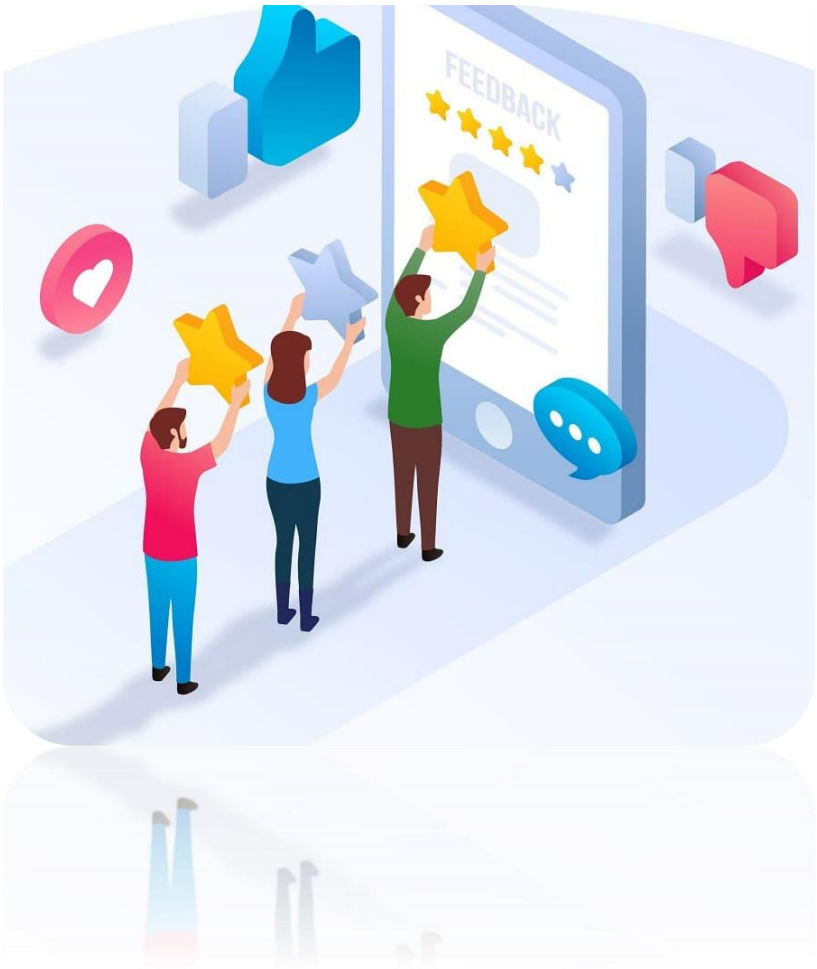

CMP7206:

Data Mining

TOPIC:

Customer Behavior Analysis

Name	ID
Thi Tinh Lo	22236226
Harriet Onoriode Otomiewor	23103939

Contents

List of Figures

List of Tables

1. Introduction

In the modern era, Data Mining has emerged as a prominent technological trend, and many organizations are seeking to maximize the insights they can obtain from their data. As a result, there has been a growing demand for individuals who can design and implement data mining solutions using various tools and methods.

The aim of this thesis is the application of data mining within the marketing. In the dynamic and ever-changing realm of marketing, comprehending customer behavior has assumed a position of paramount importance. The capacity to identify patterns, segment clientele, and forecast their reactions to marketing initiatives stands as a pivotal factor in devising personalized strategies that resonate and engage effectively.

This document delves into the realm of Customer Behavior Analysis, focusing on the identification of customer segments and the prediction of customer responses to sales calls through the utilization of techniques such as Dimensionality Reduction, Clustering, and Classification, encompassing both machine learning and deep learning algorithms. Moreover, it involves a comparative analysis and assessment of outcomes to determine the most optimal algorithms for addressing the challenges, thus contributing to informed decision making and refined marketing tactics.

The goal of this thesis is to provide a comprehensive understanding of data mining processing and their practical implementations. This is exemplified through the application of these methods to glean insights from a simulated dataset, illustrating their potential to extract valuable knowledge.

2. Domain Description

2.1. Domain Description

In contrast to earlier times, where the quality of products or services held significant important role in business for drawing and retaining customers, the contemporary landscape places paramount emphasis on delivering an unparalleled customer experience. A study conducted by SuperOffice in February 2023 underscored this shift, wherein 1920 business experts were surveyed to identify their foremost priority over the ensuing five years. The findings unequivocally revealed that customer experience emerged as the foremost concern, surpassing considerations related to products and pricing (Customer Service, 2023).

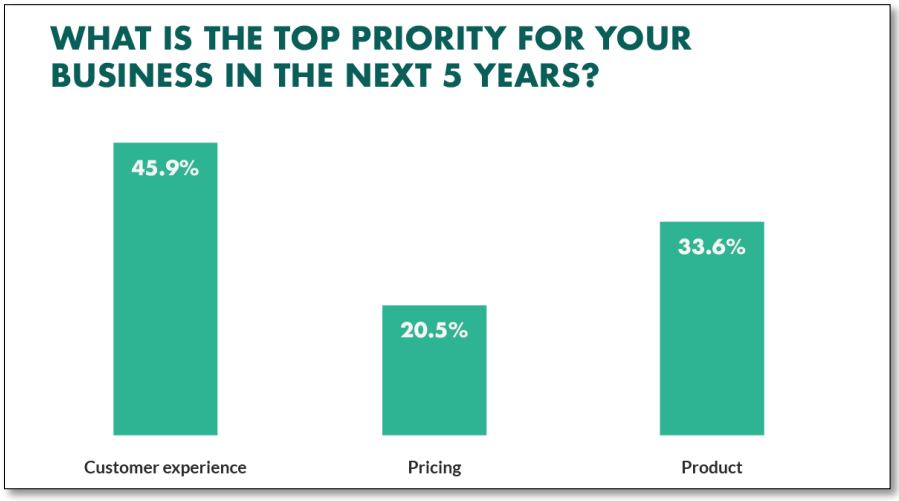

Figure 1: The Top Priority for The Business in The Next 5 years (Study by SuperOffice in February, 2023)

Furthermore, in the present age characterized by intense global competition and rapid technological advancements, enterprises engage in persistent exploration and examination of customer behavior to augment contentment, loyalty, retention rates, and avenues for market expansion. The analysis of customer behavior serves as a strategic tool enabling businesses to make well-grounded choices, fine-tune marketing approaches, elevate customer contentment, and enhance overall performance of the enterprise.

2.2. Problem Definition

In the landscape of today's business environment, customers exhibit a wide range of characteristics encompassing factors like education, age, occupation, income, gender, geographical location, and more. This contemporary era underscores individualism, wherein each customer responds differently to marketing strategy. It is crucial to identify and categorize customers into specific segments. By doing so, we can anticipate customer behavior and implement suitable marketing strategies for each group. Similarly, in juxtaposition to this scenario, when rendering services to customers and encountering instances of rejection or acceptance, there emerges an opportunity to pinpoint the specific customer segment and ascertain the underlying causes. This insight empowers businesses to adapt their approach towards the target customer, or alternatively, channel their efforts towards the primary customer base.

2.3. Literature Review

The objective of this endeavor is to delineate customer segments and forecast their responses to sales calls. To achieve this objective, the project employs clustering as its foundational technique, enabling the categorization of customers into distinct groups. Furthermore, the utilization of Classification methodologies is instrumental in predicting customer behavior, determining whether their response to offers will be affirmative ('yes') or negative ('no') based in individual customer profiles. Additionally, the adoption of Dimensionality Reduction techniques, exemplified by Principal Component Analysis (PCA), contributes to the simplification of intricate data structures. This reduction in the number of variables enhances the interpretability of the data and unveils latent patterns that are conducive to effective decision making processes.

Techniques	Algorithms	Author
Dimensionality Reduction	PCA	Brooks, 1988
Clustering	K-means	David Arthur and Sergei Vassilvitskii, 2007
	Hierarchical	Daniel A. McFadden, 2017
	DBSCAN	Martin Ester; Hans-Peter Kriegel; Jörg Sander; Xiaowei Xu, 1996
Classification	Logistic Regression	Paul D. Allison, 2019
	Decision Tree	Trevor Hastie; Robert Tibshirani; Jerome Friedman, 2009
	Random Forest	Breiman, 2001
	Neutral Network	Nielsen, 2015

Table 1: Data Mining Techniques Used

2.4. Dataset Description

- **Dataset:**

The dataset of vehicle insurance customers from Kaggle created by IBM Watson Analytics, is named "IBM Watson Marketing Customer Value Data" (Analytics, 2019). This dataset includes 9,134 instances with 24 input variables related to customers details such as income, location, insurance plan, vehicle size, and so on.

No.	Variables	Type	Description
1	Customer	character	ID for each customer
2	State	character	The state customer resides
3	Customer_Lifetime_Value	numeric	Quantify the potential revenue that customer will bring to the company
4	Response	character	Whether the customer response to campaign
5	Coverage	character	The type of insurance coverage
6	Education	character	The level of education
7	Effective_To_Date	character	The date when the data was collected
8	EmploymentStatus	character	The employment status
9	Gender	character	The gender of the customer
10	Income	integer	The income of the customer
11	Location_Code	character	The code of the location of the customer
12	Marital_Status	character	The marital status
13	Monthly_Premium_Auto	integer	The amount the customer pays as a monthly premium for auto insurance
14	Months_Since_Last_Claim	integer	The number of months since the last insurance claim
15	Months_Since_Policy_Inception	integer	The number of months since the inception of the insurance policy
16	Number_of_Open_Complaints	integer	The number of open complains the customer has
17	Number_of_Policies	integer	The number of insurance policies the customer holds
18	Policy_Type	character	The type of insurance policy
19	Policy	character	Specific group of policy

20	Renew_Offer_Type	character	The type of renewal offer presented to the customer
21	Sales_Channel	character	The channel which the customer was acquired
22	Total_Claim_Amount	numeric	The total amount claimed by the customer
23	Vehicle_Class	character	The class of vehicle
24	Vehicle_Size	character	The size category of vehicle

Table 2: Detailed Variables in the Dataset

- **Data Mining Task Procedure:**

To achieve the goal, the procedure for Data Mining involves several steps. After acquiring the data from Kaggle, the preprocessing phase encompasses tasks like detecting and handling missing values, outliers, scaling, and performing data transformations. Once the data is prepared, the subsequent step is applying Data Mining, selecting suitable techniques and algorithms. Lastly, the outcomes are evaluated, leading to conclusions and solutions to improve performance.

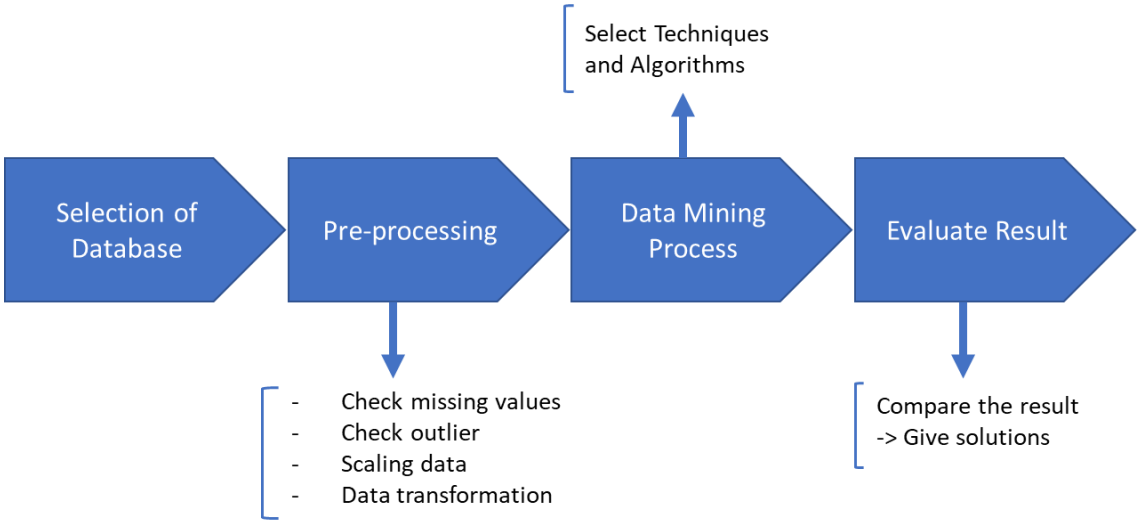

Figure 2: Data Mining Task Procedure

3. Dataset pre-processing

"How can the data be preprocessed in order to help improve the quality of the data and, consequently, of the mining results?"

In the real world, databases are extremely vulnerable to noisy, missing, and inconsistent data due to their typically huge size and their derivation from diverse and dissimilar sources. Low-quality data will lead to low-quality mining results (Han, Jiawei; Kamber, Micheline; Pei, Jian, 2012). Therefore, pre-processing is fundamentally the procedure that transform data into a format easily and effectively processed in data mining.

3.1. Missing Values

To begin with, conduct a thorough examination of missing values to sufficiently guarantee the quality of the data. There is no missing value in the dataset.

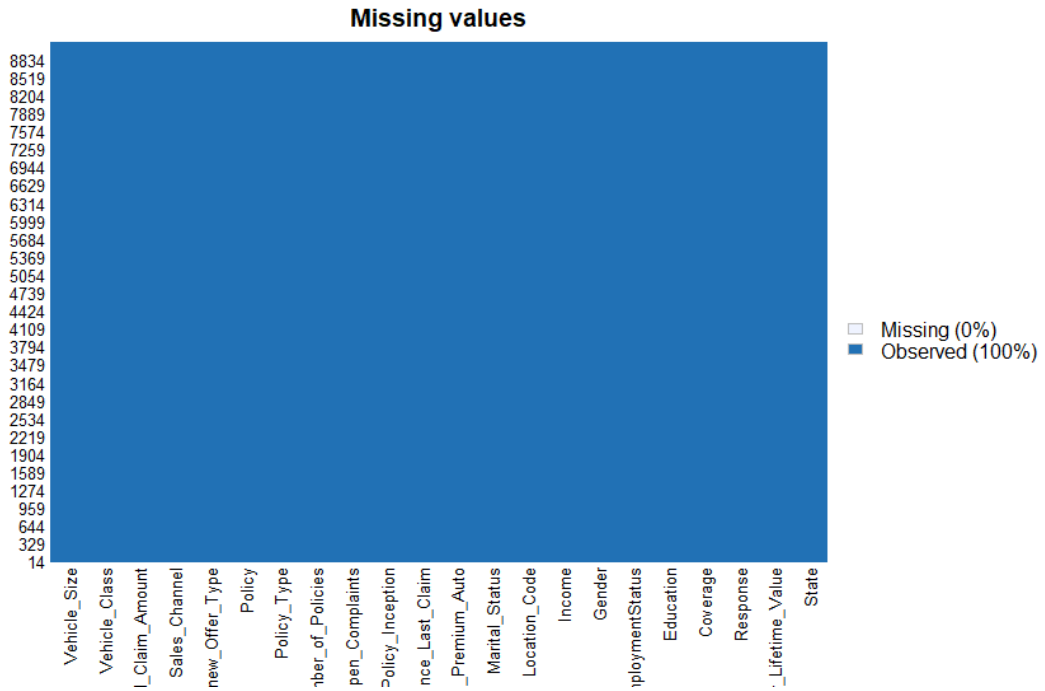

Figure 3: Check Missing Values

3.2. Outliers

Outliers can impact the reliability and significance of results. In this dataset, apply a procedure to identify outliers within numerical variables and eliminate exceptional data points. This will ensure that the dataset retains substantial and meaningful values.

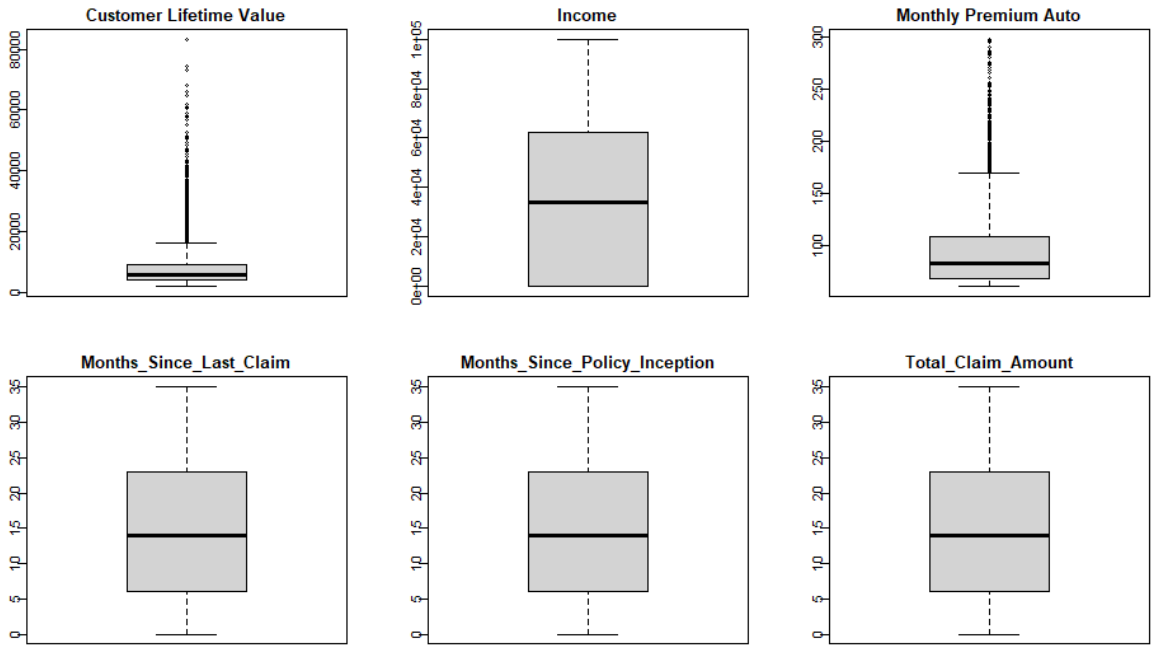

Figure 4: Boxplots for Outlier Detection

Remove records from the dataset where the values in the "Customer_Lifetime_Value" exceed 52,000.

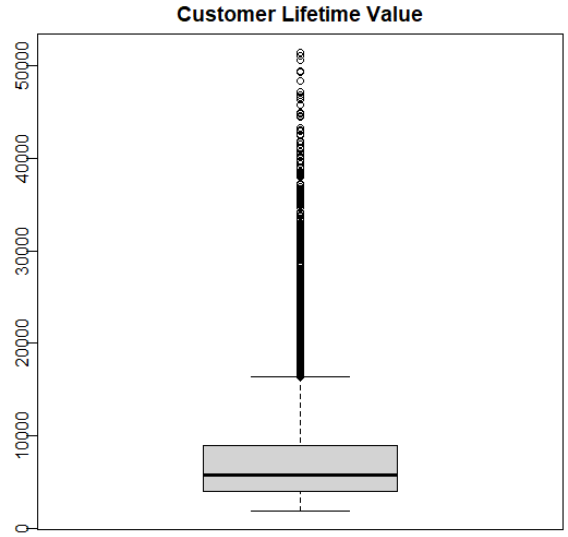

Figure 5: Boxplot for "Customer Lifetime Value" after Removing Outliers

3.3. Data Transformation

Data transformation involves changing data from its original structure or format, to the specific arrangement needed for a target system or purpose (Gareth James, 2017).

In this project, implement the conversion of categorical variables into numerical values as follows:

+ For “Response” variable: set “No” is 0 and “Yes” is 1.

+ For Nominal attributes: Use label encoding without assigning specific labels to the values.

+ For Ordinal attributes: Assign labels to the values based on their respective order.

3.4. Feature Scaling

Due to the significant variability in the range of raw data values, certain machine learning algorithms may not function correctly without normalization. As a result, it's essential to normalize the range of all features (Andrew, 2017).

Utilize the scale () function in R for the numerical columns which implement Standardization (Z-score Normalization) method (Use the scale(), 2021).

3.5. Pre-processed Dataset

Dataset before pre-processing: WA_Fn-UseC_-Marketing-Customer-Value-Analysis.csv

https://github.com/ThyTina/Data_Mining/blob/main/WA_Fn-UseC_-Marketing-Customer-Value-Analysis.csv

Dataset after pre-processing: data_processed.csv

https://github.com/ThyTina/Data_Mining/blob/main/data_processed.csv

4. Dimensionality Reduction

Dimensionality reduction involves converting data from a space with many dimensions to a space with fewer dimensions. The objective is for the reduced representation to preserve important

characteristics of the original data, ideally capturing its essential features while approaching its inherent dimension (Hinton, 2008).

Principal Component Analysis (PCA) is a statistical method used to decrease the complexity of a dataset by transforming the data into a new coordinate system. In this new space, a significant portion of the data's variability can be captured using fewer dimensions than the original dataset (Brooks, 1988). PCA was applied to reduce the dimensions of a dataset into two components, effectively reducing its dimensions. The scatter plot visually illustrates the distribution and correlations between these two crucial dimensions, enabling the identification of patterns, clusters, and the extraction of insights regarding the data's underlying structure.

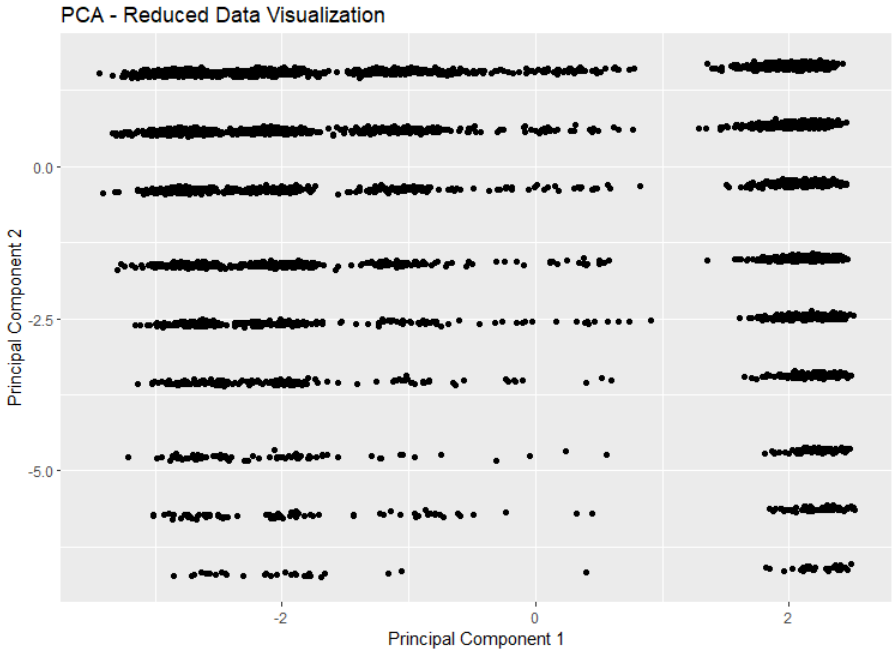

Figure 6: PCA for Reducing Data Features to Two Components

5. Clustering Algorithms

Clustering is the process of grouping a set of data objects into multiple groups or clusters so that objects within a cluster have high similarity but are very dissimilar to objects in other clusters. Unlike in classification, the class label of each customer is unknown (Anil K. Jain, Richard C. Dubes, 1988).

In this document, following the application of Principal Component Analysis (PCA) to reduce dimensionality to two components, the process involves employing clustering techniques such as K-means, Hierarchical, and DBScan for customer segmentation.

To assess the performance of the clustering model, evaluation metrics including Silhouette score and Within-cluster sum of squares are employed.

- **The silhouette score:**

The silhouette value is a measure of how similar an object is to its own cluster compared to other clusters (Peter J. Rousseeuw, 1987).

The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

- **Within-cluster sum of squares (WCSS):**

Within-Cluster Sum of Squares (WCSS) measures the dispersion of data points within individual clusters. Typically, a cluster with a lower sum of squares is denser, while a higher sum of squares indicates greater spread and variability among the observations within the cluster (Trevor Hastie, Robert Tibshirani, Jerome Friedman, 2009).

5.1. PCA and K-means

K-Means Clustering is an unsupervised learning algorithm that aims to find k cluster centers and assign objects to the nearest cluster center, minimizing the squared distances from the clusters. The Elbow Method is commonly used approaches to determine the optimal number of clusters for a given dataset (David Arthur and Sergei Vassilvitskii, 2007).

The Elbow Method is a technique that involves calculating the Within-Cluster-Sum of Squared Errors (WSS) for various values of k and selecting the value of k where the WSS starts to diminish. This can be visually observed in the plot of WSS-versus-k, where a distinct elbow point indicates the optimal number of clusters (Tim K. Marks; Thomas J. Cova, 2011).

- **Find the Optimal Number of Clusters:**

Apply the K-means algorithm with varying cluster numbers ranging from 1 to 10. Calculate the Within-Cluster Sum of Squares (WSS) for each value of k as follows:

```
Number of clusters (k): 1       WSS: 76211.11
Number of clusters (k): 2       WSS: 34241.06
Number of clusters (k): 3       WSS: 19538.75
Number of clusters (k): 4       WSS: 10697.5
Number of clusters (k): 5       WSS: 8783.394
Number of clusters (k): 6       WSS: 6618.794
Number of clusters (k): 7       WSS: 5440.401
Number of clusters (k): 8       WSS: 4529.606
Number of clusters (k): 9       WSS: 3525.282
Number of clusters (k): 10      WSS: 3050.173
```

Table 3: WSS for Different Number of Clusters (k) from 1 to 10 in K-means

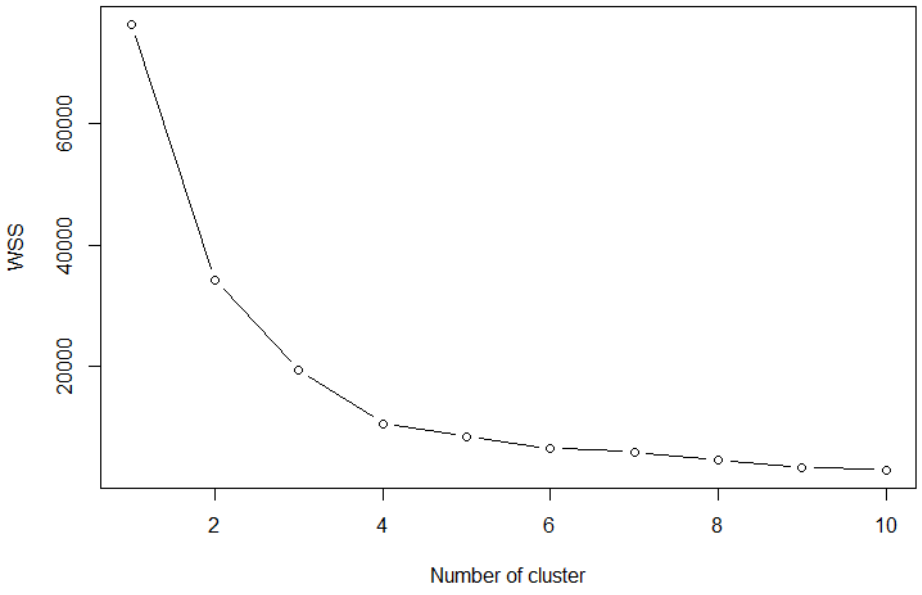

Figure 7: WSS for Different Number of Clusters (k) from 1 to 10 in K-means

- **Build Model:**

Creating the model using k=4. The outcome illustrates the distribution of data points among the four clusters.

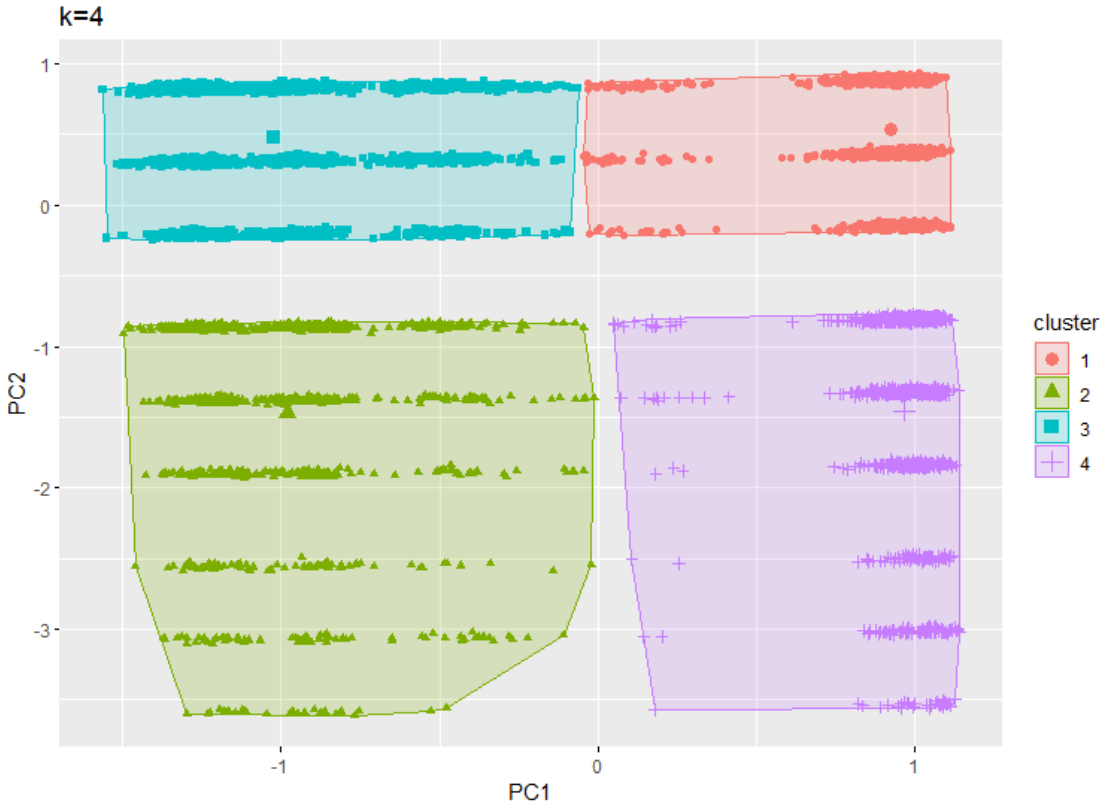

Figure 8: K-means Clusters

- **Clustering Statistics:**

Employing the "factoextra" library in R to compute and extract cluster statistics from the outcomes of k-means clustering.

No.	Statistics	Values
1	Within-Cluster Sum of Squares (WCSS)	10697.5
2	Silhouette Score	0.6406251

Table 4: WCSS and Silhouette Score for K-means

The WCSS value of 10697.5 suggests that the data points within each cluster are relatively close to their cluster centers. Lower WCSS values often indicate well-separated clusters.

The Silhouette Score of 0.6406251 indicates that the clusters are relatively well-defined and appropriately separated.

5.2. PCA and Hierarchical

Like K-means, Hierarchical clustering also groups similar data points together (Daniel A. McFadden, 2017). There are two main types of Hierarchical clustering: divisive (top-down) and agglomerative (bottom-up).

- **Build Model:**

Apply Agglomerative Hierarchical Clustering using specific parameters:

No.	Parameters	Values
1	Cluster Distance Measure	Euclidean
2	Linkage method	Centroid
3	Number of clusters	4

Table 5: Hierarchical Clustering Model Parameters

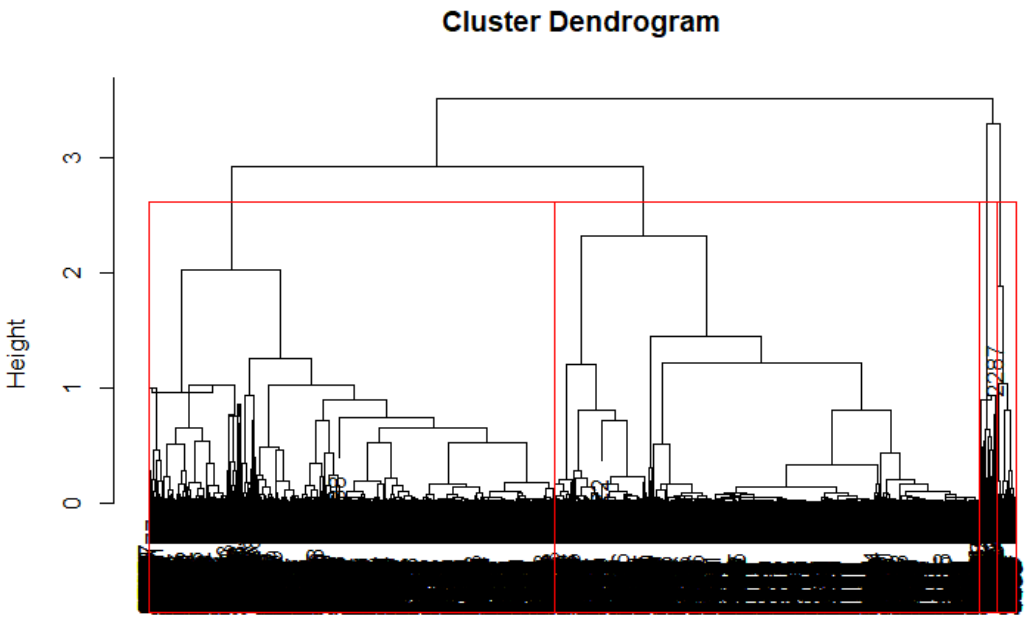

Figure 9: Cluster Dendrogram for Hierarchical Clustering Model

- **Clustering Statistics:**

No.	Statistics	Values
1	Within-Cluster Sum of Squares (WCSS)	22663.27
2	Silhouette Score	0.5491875

Table 6: WCSS and Silhouette Score for Hierarchical

The WCSS value of 22663.27 suggests that the clusters are relatively compact and data points are closer to their cluster centers.

The Silhouette Score of 0.5491875 suggests that the clusters have moderate separation and cohesion.

5.3. PCA and DBScan

DBScan is a non-parametric density-based clustering algorithm. It identifies groups of points that are densely concentrated, considering points with numerous close neighbors. Additionally, it designates isolated points in low-density regions, those situated far from their nearest neighbors, as outliers (Martin Ester; Hans-Peter Kriegel; Jörg Sander; Xiaowei Xu, 1996).

- **Build Model:**

Implement DBScan using specific parameters:

No.	Parameters	Values
1	Epsilon (maximum distance between two points)	0.5
2	Minimum number of points	4

Table 7: DBScan Clustering Model Parameters

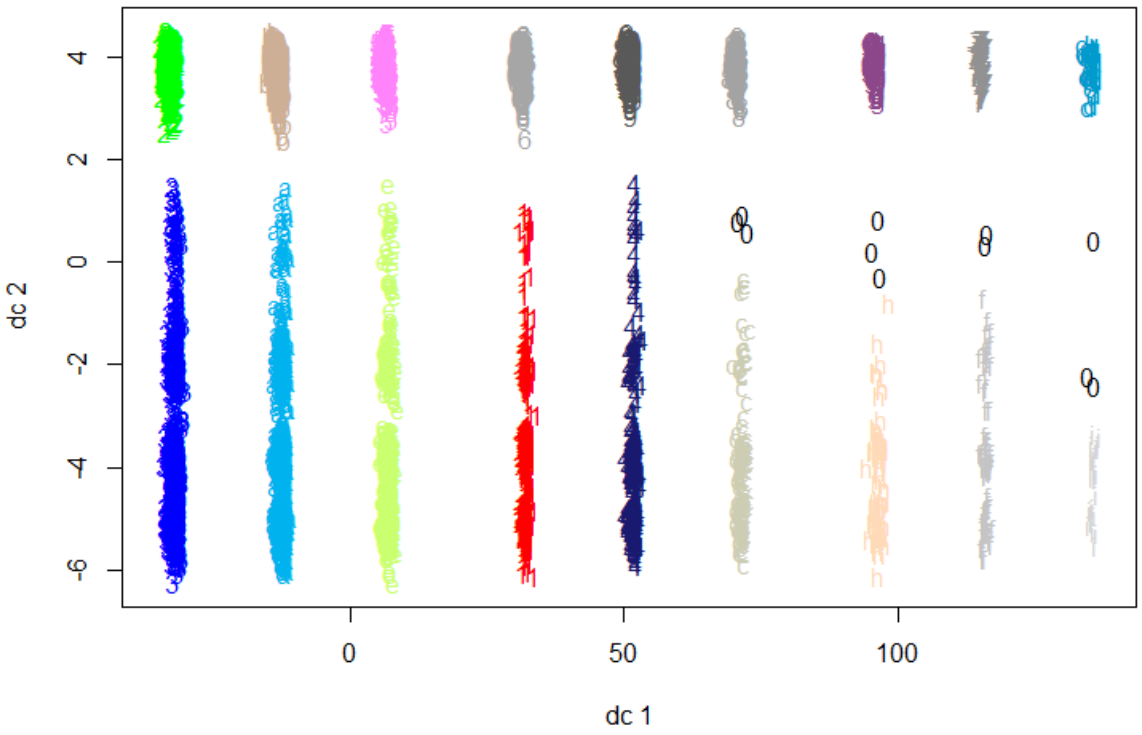

Figure 10: DBScan Clusters

- **Clustering Statistics:**

No.	Statistics	Values
1	Within-Cluster Sum of Squares (WCSS)	2734.483
2	Silhouette Score	0.6162586

Table 8: WCSS and Silhouette Score for DBScan

WCSS is commonly associated with K-means clustering, where it measures the compactness of clusters. WCSS might not be the most appropriate metric for evaluating DBScan results.

The Silhouette Score of 0.6162586 indicates that the DBScan model has generated clusters with a reasonable degree of separation and cohesion.

5.4. Compare Clustering Algorithms

No.	Algorithm	Silhouette Score
1	PCA & K-means	**0.6406251**

2	PCA & Hierarchical	0.5491875
3	PCA & DBScan	0.6162586

Table 9: Comparing Silhouette Score of K-means, Hierarchical and DBScan

Generally, the K-means clustering model applied to PCA-transformed data has achieved the highest Silhouette Score (0.64), indicating strong cluster separation and cohesion. DBScan clustering model is slightly lower than K-means but still indicative of meaningful clusters with Silhouette Score at 0.62. The hierarchical model yields lower effectiveness score than the others.

6. Classification Algorithms

Classification is a process related to categorization, the process in which ideas and objects are recognized, differentiated, and understood. Classification is the grouping of related facts into classes. It may also refer to a process which brings together like things and separates unlike things (Trevor Hastie; Robert Tibshirani; Jerome Friedman, 2009).

Within this thesis, we employ classification algorithms such as logistic regression, decision tree, random forest, and neural network to anticipate customer responses.

For assessing the classification model's performance, the utilization of ROC curve, confusion matrix, accuracy, precision, recall, and F1 score is employed.

- ROC Curve shows how well a binary classifier system can distinguish between classes as the discrimination threshold is adjusted. This curve depicts the relationship between the true positive rate (TPR) and the false positive rate (FPR) across different threshold values (Fawcett, T., 2004).
- Confusion Matrix is a matrix utilized to evaluating the classification model performs. The matrix contrasts the real target values with the predictions made by the machine learning model (Daniel P. Russo, 2019). The Confusion Matrix is utilized to compute metrics that assess the model's performance, such as Accuracy, Precision, Recall, and F1 Score.

		Predicted	
	Total population = P + N	**Negative (PN)**	**Positive (PP)**
Actual	**Negative (N)**	True negative (TN)	False positive (FP)
	Positive (P)	False negative (FN)	True positive (TP)

Table 10: Confusion Matrix

- Accuracy score is a common evaluation metric used to assess the performance of a classification model. It measures the proportion of correctly classified instances out of the total number of instances in the dataset. The formula is:

$$\text{Accuracy} = \frac{TN+TP}{TN+TP+FN+FP}$$

- Precision measures the accuracy of positive predictions by measuring the proportion of correctly identified true positives. The formula is:

$$\text{Precision} = \frac{TP}{TP+FP}$$

- Recall assesses the classifier's ability to correctly predict positive cases among all the positive instances present in the dataset. The formula is:

$$\text{Accuracy} = \frac{TP}{TP+FN}$$

- F1-Score quantifies combine both precision and recall. It calculates an average of values, precision and recall. F1-Score ranges from 0 to 1, a higher F1-Score indicates a better trade-off between precision and recall. The formula is:

$$\text{F1 Score} = 2\,x\,\frac{\text{Precision x Recall}}{\text{Precision+ Recall}}$$

Data Splitting:

Divide the training and testing data for classification algorithms: using “sample” function in R to generate a random 70% for training data and 30% for testing data to apply classification algorithms. “Response” is binary outcome variable that try to predict. All other variables in the dataset are being used as predictors for the “Response” variable.

6.1. Logistic Regression

Logistic regression is the form of regression analysis when dealing with a binary dependent variable. It is employed to characterize data and elucidate the connection between a single binary dependent variable and multiple independent variables that could be nominal, ordinal, interval, or ratio-level (Paul D. Allison, 2019).

Building a logistic regression model using the Generalized Linear Model (glm) framework.

These are the estimated coefficients for each predictor variable in the model. indicating the significance of each predictor's effect on the response: Vehicle_Size, Total_Claim_Amount, Sales_Channel, Marital_Status, Renew_Offer_Type are very significant effect on the response.

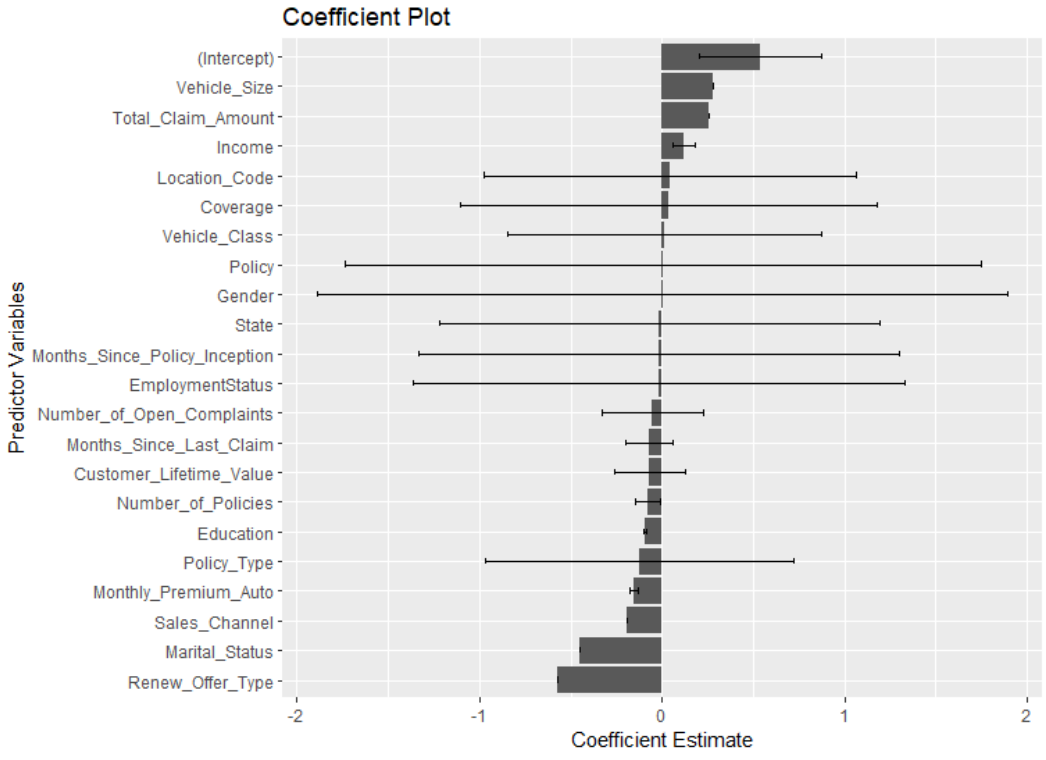

Figure 11: Coefficient Estimate by Logistic Regression Model

Evaluate the Logistic Regression model:

- ROC Curve:

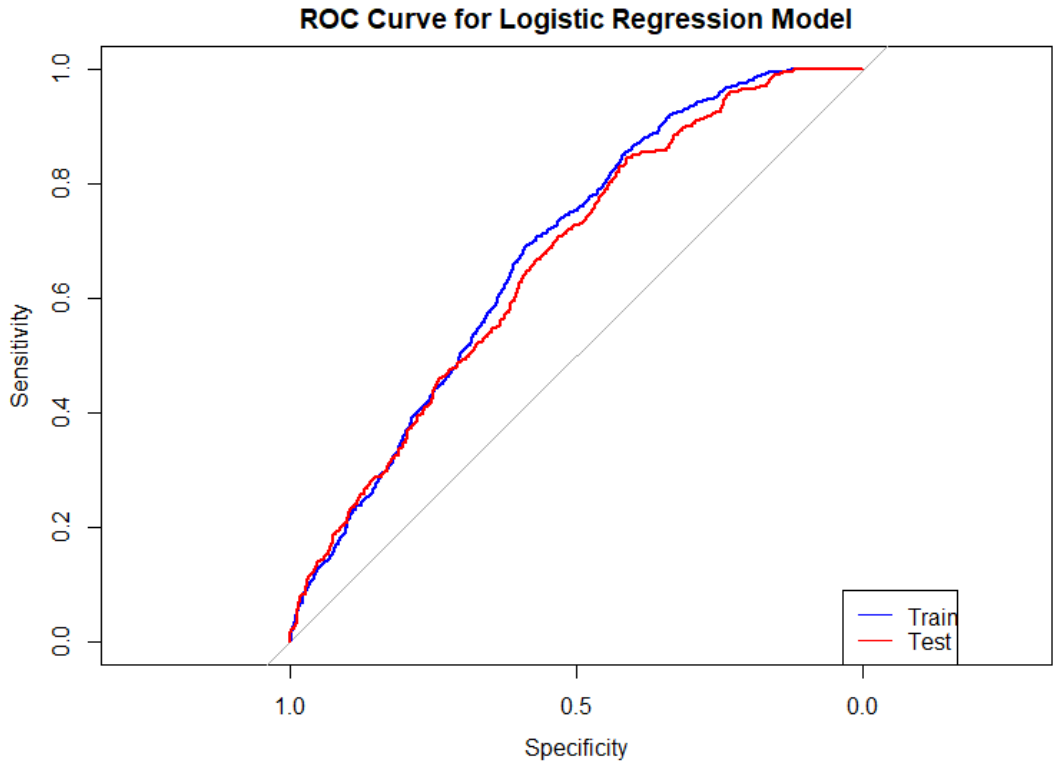

Figure 12: ROC Curve for Logistic Regression Model

- Confusion matrix:

		Train_Predicted		Test_Predicted	
	n= 6394	**No**	**Yes**	**No**	**Yes**
Actual	**No**	TN= 5460	FP= 932	TN= 2363	FP= 376
	Yes	FN= 2	TP= 0	FN= 1	TP= 0

Table 11: Confusion Matrix for Logistic Regression Model

- Classification Statistics:

No.	Statistics	Train Data	Test Data
1	Accuracy	0.8539256	0.8624088
2	Precision	0	0
3	Recall	0	0
4	F1 Score	NaN	NaN

Table 12: Accuracy, Precision, Recall, F1-score for Logistic Regression Model

TP value is 0, it means that the model did not correctly predict any positive instances. Both precision and recall will also be 0, and the F1 score cannot be calculated.

In this case, even though the accuracy is high for both train and test data, but the model is not effectively identify positive instances, this is possibly because of the potential class imbalance, or data quality, feature selection. Logistic Regression model is not suitable algorithm for imbalance dataset.

6.2. Decision Tree

Decision tree learning is a type of supervised learning that aims to construct a predictive model by considering multiple input variables to predict the value of a target variable. The construction of a decision tree involves dividing the initial dataset, starting with the root node, into subsets that become the child nodes. These divisions are determined using specific rules based on classification features (Trevor Hastie; Robert Tibshirani; Jerome Friedman, 2009).

Build Decision Tree model use “rpart” function in R with method = "class" specifies that the decision tree is being built for a classification task. The chart illustrates the division of data at each node, influenced by various variables and conditions. There are a total of four nodes in the model.

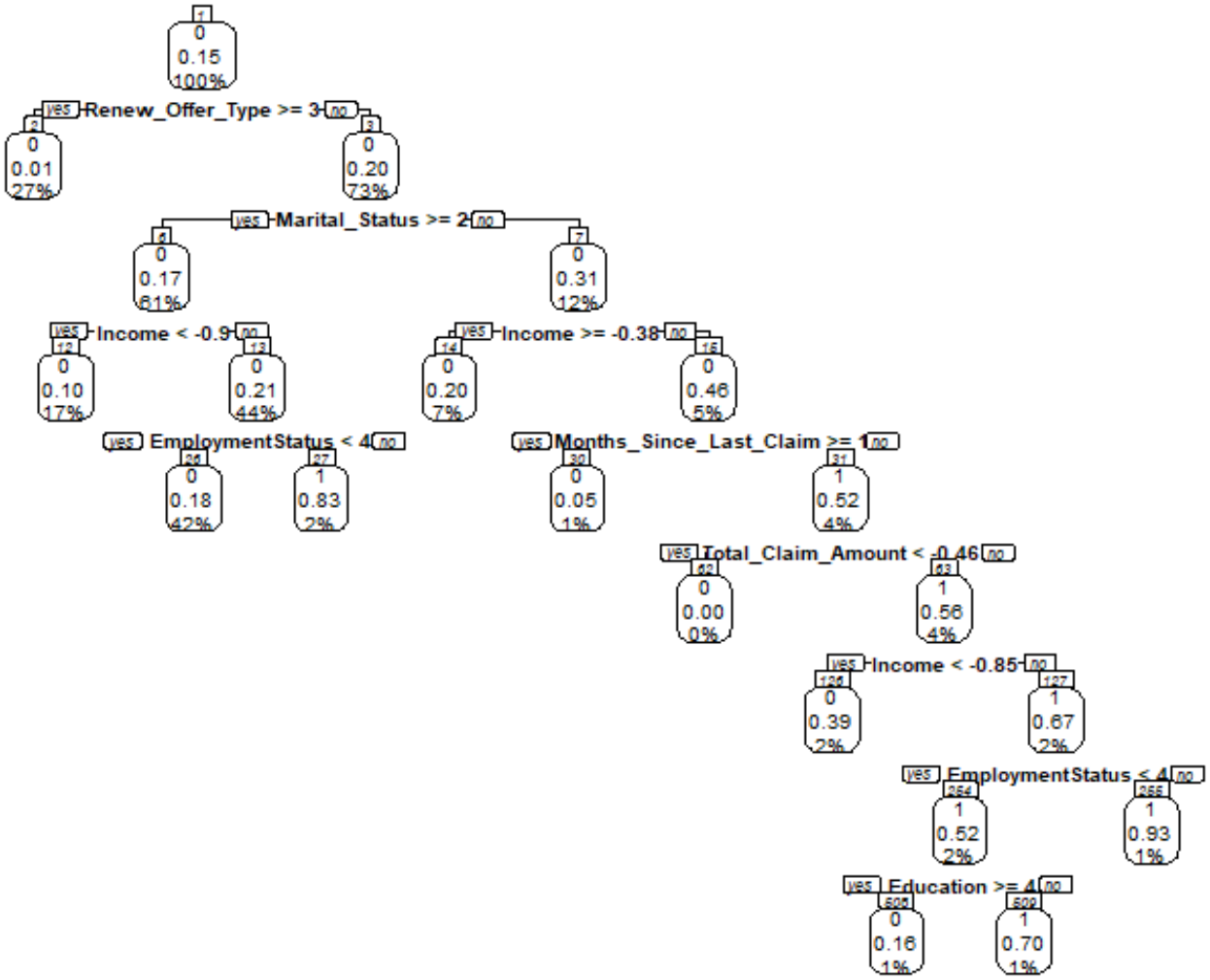

Figure 13: Plot Decision Tree Model

The variables used for splitting:

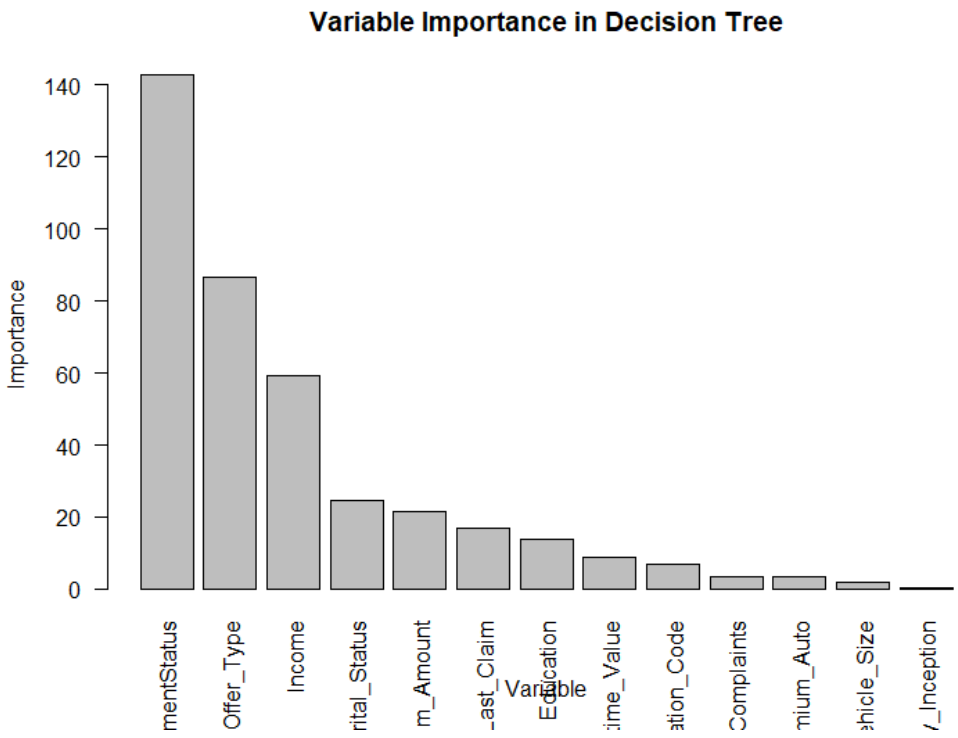

Figure 14: Variable Importance in Decision Tree

Evaluate the Decision Tree model:

- ROC Curve:

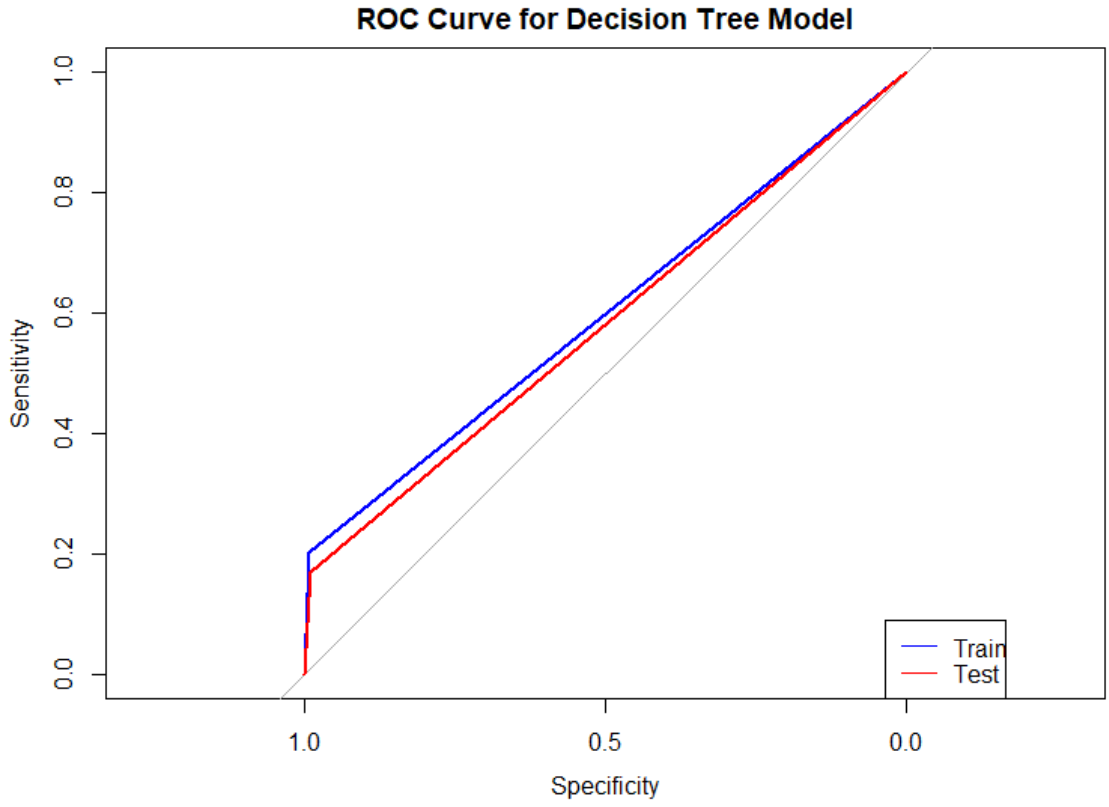

Figure 15: ROC Curve for Decision Tree Model

- Confusion matrix:

		Train_Predicted		Test_Predicted	
	n= 6394	**No**	**Yes**	**No**	**Yes**
Actual	**No**	TN= 5420	FP= 744	TN= 2340	FP= 312
	Yes	FN= 42	TP= 188	FN= 24	TP= 64

Table 13: Confusion Matrix for Decision Tree Model

- Classification Statistics:

No.	Statistics	Train Data	Test Data
1	Accuracy	0.8770723	0.8773723
2	Precision	0.2017167	0.1702128
3	Recall	0.8173913	0.7272727
4	F1 Score	0.32358	0.2758621

Table 14: Accuracy, Precision, Recall, F1-score for Decision Tree Model

The model's accuracy is approximately 87.7% for both the train and test datasets. This suggests that the model is performing fairly well in correctly classifying instances overall.

The precision values are approximately 0.2 are low for both train and test datasets. This indicates that when the model predicts a positive instance, it is often incorrect, leading to a relatively high number of false positives.

The recall values are approximately 0.8 are moderate to high, indicating that the model is better at capturing a significant portion of the actual positive instances, especially in the train dataset.

The F1 Score values are approximately 0.3 indicates that the model's performance in terms of both precision and recall is not optimal.

6.3. Random Forest

Random forests are an ensemble learning method that uses multiple decision trees for tasks like classification and regression. It constructs a collection of decision trees during training, combining their results to make predictions. By averaging the outcomes of many trees trained on different parts of the same data, random forests aim to decrease variance and enhance accuracy. To improve performance, multiple trees are utilized, each considering a random subset of features for splitting (Breiman, 2001).

Build Random forest model using parameters:

No.	Parameters	Values
1	ntree (number of trees in the forest ensemble)	15
2	mtree (Number of Variables Considered at Each Split)	4

Table 15: Random Forest Parameters

The plot showcasing the importance of each variable in your random forest model's decision-making process.

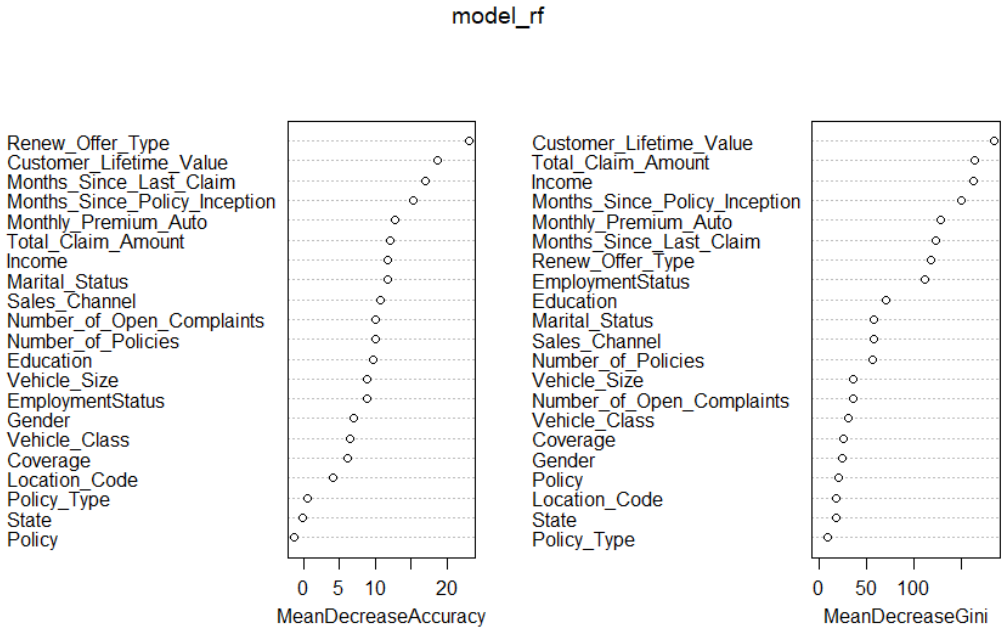

Figure 16: The Importance Scores of Each Predictor Variable in Random Forest

The tree shows each parent node leads to two child nodes (left and right daughters), representing the different paths that data instances can take based on the decision criteria. The tree structure is recursive, with parent nodes leading to further splits and child nodes until reaching the terminal nodes (leaf nodes) where predictions are made.

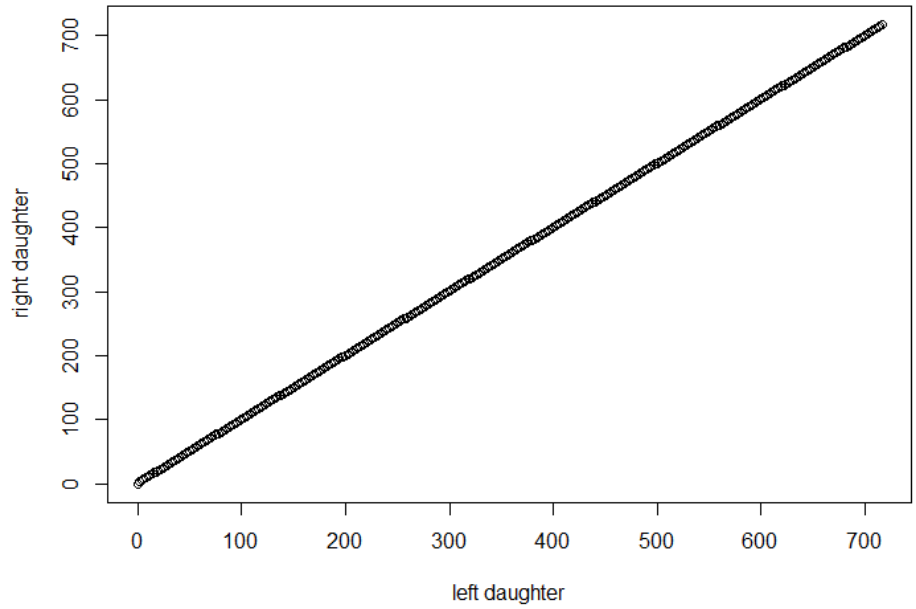

Figure 17: Binary Decision Tree in Random Forest

Evaluate the Random Forest model:

- ROC Curve:

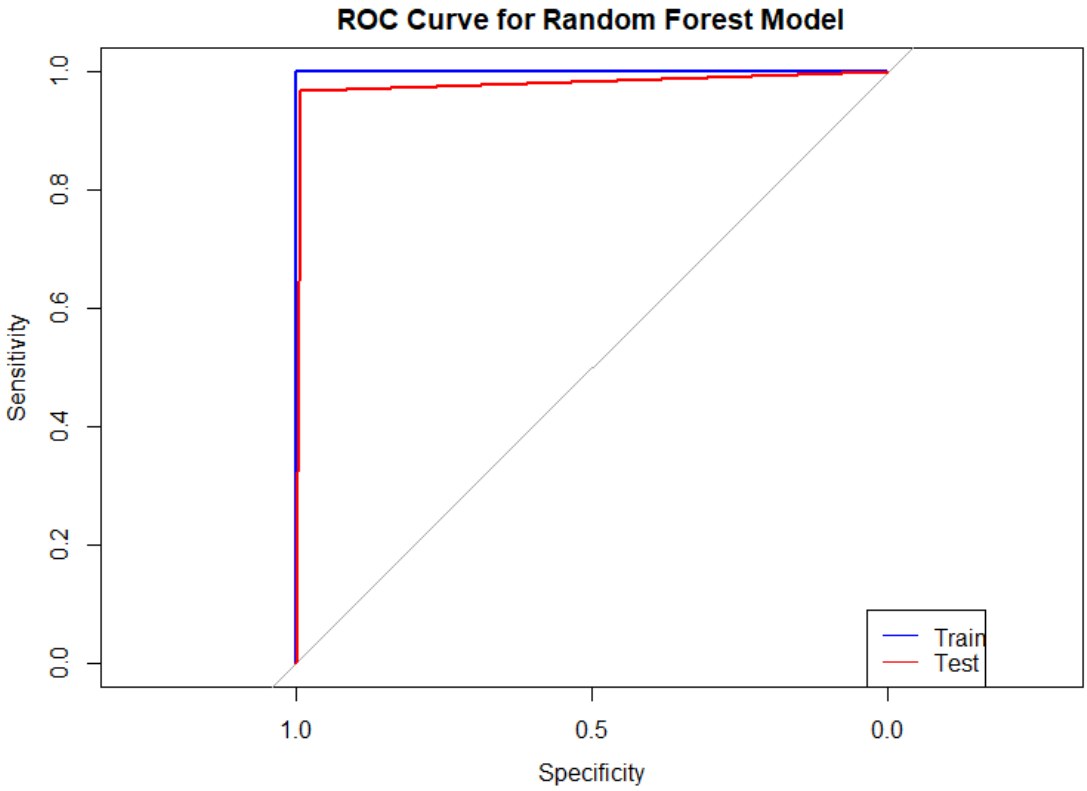

Figure 18: ROC Curve for Random Forest Model

- Confusion matrix:

		Train_Predicted		Test_Predicted	
	n= 6394	**No**	**Yes**	**No**	**Yes**
Actual	**No**	TN= 5462	FP= 1	TN= 2348	FP= 12
	Yes	FN= 0	TP= 931	FN= 16	TP= 364

Table 16: Confusion Matrix for Random Forest Model

- Classification Statistics:

No.	Statistics	Train Data	Test Data
1	Accuracy	0.9998436	0.989781
2	Precision	0.998927	0.9680851
3	Recall	1	0.9578947
4	F1 Score	0.9994632	0.962963

Table 17: Accuracy, Precision, Recall, F1-score for Random Forest Model

The random forest model shows exceptional accuracy on both the train and test datasets. It can correctly classify instances with an accuracy of nearly 99.98% on the train data and around 98.98% on the test data.

The model achieves very high precision values on both train and test datasets, indicating that when it predicts a positive instance, it is highly likely to be correct.

The recall values are very high, particularly on the train dataset where it is perfect (1). On the test dataset, the model captures a substantial portion of actual positive instances, suggesting strong performance in identifying positive cases.

The F1 Score values are close to 1 on both train and test datasets, which indicates an excellent balance between precision and recall. This suggests that the model is consistently performing well across both aspects of classification.

Generally, the Decision Tree model exhibits exceptional predictive capabilities, it can accurately capture the underlying patterns in the data and effectively differentiate between the classes.

6.4. Neutral Network

Neural networks, or artificial neural networks (ANNs), constitute a subset of machine learning techniques and form the core of deep learning methodologies. ANNs consist of interconnected layers of nodes, including an input layer, one or more hidden layers, and an output layer. Each of these nodes, or artificial neurons, links to others and possesses an assigned weight and threshold. These networks hold significant capabilities in the fields of computer science and artificial intelligence, enabling rapid classification and grouping of data (Nielsen, 2015).

Build Neutral Network model using parameters:

No.	Parameters	Values
1	The first hidden layer	4 neurons
2	The second hidden layer	2 neurons
3	The activation function for the hidden layers	logistic (sigmoid) function

Table 18: The Neutral Network Parameters

The plot provides a visualization of the neural network model. It organized in layers, with nodes (neurons) in each layer. The leftmost layer represents the input layer, followed by hidden layers, and finally the output layer on the right. The value displayed within each node indicates the calculated activation level of that neuron for the given input data.

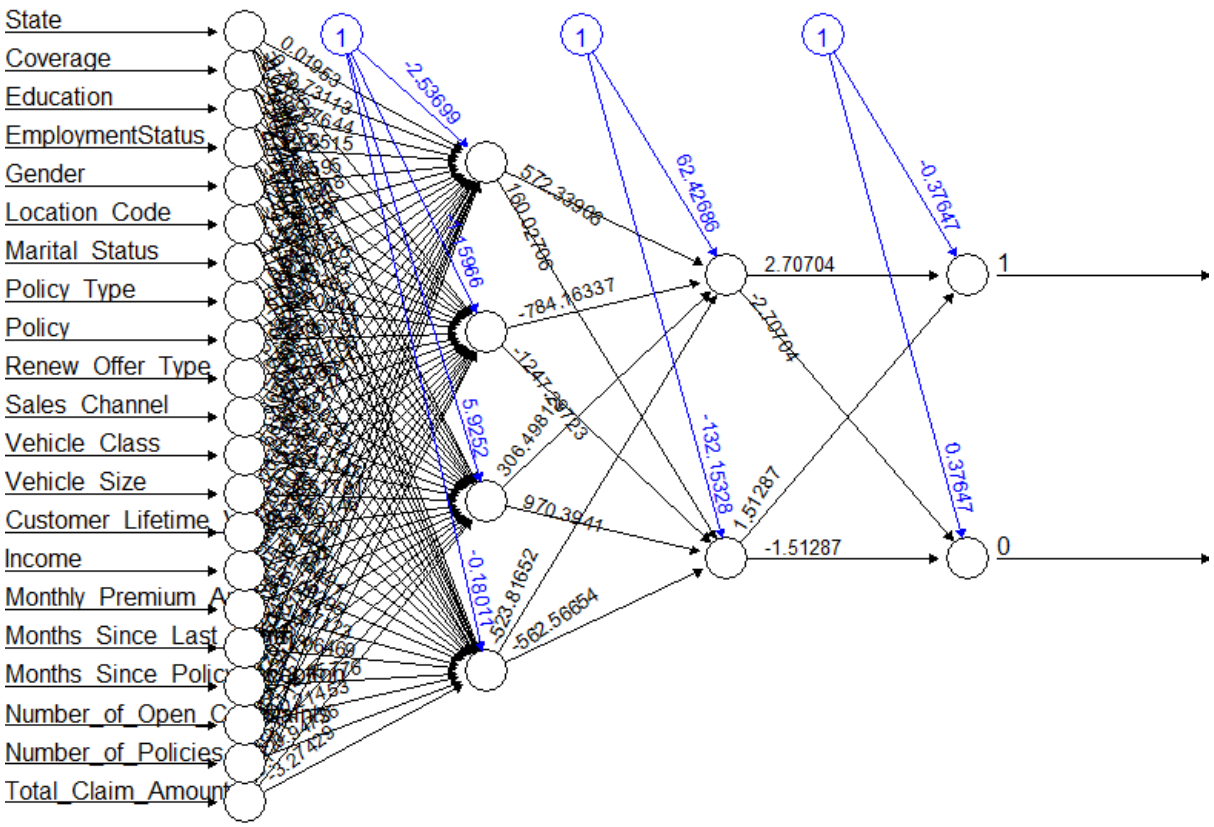

Figure 19: The Plot Neural Network Model

Evaluate the Neutral Network model:

- ROC Curve:

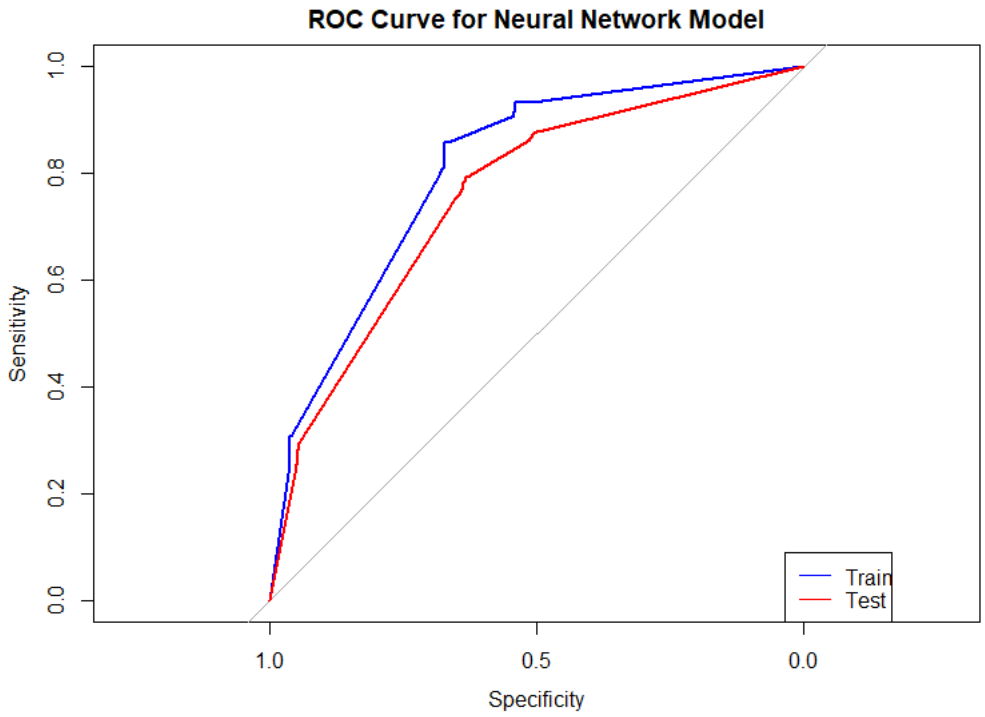

Figure 20: ROC Curve for Neural Network Model

- Confusion matrix:

		Train_Predicted		Test_Predicted	
	n= 6394	**No**	**Yes**	**No**	**Yes**
Actual	**No**	TN= 5265	FP= 197	TN= 2237	FP= 127
	Yes	FN= 645	TP= 287	FN= 265	TP= 111

Table 19: Confusion Matrix for Neural Network Model

- Classification Statistics:

No.	Statistics	Train Data	Test Data
1	Accuracy	0.868314	0.8569343
2	Precision	0.5929752	0.4663866
3	Recall	0.3079399	0.2952128
4	F1 Score	0.4053672	0.3615635

Table 20: Accuracy, Precision, Recall, F1-score for Neural Network Model

The Neural Network model indicate a similar performance pattern as observed in the Decision Tree model, with moderate levels of accuracy, precision, recall, and F1 Score on both the train and test datasets.

The Neural Network model demonstrates a performance profile that closely resembles the Decision Tree model, with similar levels of accuracy, precision, recall, and F1 Score. While the Neural Network's ability to capture complex relationships in the data may offer potential advantages over Decision Trees, the observed performance suggests that both models are facing challenges in effectively identifying positive instances.

6.5. Compare Classification Algorithms

No.	Algorithms	Accuracy		F1 Score	
		Train Data	Test Data	Train Data	Test Data
1	Logistic Regression	0.8539256	0.8624088	NaN	NaN
2	Decision Tree	0.8770723	0.8773723	0.32358	0.2758621
3	Random Forest	0.9998436	0.989781	0.9994632	0.962963
4	Neutral Network	0.863153	0.8609489	0.146341	0.116009

Table 21: Compare Logistic Regression, Decision Tree, Random Forest, and Neural Network

Among the Classification models, the Random Forest stands out with exceptionally high accuracy and F1 Score values, making it a strong performer for this dataset.

The Decision Tree and Neural Network model also performs reasonably well, with good recall but lower F1 Score values.

The Logistic Regression model with NaN F1 Score, indicating that it may not be the best fit for this dataset.

7. Discussion

In the context of this Customer Behavior Analysis project, utilizing data mining methods such as Dimensionality Reduction, Clustering, and Classification appears to be fitting for addressing the challenge of segmenting customers and predicting their behavior.

Dimensionality Reduction: Employing Principal Component Analysis (PCA) aids in diminishing the data's dimensions to two components, facilitating a clearer comprehension of data patterns, and enhancing the construction of a more effective clustering model.

Clustering: Among the clustering algorithms applied, PCA & K-means demonstrate superior performance with the highest Silhouette Value. While this outcome is satisfactory, there is potential for further enhancement by accessing data quality, conducting pre-processing, scaling, normalization, and experimenting with alternative algorithms to improve efficiency.

Classification: It is important to highlight that the machine learning approach (Random Forest) yields more favorable outcomes compared to the deep learning algorithm (Artificial Neural Networks - ANNs), possibly due to considerations of Data Size and Complexity. Random Forest is notably proficient with smaller or moderately complex datasets, while deep learning models excel when presented with larger datasets to capture intricate patterns. In this scenario, Random Forest demonstrates robust generalization capabilities with minimal overfitting risk, showcasing substantial accuracy for both training and test data. Nonetheless, it remains imperative to continuously monitor and assess the model's real-world effectiveness, allowing for algorithm adjustments to optimize efficiency.

8. Conclusions

In conclusion, this project has provided a practical and hands-on experience in the application of data mining techniques, identifying their application and significance within the marketing domain. By deploying data mining techniques, including customer segmentation, predictive modeling, has provided a deeper understanding of customer behavior patterns and preferences. These invaluable insights empower enterprises to customize their marketing endeavors, effectively catering to the diverse requirements and anticipations of distinct customer segments.

Amidst intense competition and the evolving marketing, recognizing customer psychology and crafting apt strategies emerge as pivotal drivers for sustained business growth and viability. In the future, it is evident that Customer Behavior Analytics will remain an essential tool in the marketer's toolkit, guiding strategic decisions that resonate with customers on a deeper level and drive success in the business. This analytical tool will continue to inform strategic decisions that strike a profound chord with customers, fostering success in an ever-vigilant competitive environment.

References

Analytics, I. W. (2019). *IBM Watson Marketing Customer Value Data*. Retrieved from Kaggle: https://www.kaggle.com/datasets/pankajjsh06/ibm-watson-marketing-customer-value-data

Andrew. (2017, 5 27). *Machine Learning Yearning.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Feature_scaling

Anil K. Jain, Richard C. Dubes. (1988). *Algorithms for Clustering Data.* Prentice Hall.

Breiman, L. (2001). Random Forests. *Machine Learning, Volume 45, Issue 1*, 5-32.

Brooks. (1988). *Principal component analysis*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Principal_component_analysis

Customer Service. (2023, 2 24). Retrieved from superoffice: https://www.superoffice.com/blog/customer-experience-statistics/

Daniel A. McFadden. (2017). Clustering and the Continuous Logit Model: A Bayesian Approach. *Journal of Business & Economic Statistics, Volume 35, Issue 1*, 1-16.

Daniel P. Russo. (2019). Clever Machines Learn How to Learn: The Confusion Matrix Through the Eyes of a Neural Network. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, (pp. 4189-4198).

David Arthur and Sergei Vassilvitskii. (2007). k-means++: The Advantages of Careful Seeding. *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '07).*

Fawcett, T. (2004). ROC Graphs: Notes and Practical Considerations for Data Mining Researchers. *ACM SIGKDD Explorations Newsletter, Volume 6, Issue 1*, 12-16.

Gareth James, D. W. (2017). *An Introduction to Statistical Learning: With Applications in R.* Retrieved from Talend: https://www.talend.com/resources/data-transformation-defined/

Han, Jiawei; Kamber, Micheline; Pei, Jian. (2012). Data Preprocessing. In J. Han, M. Kamber, & J. Pei, *Data Mining Concepts and Techniques (Third Edition)* (p. 83). 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann.

Hinton, L. v. (2008). *Dimensionality reduction*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Dimensionality_reduction

Martin Ester; Hans-Peter Kriegel; Jörg Sander; Xiaowei Xu. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)* (pp. 226–231). AAAI Press.

Nielsen, M. (2015). *Neural Networks and Deep Learning*. Retrieved from Neural Networks and Deep Learning: http://neuralnetworksanddeeplearning.com/

Paul D. Allison. (2019). *Logistic Regression Using SAS: Theory and Application.* SAS Institute.

Peter J. Rousseeuw. (1987). Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics, Volume 20*, 53-65.

Tim K. Marks; Thomas J. Cova. (2011). Spatial Cluster Analysis and Inference with the Bootstrap and GIS. *Geographical Analysis, Volume 43, Issue 2*, 216-235.

Trevor Hastie, Robert Tibshirani, Jerome Friedman. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer.

Trevor Hastie; Robert Tibshirani; Jerome Friedman. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer.

Trevor Hastie; Robert Tibshirani; Jerome Friedman. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer.

Use the scale(). (2021, 12 16). Retrieved from R-Bloggers: https://www.r-bloggers.com/2021/12/how-to-use-the-scale-function-in-r/

Appendix A: Code R

Github: https://github.com/ThyTina/Data_Mining