaning using Talend Data Preparation.

Based on *Figure 4.1*, the dataset exhibits invalid entries in the "Gender" attribute and missing values in the "TotalSpent" and "Churn" attributes. During this stage, the correction of invalid values will primarily rely on the Talend Data Preparation Tools, while addressing the missing values will be carried out through the utilization of SAS Enterprise Miner.

From the data profiling of “Gender” attributes, I found that the invalid values are contributed by the typing errors for the word “Female” and “Male”. Therefore, by utilising the built-in function - “find and group similar text…”, a quick fix on the data can be made as shown in *Figure 4.2*.

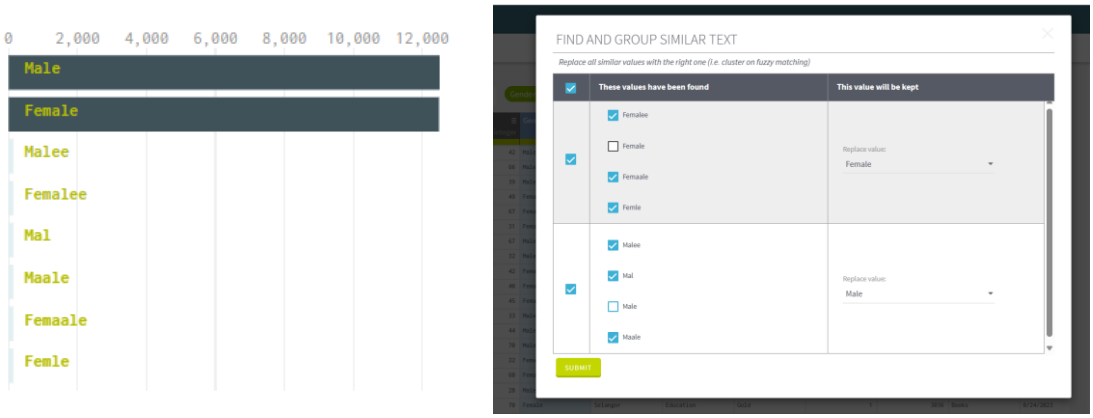

Figure 4.2 The left image shows the data profiling for “Gender” attribute. The right image shows the method used to correct the invalid values.

Although, the tool does not show invalid value for “Location” attribute, but I’ve found out there are inconsistencies in format for the label “Kuala Lumpur” as the abbreviation of the word like “KL” is found in the dataset. As the built-in function - “find and group similar text…” is not able to detect any cluster, I use the “replace the cells that match …” function to replace “KL” to “Kuala Lumpur” as shown in *Figure 4.3*.

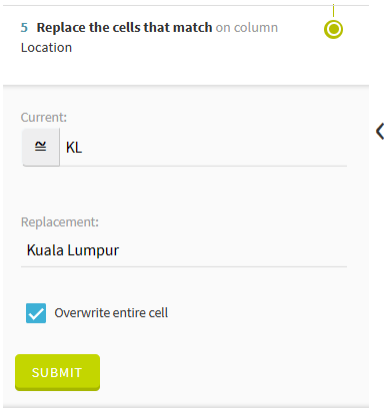

Figure 4.3 Replacing the word "KL" with "Kuala Lumpur".

In addition, I have also utilized Talend Data Preparation to check for data consistency like the date format for “LastPurchaseDate” attribute in the dataset. From the data profiling, it shows that the date format is consistent throughout the dataset as shown in *Figure 4.4*. Therefore, there is no additional work needed to standardize the date format.

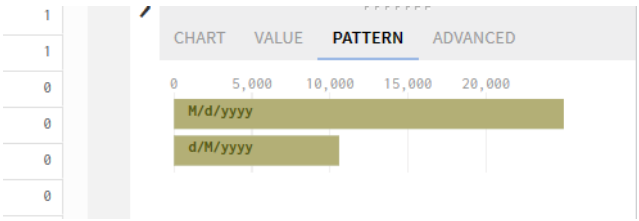

Figure 4.4 Check the date format for LastPurchaseDate attribute.

Since, I am not using Talend Data Preparation to handle the missing values, after conducting the necessary basic data cleaning, the dataset is then exported and ready to be imported to SAS Enterprise for handling missing values and specifying variable roles.

5 Data Import and Data Preprocessing using SAS Enterprise Miner

After the initial data quality check and basic cleaning, the dataset is now ready to be imported into SAS Enterprise Miner. However, prior to data import, a new Enterprise Miner Project was first created in the SAS Enterprise Miner tool. Then, a new diagram is created within it and nodes like “File Import”, “Metadata” and “Save Data” are added in *Figure 5.1*.

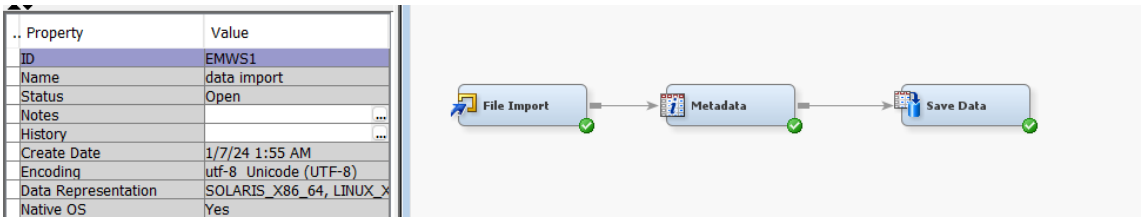

Figure 5.1 Data import in SAS Enterprise Miner.

After saving the imported data, a new library is created. Then, a new data source is created with reference to the library previously created as shown in *Figure 5.2*. Within the process of a creating a new data source, I specify the dataset variables roles during column metadata steps as shown in *Figure 5.3*.

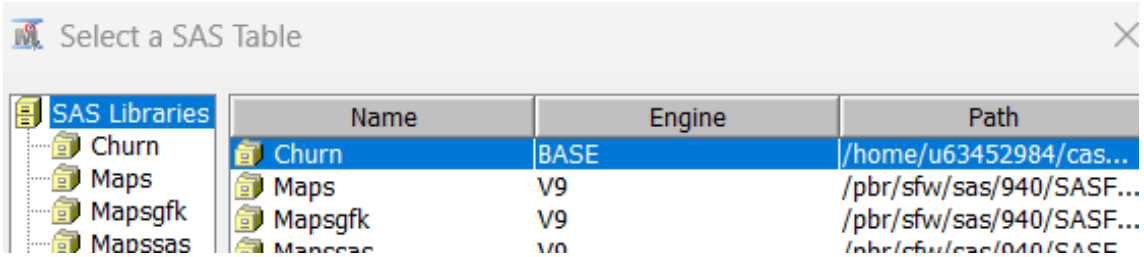

Figure 5.2 Creating new data source with reference to the created library – Churn.

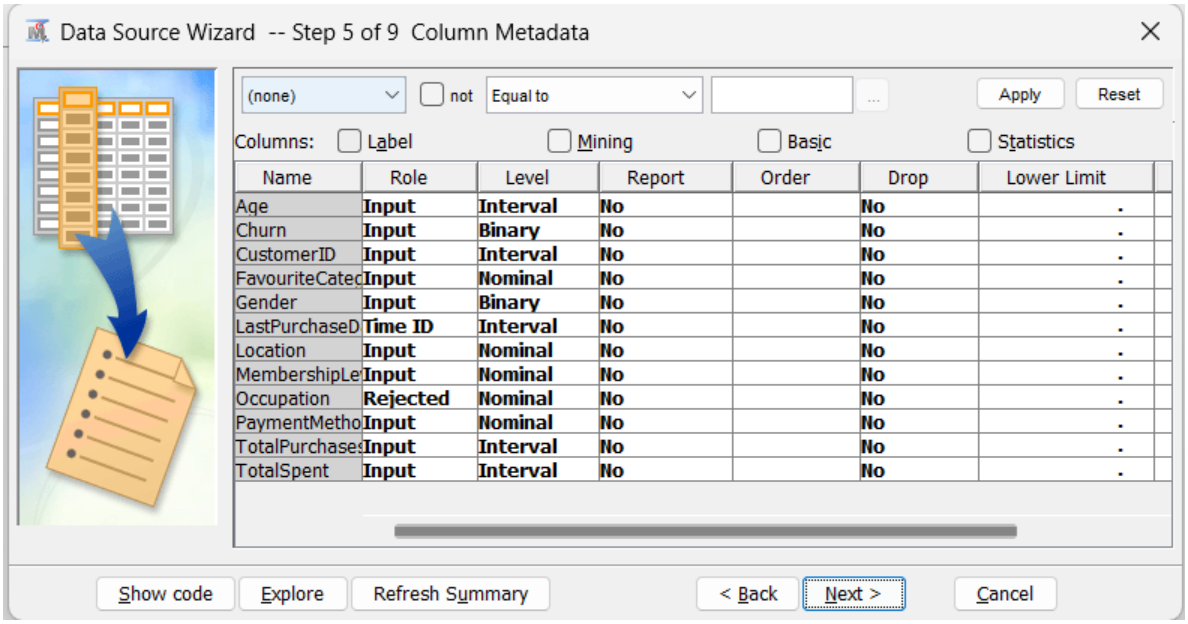

Figure 5.3 Specifying the variable roles during column metadata steps.

Referring to *Figure 5.3*, there are some minor errors in the auto-classified variables role and level by the SAS Enterprise Miner. Therefore, I performed a manual re-classification on the variables' role and level to make it suitable for this case study as shown in Table 5.1.

Variable	Auto Classification by SAS		Manual Re-classification	
	Role	Level	Role	Level
Age	Input	Interval	Input	Interval
Churn	Input	Binary	Target	Binary
CustomerID	Input	Interval	ID	Nominal
FavouriteCategory	Input	Nominal	Input	Nominal
Gender	Input	Binary	Input	Binary
LastPurchaseDate	Time ID	Interval	Time ID	Interval
Location	Input	Nominal	Input	Nominal
MembershipLevel	Input	Nominal	Input	Nominal
Occupation	Rejected	Nominal	Input	Nominal
PaymentMethod	Input	Nominal	Input	Nominal
TotalPurchase	Input	Interval	Input	Interval
TotalSpent	Input	Interval	Input	Interval

Table 5.1 Comparison between column metadata auto-classified by SAS and after manual re-classification.

With the data source ready, StatExplore is connected to the customer_churn data source to investigate the missing values found in the data source. To handle the missing values found in the data source, a built-in node called "Impute" is connected to the data source. The connections of the nodes are shown in *Figure 5.4*. From the previous StatExplore, missing values are found in two attributes, namely "TotalSpent" and "Churn". To deal with the missing values in the attribute "TotalSpent", the mean imputation is used as it helps to preserve the central tendency of that variable. For missing values in the attribute "Churn", the mode imputation is used as it is the common approach taken for binary class variables.

From the results of the"StatExplore_before" (*Figure 5.5*) and "StateExplore_after" (*Figure 5.6*), 26 missing values from the target variable – "Churn" and 8 missing values from the input variable – "TotalSpent" have been successfully replaced. Therefore, the data source is now ready for the modelling phase.

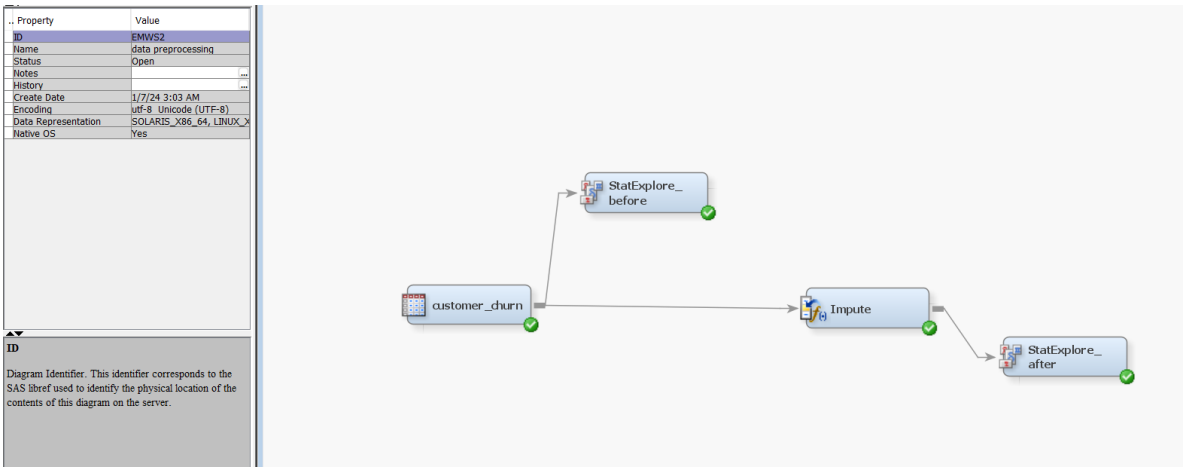

Figure 5.4 Connections of nodes for handling missing values in data source.

Distribution of Class Target and Segment Variables
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Level	Frequency Count	Percent
TRAIN	Churn	TARGET	0	19389	77.5591
TRAIN	Churn	TARGET	1	5584	22.3369
TRAIN	Churn	TARGET	.	26	0.1040

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness
Age	INPUT	44.95024	15.33948	24999	0	18	45	70	-0.07062
TotalPurchases	INPUT	5.341574	4.059632	24999	0	1	4	67	2.592697
TotalSpent	INPUT	4848.782	3750.709	24991	8	101	4132	62273	2.406783

Figure 5.5 Results of the StatExplore on data source BEFORE imputation.

Distribution of Class Target and Segment Variables
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Level	Frequency Count	Percent
TRAIN	IMP_Churn	TARGET	0	19410	77.6431
TRAIN	IMP_Churn	TARGET	1	5589	22.3569

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness
Age	INPUT	44.95024	15.33948	24999	0	18	45	70	-0.07062
IMP_TotalSpent	INPUT	4848.782	3750.108	24999	0	101	4133	62273	2.407169
TotalPurchases	INPUT	5.341574	4.059632	24999	0	1	4	67	2.592697

Figure 5.6 Results of the StatExplore on data source AFTER imputation.

6 Decision Tree Analysis using SAS Enterprise Miner

6.1 Variable Selections

Prior to performing decision tree analysis, it is important to perform variable selections to identify the suitable variables to be used for the decision tree analysis part. Therefore, I have used the "Variable Clustering" node in the SAS Enterprise Miner to determine the variable correlation as shown in *Figure 6.1*. The variable correlation plots are shown in *Figure 6.2*. By analyzing the results from the variable correlation plots, the suitable variables for predicting the target variable are – "TotalSpent", "TotalPurchases", "Age", "MembershipLevel", and "PaymentMethod"

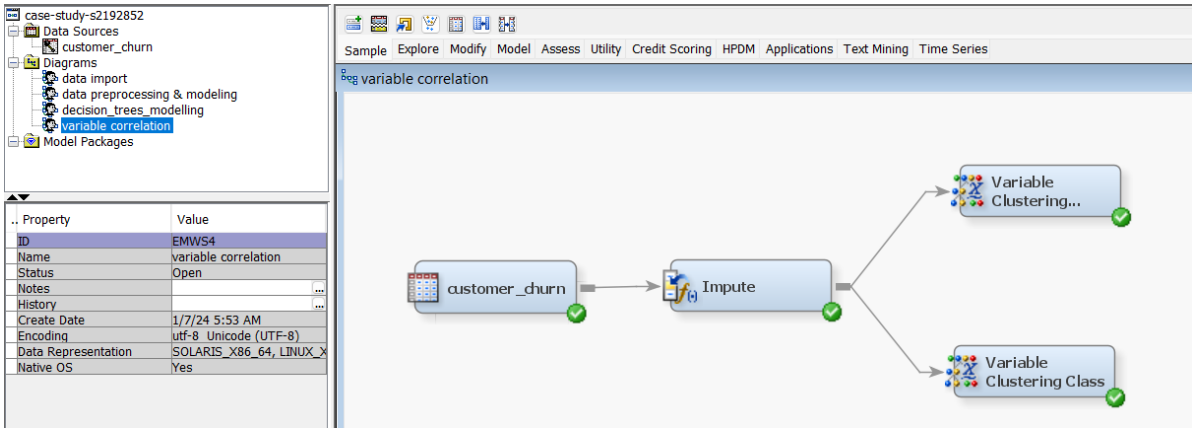

Figure 6.1 Nodes connections for variable clustering.

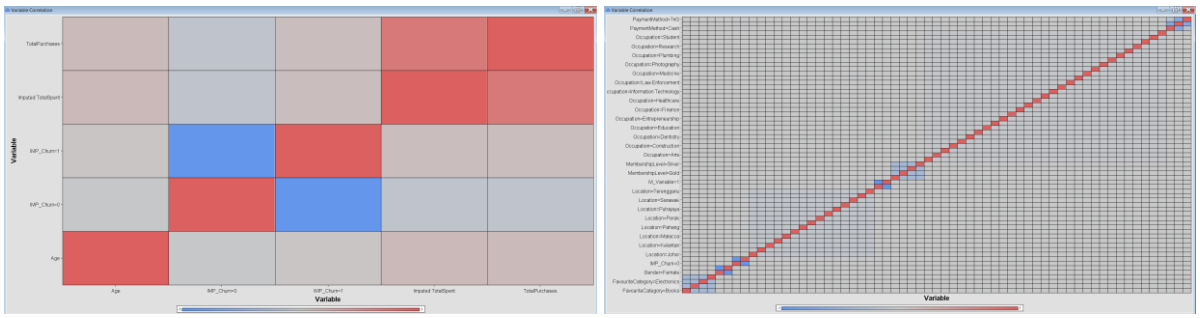

Figure 6.2 Variable correlations for interval variables (LEFT) and class variables (RIGHT) toward the targe variable "Churn".

6.2 Identifying Data Splitting Ratio

After identifying the variables, the next step will be identifying the data splitting ratio. I have decided to pre-run a decision tree modelling using three different data splitting ratio – 60:40 ,70:30 and 80:20 as shown in *Figure 6.3* and the results are shown in *Figure 6.4*. Based on the comparison of the results, there is no significant difference in terms of the model performance when tested on different data splitting ratios.

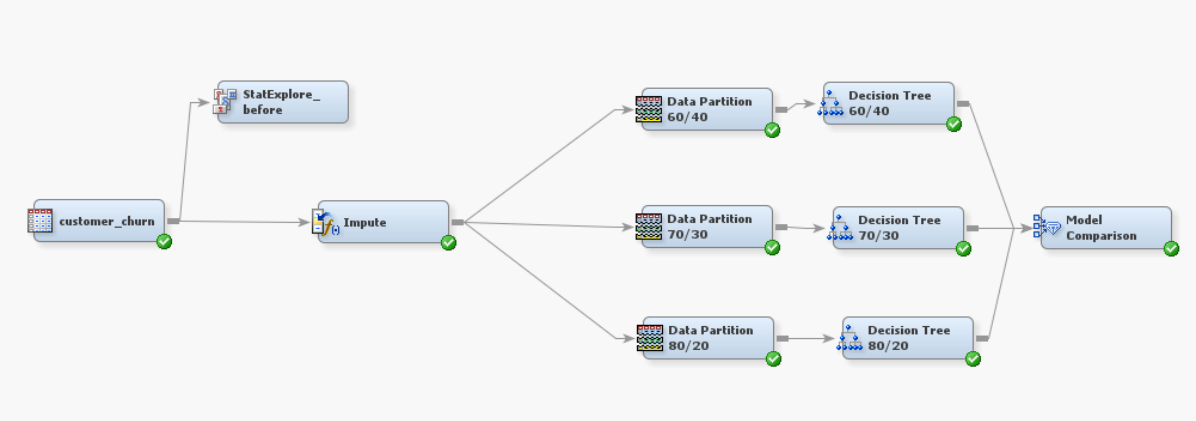

Figure 6.3 Decision Tree modelling using three different data splitting ratios.

Fit Statistics

Model Selection based on Train: Misclassification Rate (_MISC_)

Selected Model	Model Node	Model Description	Train: Misclassification Rate	Train: Average Squared Error	Train: Roc Index	Train: Gini Coefficient
Y	Tree2	Decision Tree	0.22170	0.17180	0.517	0.035
	Tree3	Decision Tree	0.22198	0.17188	0.517	0.034
	Tree	Decision Tree	0.22215	0.17164	0.518	0.036

Figure 6.4 Results of the comparison of model's performance.

6.3 Decision Tree Modelling Results

Since there is no significant difference in terms of the model performance when using different data splitting ratio, I've standardized the data splitting ratio using 70:30. The output of the decision tree is illustrated in *Figure 6.5*. Based on the decision tree diagram, the prediction of "Churn" attribute is mainly governed by the attrbutes "TotalPurchases" and "TotalSpent" after pruning unnecessary branches or attributes that do not provide significant value to the prediction.

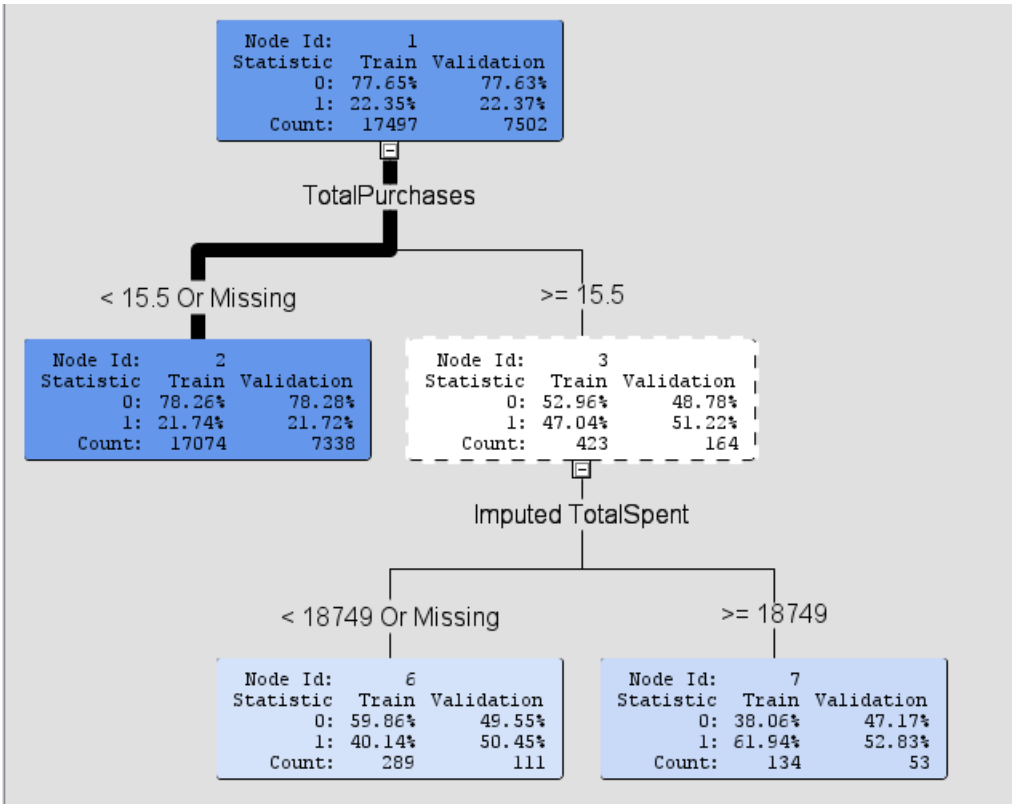

Figure 6.5 Decision Tree Diagram

In addition, the confusion matrix from the Decision Tree model with 70:30 data spitting ratio is shown in *Figure 6.6*. With the available confusion matrix, the precision, recall, F1-score, accuracy and specificity values are calculated and tabulated in *Table 6.1*. The formulas use to calculate those performance metrics are listed below:

- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- $F1 = \frac{2 \times precision \times recall}{precision + recall}$
- $Accuracy = \frac{TP+TN}{TP+FN+TN+FP}$
- $Specificity = \frac{TN}{TN+FP}$

Event Classification Table

Data Role=TRAIN Target=IMP_Churn Target Label=Imputed Churn

False Negative	True Negative	False Positive	True Positive
3828	13535	51	83

Data Role=VALIDATE Target=IMP_Churn Target Label=Imputed Churn

False Negative	True Negative	False Positive	True Positive
1650	5799	25	28

Figure 6.6 Confusion Matrix results for decision tree with data splitting ratio 70:30.

	Decision Tree	
	Train	Validate
Precision	0.619	0.528
Recall	0.021	0.017
F1-score	0.041	0.032
Accuracy	0.778	0.777
specificity	0.996	0.996

Table 6.1 The calculated results for precision, recall, f1-score, accuracy and specificity from the confusion matrix for decision tree with data splitting ratio 70:30.

6.4 Discussion on Decision Tree Model Performance

The results obtained from the decision tree model training and validation phases provide valuable insights into its performance. Precision, recall, F1-score, accuracy, and specificity are key metrics that offer a comprehensive evaluation of the model's ability to make accurate predictions.

Firstly, precision measures the accuracy of positive predictions made by the model. In training phase, the precision value ss 0.619. However, in validation phase, the precision value decreases to 0.528 when unseen data is fed into the model. This change could be caused by the data imbalance or overfitting during the training phase.

For recall, it helps to access the model’s ability to correctly identify positive instances among all the actual positives. The low recall values of 0.021 in training and 0.017 in validation point out that the decision tree model is struggling to capture a significant portion of positive instance due to overly high number of false negatives, which lead to positive cases are incorrectly classified as negative.

Next, the F1-score reflects the overall performance of the model. Based on the low F1-score recorded by both training and validation phase, it indicates there is still room for improvement by ensuring a better balance between precision and recall.

On the other hand, accuracy helps to measure the overall correctness of the model’s prediction. The model exhibits a relatively high accuracy of 0.778 in the training phase and 0.777 in the validation. However, the high accuracy rate of the decision tree model is misleading as the dataset used to train the model is highly imbalance, and the high accuracy rate is driven by the dominance of the majority class.

Lastly, for specificity, it gauges the model’s ability to correctly identify the negative instances. The high specificity values of 0.996 in both training and validation phases indicate that the

model has a strong ability to capture the negative instances. The high specificity values could also be an indicator that the negative is the majority class in the dataset.

In general, while the decision tree model showcases high accuracy and specificity, there is a notable need to focus on enhancing its capability to identify positive instances due to low precision and recall. This is due to the fact that the dataset used for training the model is highly imbalanced and resulted in overfitting of the majority class in the dataset. Therefore, it is essential to address the overfitting and data imbalances issues, so that the model's overall performance can be improved.

7 Ensemble Methods using SAS Enterprise Miner

7.1 Bagging and Boosting Model Results

In this section, I will be applying the "HP Forest" model as the bagging modelling technique and "Gradient Boosting" as the boosting modelling technique to predict customer churn. By employing the same selected variables and data splitting ratio, additional nodes like "Gradient Boosting" and "HP Forest" are added into the diagram and connected to the same Data Partition share by the Decision Tree 70:30 model as shown in *Figure 7.1*.

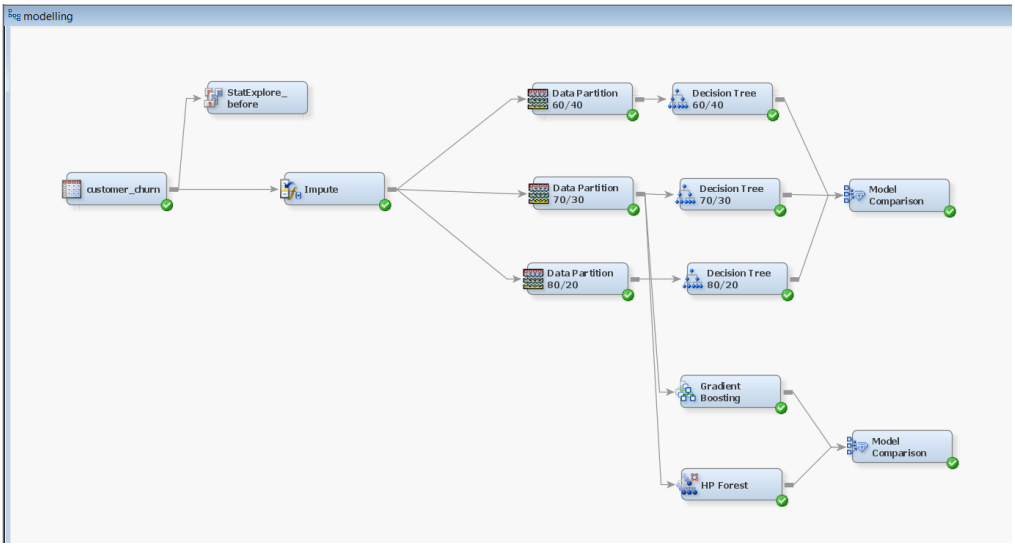

Figure 7.1 Added "Gradient Boosting" and "HP Forest" nodes to the modelling diagram.

The confusion matrix outputs from both Gradient Boosting and HP Forest model with 70:30 data spitting ratio is shown in *Figure 7.2*. With the available confusion matrix, the precision, recall, F1-score, accuracy and specificity values for Gradient Boosting and HP Forest model are calculated and tabulated in *Table 7.1*

Event Classification Table
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Model Node	Model Description	Data Role	Target	Target Label	False Negative	True Negative	False Positive	True Positive
Boost	Gradient Boosting	TRAIN	IMP_Churn	Imputed Churn	3808	13521	65	103
Boost	Gradient Boosting	VALIDATE	IMP_Churn	Imputed Churn	1645	5803	21	33
HPDMForest	HP Forest	TRAIN	IMP_Churn	Imputed Churn	3518	13578	8	393
HPDMForest	HP Forest	VALIDATE	IMP_Churn	Imputed Churn	1639	5781	43	39

Figure 7.2 Confusion matrix results for Gradient Boosting and HP Forest with data splitting ratio 70:30.

	Gradient Boosting		HP Forest	
	Train	Validate	Train	Validate
Precision	0.613	0.611	0.980	0.476
Recall	0.026	0.020	0.100	0.007
F1-score	0.051	0.038	0.182	0.013
Accuracy	0.779	0.778	0.798	0.224
specificity	0.995	0.996	0.999	0.974

Table 7.1 The calculated results for precision, recall, f1-score, accuracy and specificity from the confusion matrix for Gradient Boosting and HP Forest with data splitting ratio 70:30

7.2 Discussion on Bagging and Boosting Model Performance

The performance from both Gradient Boosting model and HP Forest model shows similar performance as the Decision Tree model, as summarized below: -

- Relatively low for precision, recall and F1-score.
- High accuracy and specificity.

The above descriptions have highlighted that both Gradient Boosting model and HP Forest model encounter issues in overfitting due to data imbalance. Therefore, it is essential to address the overfitting and data imbalances issues, so that both models' overall performance can be improved.

8 Solutions to Data Imbalances and Overfitting

To resolve the data imbalances and overfitting issues, there are two common approaches which are oversampling and undersampling.

Oversampling focuses on increasing the number of instances in the minority class through methods such as duplicating values or generating synthetic examples, like using the Synthetic Minority Over-sampling Technique (SMOTE). This process aims to create a more balanced class distribution between the minority and majority classes. Achieving balance in the dataset enables the model to effectively capture patterns within the minority class during training, thereby mitigating any bias towards the majority class. As a result, the overall performance of the model experiences improvement.

On the other hand, undersampling tackles data imbalance by reducing the number of instances in the majority class, typically done randomly to bring the distribution closer to that of the minority class. However, the use of undersampling comes with a trade-off, as there is a potential risk of discarding valuable information present in the majority class. Therefore, careful consideration of dataset characteristics becomes crucial when opting for undersampling.

Both oversampling and undersampling contribute to resolving the overfitting issue in a model by preventing the model from excessively memorizing the training data. By training the model on a balanced dataset, it allows for better generalization to unseen instances, resulting in improved prediction performance.

9 Suggestions for Business Strategy

In this case study for E-commerce customer behaviour analysis, customer churn is one of the important customer behaviours to be analysed. Customer churn predictions in e-commerce play a crucial role in guiding the business strategy by providing valuable insight that can retain their customer.

As previously mentioned in the context of the decision tree model, customer churn is influenced by both the total spent and total purchase amount by the customers through an e-commerce platform. There are several potential strategies which can be utilized by the e-commerce business owner to retain their customers such as: -

- Promotional Offers and Discounts:

E-commerce businesses can adjust the pricing and incentives to provide special offers or personalized discounts to customers to encourage their spending and purchasing through their e-commerce platform.

- Free Shipping Thresholds:

E-commerce businesses can set minimum order thresholds for free shipping. Customers may be inclined to add more items to their cart to meet the free shipping requirement, providing an extra incentive for larger purchases.

- Flash Sales and Limited-Time Offers:

E-commerce businesses can create a sense of urgency by hosting flash sales or limited time offers. This can trigger impulse purchases and encourage customers to buy more during the promotional period.

By combining these strategies, there is a higher possibility for the e-commerce business to retain their customers by encouraging their customers to make more purchases and overall spending.

10 Conclusion

In summary, this case study has successfully achieved the outlined objectives. By utilizing tools such as Talend Data Integration, Talend Data Preparation, and SAS Enterprise Miner, the dataset used underwent preprocessing to prepare it for modeling in SAS Enterprise Miner. Three distinct customer churn prediction models were employed – Decision Tree model, Bagging model (HP Forest), and Boosting model (Gradient Boosting), and the performance of each model is evaluated. However, the challenge of data imbalance within the dataset has led to compromises in the overall model performance for all three models. Therefore, in future work, it is recommended to implement data balancing techniques, such as oversampling or undersampling prior to model training. This step aims to rectify the data imbalance and improve the overall performance of the models. Finally, within this case study report, several potential business strategies have been proposed to aid in customer retention within the e-commerce domain.