DM PROJECT BUSSINESS REPORT

Kratik Mehta

PGP-DSBA Feb 2022

Table of Contents

List of Figures

List of Tables

Problem 1: Clustering

Executive Summary

A leading bank wants to develop a customer segmentation to give promotional offers to its customers. They collected a sample that summarizes the activities of users during the past few months. You are given the task to identify the segments based on credit card usage.

1.1 Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis).

Sample of the Bank Marketing Dataset.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837

Table 1: Bank Marketing Dataset

Data Dictionary

1. **spending**: Amount spent by the customer per month (in 1000s)
2. **advance_payments**: Amount paid by the customer in advance by cash (in 100s)
3. **probability_of_full_payment**: Probability of payment done in full by the customer to the bank
4. **current_balance**: Balance amount left in the account to make purchases (in 1000s)
5. **credit_limit**: Limit of the amount in credit card (in 10000s)
6. **min_payment_amt**: minimum paid by the customer while making payments for purchases made monthly (in 100s)
7. **max_spent_in_single_shopping**: Maximum amount spent in one purchase (in 1000s)

Checking the types of variables in the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   spending                      210 non-null    float64
 1   advance_payments              210 non-null    float64
 2   probability_of_full_payment   210 non-null    float64
 3   current_balance               210 non-null    float64
 4   credit_limit                  210 non-null    float64
 5   min_payment_amt               210 non-null    float64
 6   max_spent_in_single_shopping  210 non-null    float64
dtypes: float64(7)
memory usage: 11.6 KB
```

From the above output we can see that:

- There are **210 observations** of different individuals in the data.
- There are **7 variables** in the dataset.
- All the variables are of **continuous numerical (float) type**.
- The dataset **does not have any missing values**.

Descriptive Statistics of the dataset

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000
mean	14.847524	14.559286	0.870999	5.628533	3.258605	3.700201	5.408071
std	2.909699	1.305959	0.023629	0.443063	0.377714	1.503557	0.491480
min	10.590000	12.410000	0.808100	4.899000	2.630000	0.765100	4.519000
25%	12.270000	13.450000	0.856900	5.262250	2.944000	2.561500	5.045000
50%	14.355000	14.320000	0.873450	5.523500	3.237000	3.599000	5.223000
75%	17.305000	15.715000	0.887775	5.979750	3.561750	4.768750	5.877000
max	21.180000	17.250000	0.918300	6.675000	4.033000	8.456000	6.550000

Table 2: Descriptive Statistics of the dataset

Looking at the mean values from the above 5-point summary, we see that on an average customers spend approximately 14847 per month. They pay 1455 in advance on average. The average probability that a customer will pay the bank in full is 0.871.

The average current balance for all customers is around 5628, while the credit limit is around 32586. From all the customers, the minimum amount paid for monthly purchases is 370 on average while maximum amount spent in one shopping is 5408.

Distribution of the spending variable.

```
Skewness: 0.3999
```

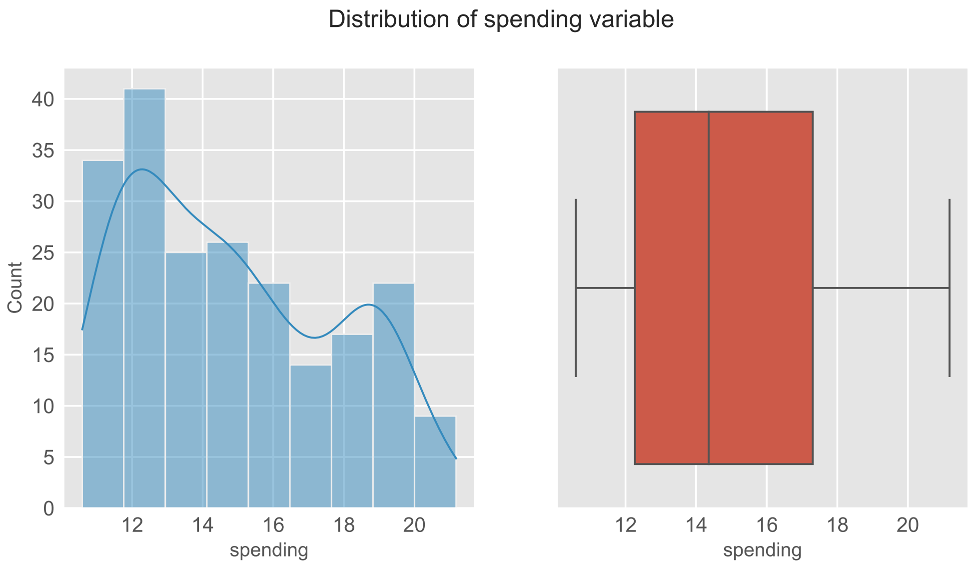

Figure 1: Distribution of the spending variable

The data in the **spending** variable is *moderately skewed to the right*. Also, it does not contain any outliers.

Distribution of the advance_payments variable.

```
Skewness: 0.3866
```

From the below figure we see that, the data in the **advance_payments** variable is *moderately skewed to the right*. Also, it does not contain any outliers.

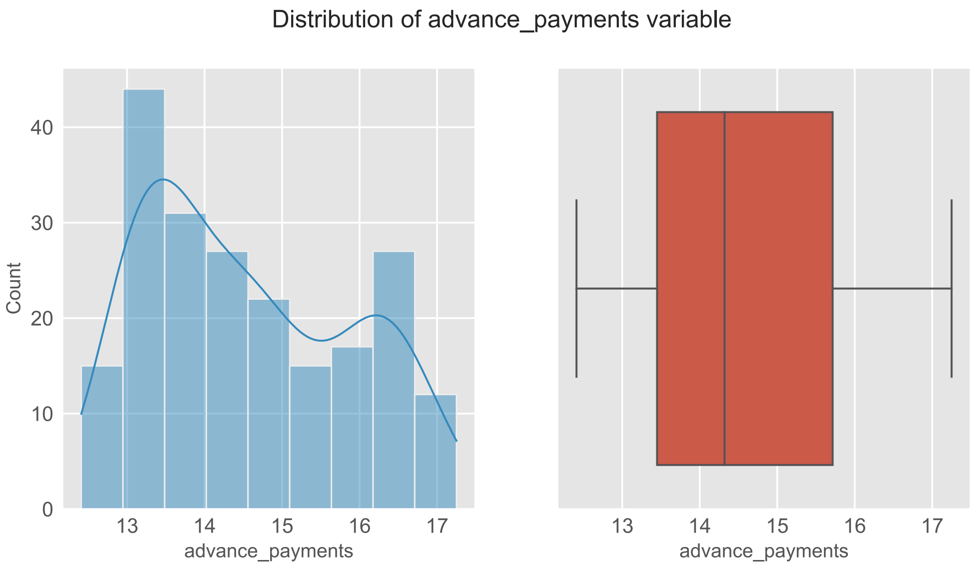

Figure 2: Distribution of the advance_payments variable.

Distribution of the probability_of_full_payment variable.

```
Skewness: -0.5380
```

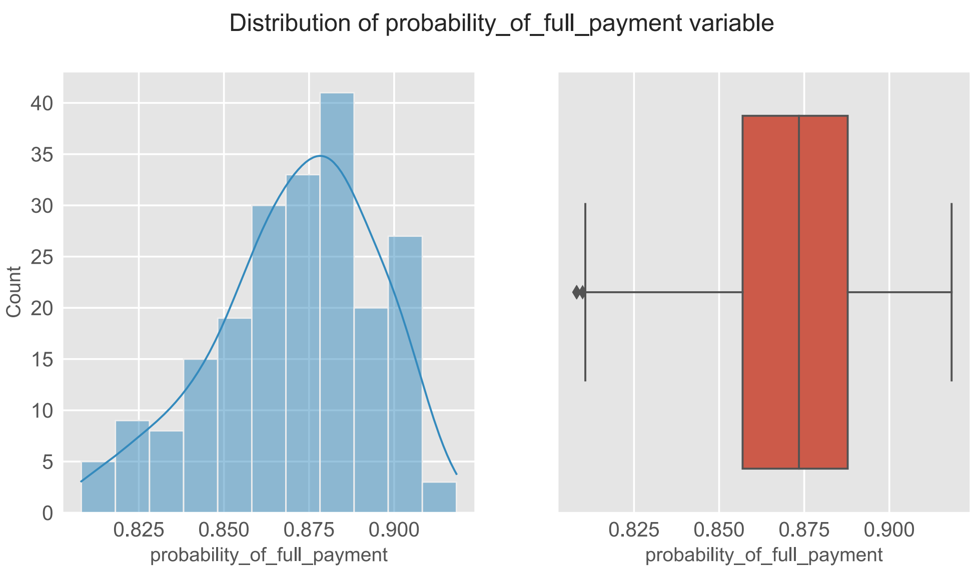

Figure 3: Distribution of the probability_of_full_payment variable.

The data in the **probability_of_full_payment** variable is *skewed to the left*. Also, it contains **two outliers** on the lower side.

Distribution of the current_balance variable.

```
Skewness: 0.5255
```

From the below figure we see that, the data in the **current_balance** variable is *skewed to the right*. Also, it does not contain any outliers.

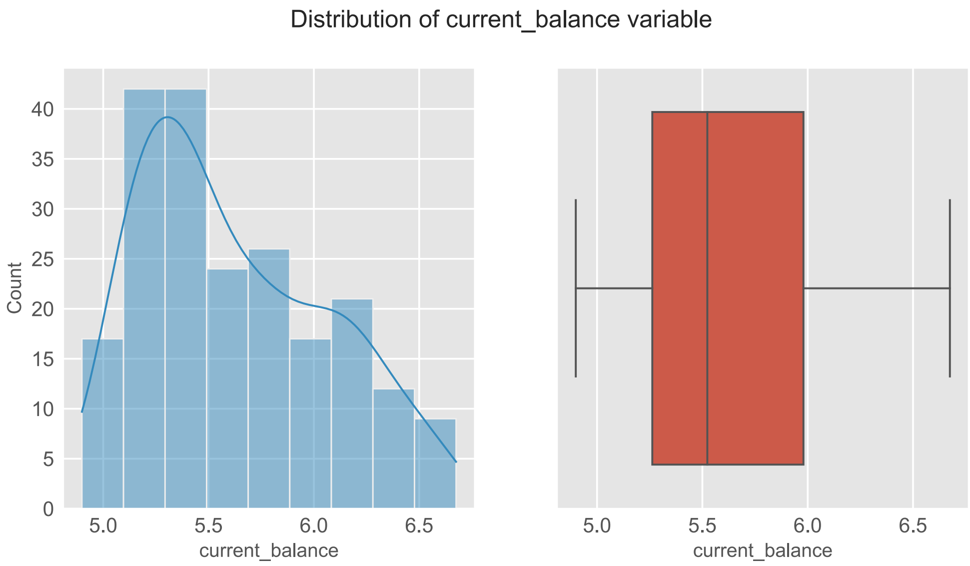

Figure 4: Distribution of the current_balance variable.

Distribution of the credit_limit variable.

```
Skewness: 0.1344
```

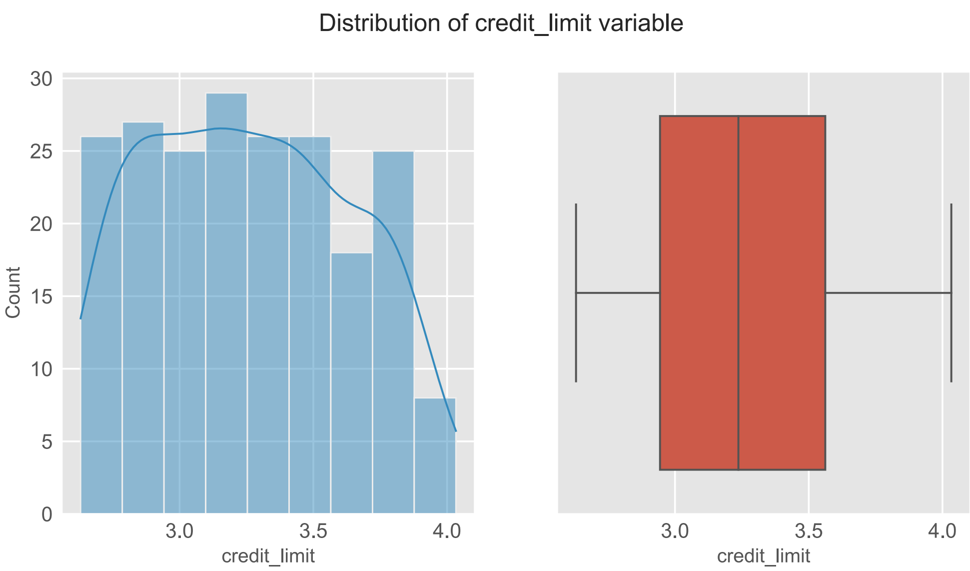

Figure 5: Distribution of the credit_limit variable.

The data in the **credit_limit** variable is *slightly skewed to the right*. Also, it does not contain any outliers.

Distribution of the min_payment_amt variable.

```
Skewness: 0.4017
```

From the below figure we see that, the data in the **min_payment_amt** variable is *moderately skewed to the right*. Also, it contains **two outliers** on the upper side.

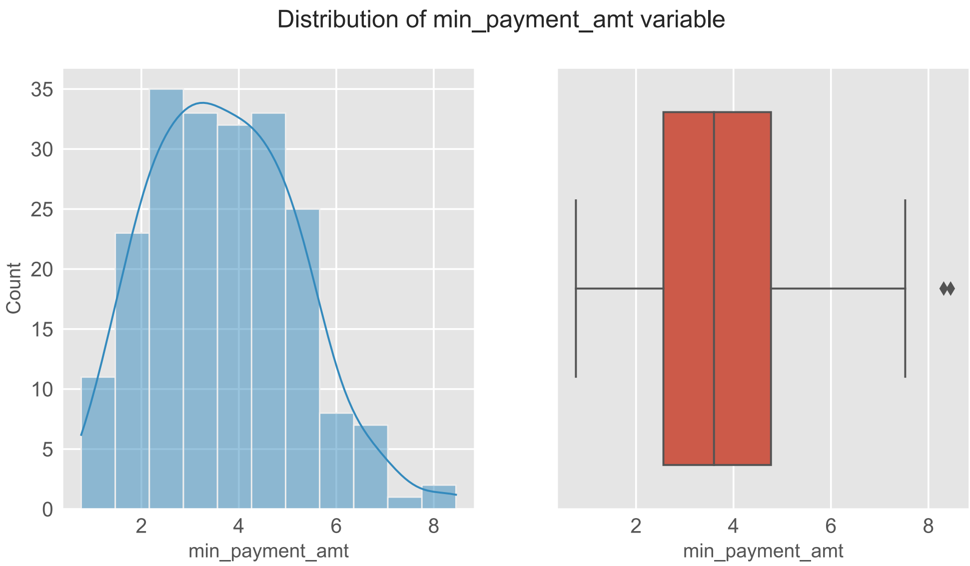

Figure 6: Distribution of the min_payment_amt variable.

Distribution of the max_spent_in_single_shopping variable.

```
Skewness: 0.5619
```

Figure 7: Distribution of the max_spent_in_single_shopping variable.

The data in the **max_spent_in_single_shopping** variable is *skewed to the right*. Also, it does not contain any outliers.

Inference

From the above plots we can conclude that most of the individuals in our dataset have a higher spending capacity, high current balance in their accounts and these customers spent a higher amount during a single shopping transaction. Most of the individuals have a higher probability to make full payment to the bank.

Correlation Heatmap of continuous variables.

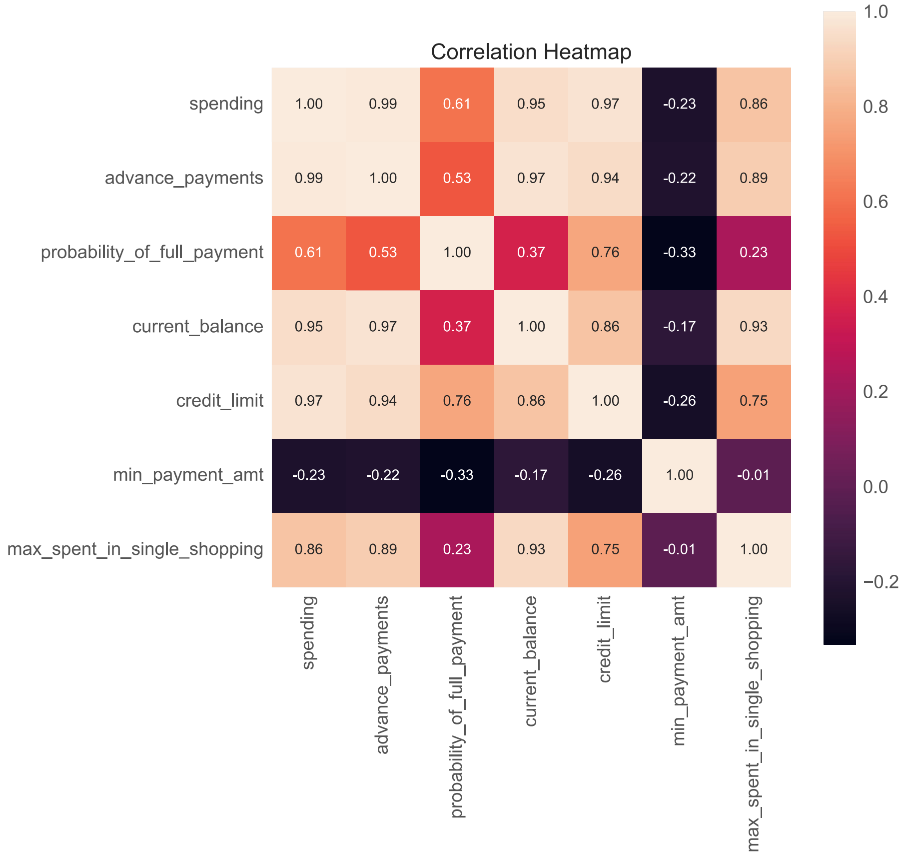

Figure 8: Correlation Heatmap

From the above correlation plot we can see that:

1. Variables **spending**, **advance_payments**, **current_balance**, **credit_limit** and **max_spent_in_single_shopping** are *highly correlated* to each other.
2. **probability_of_full_payment** is *highly correlated* to **credit_limit** and *moderately correlated* to **spending** and **advance_payments**.
3. **min_payment_amt** is *negatively correlated* to **spending**, **advance_payments**, **probability_of_full_payment**, **current_balance**, and **credit_limit.**

1.2 Do you think scaling is necessary for clustering in this case? Justify

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
mean	14.847524	14.559286	0.870999	5.628533	3.258605	3.700201	5.408071
std	2.909699	1.305959	0.023629	0.443063	0.377714	1.503557	0.491480

Table 3: Mean and Standard deviation of the data.

From the above table we can confirm that the **means and standard deviations of the variables** in the dataset are **different from each other**. This confirms that the **scales differ** for the variables. Hence, we will have to scale the dataset.

Here scaling is done using the **StandardScaler** function. This function calculates the mean and standard deviation of the variables separately, and scales the data such that the mean is zero and standard deviation is one for all the variables. Below formula is used to obtain scaled values from original values.

Here, μ is the mean of the variable and σ is the standard deviation.

$$X_{scaled} = \frac{X - \mu}{\sigma}$$

Equation 1: Standardization Formula

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754355	1.811968	0.178230	2.367533	1.338579	-0.298806	2.328998
1	0.393582	0.253840	1.501773	-0.600744	0.858236	-0.242805	-0.538582
2	1.413300	1.428192	0.504874	1.401485	1.317348	-0.221471	1.509107
3	-1.384034	-1.227533	-2.591878	-0.793049	-1.639017	0.987884	-0.454961
4	1.082581	0.998364	1.196340	0.591544	1.155464	-1.088154	0.874813

Table 4: Scaled Bank Marketing Dataset

1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them.

Hierarchical clustering is an unsupervised machine learning technique in which the algorithm combines data points which are close to each other into clusters in a bottom-up approach.

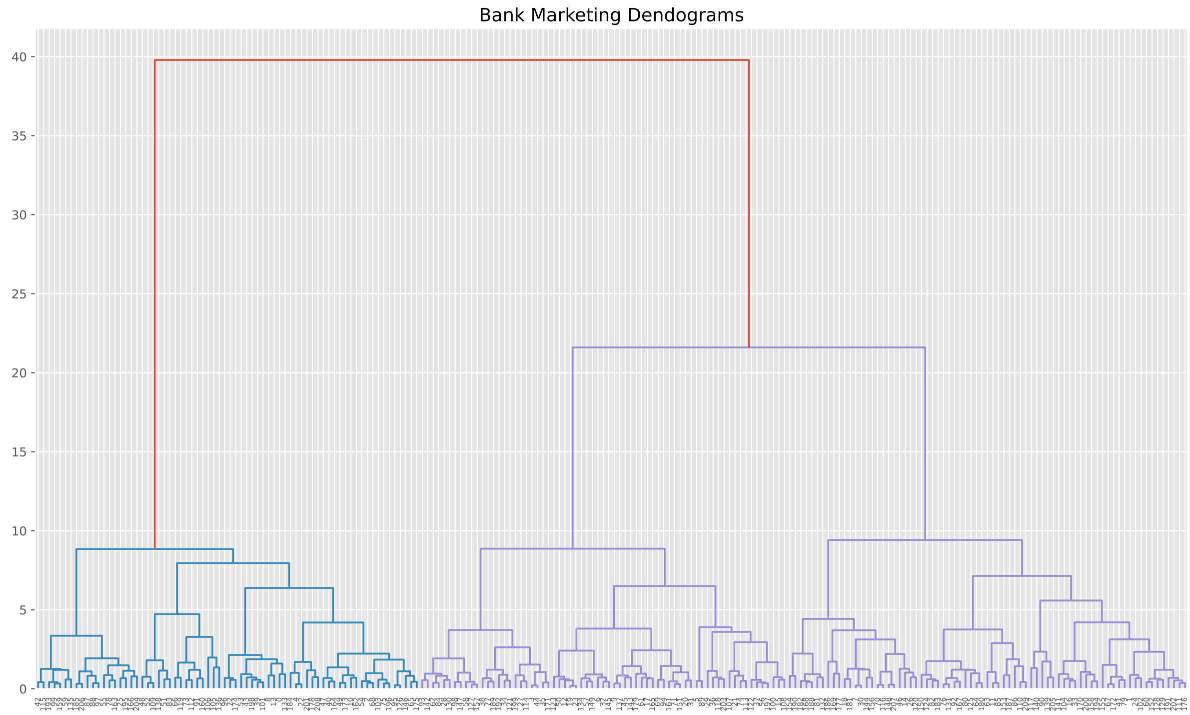

Figure 9: Dendrogram

From the above plot we see that, the dendrogram is suggesting 2 clusters i.e., blue and purple clusters. We can also see that at a threshold distance of 10, the number of clusters are 3 which are of almost equal sizes. Hence, we can divide the data into either 2 or 3 clusters. We proceed with 2 clusters for further analysis.

Here, the hierarchical clustering is done using AgglomerativeClustering function. The distance measure (or affinity) used is the **Euclidean distance**. **Ward linkage** method is used for calculating the distance between clusters for merging criterion.

To visualize the clusters, **Principal Component Analysis** was used to reduce the **dimensionality from 7 to 2**. The scatter plot with color coding using the 2 cluster labels is shown below. From the below figure we see that; the clusters are well separated from each other. The purple cluster at the top is a smaller cluster while the orange cluster at the bottom is a larger cluster.

Visualising Customer Segmentation

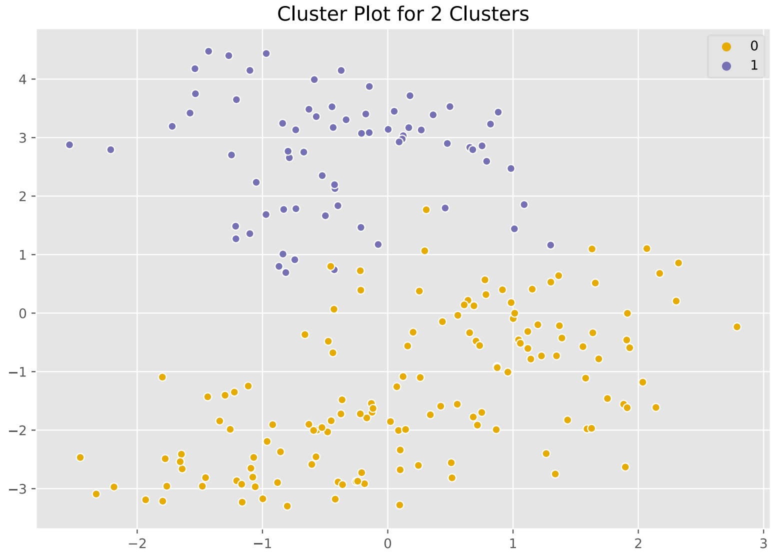

Figure 10: Cluster plot for 2 clusters.

1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Explain the results properly. Interpret and write inferences on the finalized clusters.

K-means is an unsupervised machine learning technique in which the algorithm identifies k number of centroids and assigns every data point to the nearest cluster.

Elbow Method

To determine optimum number of clusters, Elbow Plot can be used. In this plot, the number of clusters are on the X-axis and the corresponding within-cluster sum of squares (WSS) are on the Y-axis. The Elbow Plot for the Bank Marketing dataset with 1 to 10 clusters is shown below.

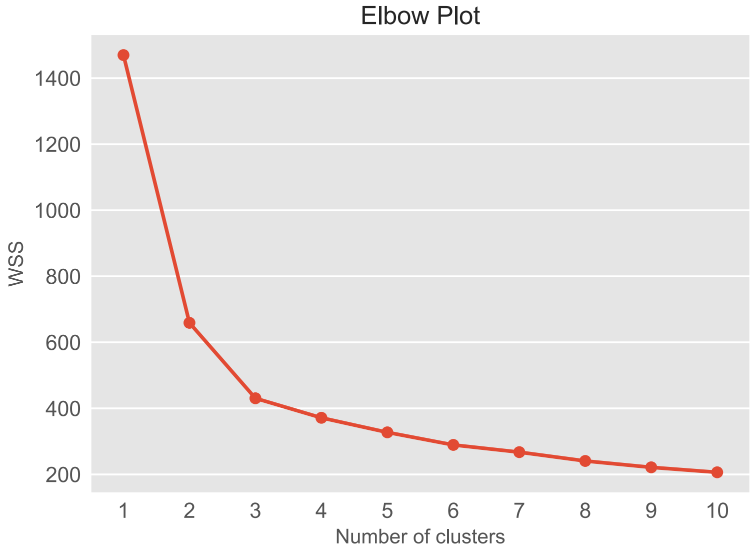

Figure 11: Elbow Plot for K-Means Clustering

From the above plot we see that, the drop in WSS for clusters 1 to 2 is very large. Similarly, the drop from clusters 2 to 3 is also significant. After cluster 3 the curve gets flatter. Therefore, the plot suggests the optimal number of clusters as 3.

Silhouette Method

Silhouette method measures how tightly the observations are clustered and the average distance between clusters. For each observation a silhouette score is constructed which is a function of the average distance between the point and all other points in the cluster to which it belongs, and the distance between the point and all other points in all other clusters, that it does not belong to.

```
The Average Silhouette Score for 2 clusters is 0.46577
The Average Silhouette Score for 3 clusters is 0.40073
The Average Silhouette Score for 4 clusters is 0.3369
The Average Silhouette Score for 5 clusters is 0.28314
The Average Silhouette Score for 6 clusters is 0.29034
The Average Silhouette Score for 7 clusters is 0.26541
The Average Silhouette Score for 8 clusters is 0.25194
The Average Silhouette Score for 9 clusters is 0.25558
The Average Silhouette Score for 10 clusters is 0.25952
```

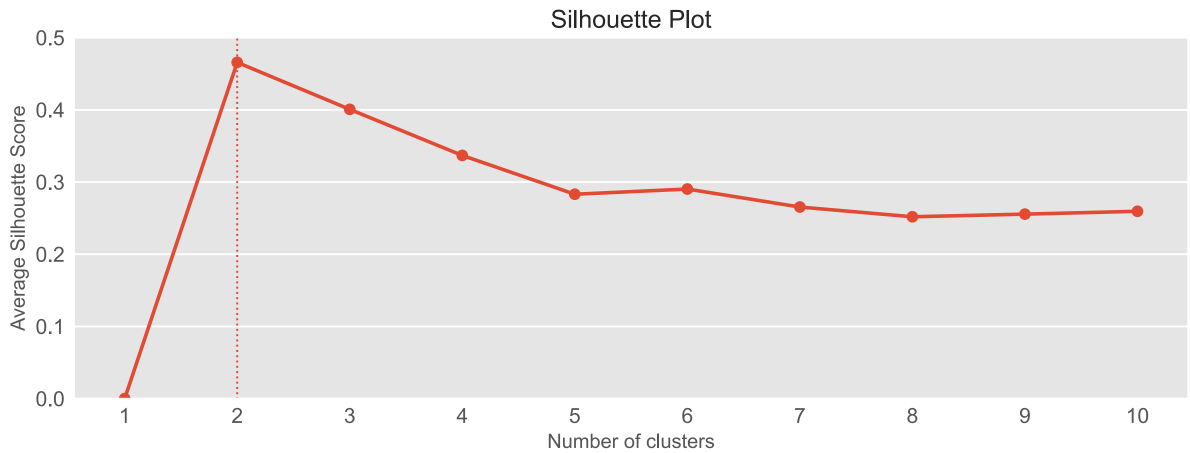

Figure 12: Silhouette Plot

From the above Silhouette Plot, we see that the silhouette score for 2 clusters is maximum. But the Elbow method suggested the optimal number of clusters as 3. Having only 2 clusters for market segmentation might not be inferential for the business. As the difference between the silhouette score for 2 and 3 clusters is not much, we choose optimal number of clusters as 3.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Cluster
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	2
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	0
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	2
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	1
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837	2

Table 5: Sample of Original dataset with Clusters

Visualising Customer Segmentation

To visualize the clusters, **Principal Component Analysis** was used to reduce the **dimensionality from 7 to 2**. The scatter plot with color coding using the 3 cluster labels is shown below. From the below figure we see that; all the clusters are well separated from each other. Also, all the 3 clusters are approximately of the same size.

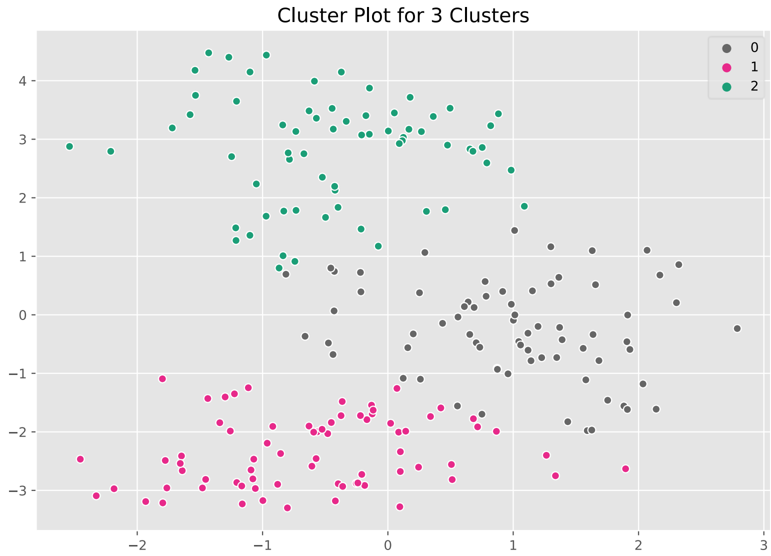

Figure 13: Cluster plot for 3 clusters.

1.5 Describe cluster profiles for the clusters defined. Recommend different promotional strategies for different clusters.

In the below table, the averages of the data for all the variables in the 3 clusters obtained from the K-means clustering methods are computed.

Cluster	0	1	2
spending	14.438	11.857	18.495
advance_payments	14.338	13.248	16.203
probability_of_full_payment	0.882	0.848	0.884
current_balance	5.515	5.232	6.176
credit_limit	3.259	2.850	3.698
min_payment_amt	2.707	4.742	3.632
max_spent_in_single_shopping	5.121	5.102	6.042

Table 6: Cluster Profiles

Inferences and Recommendations

1. For Cluster 0, the mean amount spent by the customers per month is 14438 while the amount paid in advanced is 1433. The average probability of full payment is 88.2%. The balance left in account is 5515 and the credit limit is 32590. The minimum amount paid by the customers is 270 while maximum amount spent in one purchase is 5121. **This group represents customers who spend moderately. We can increase credit limit or can lower interest rate by promoting premium cards/loyalty cards to increase spending.**
2. For Cluster 1, the mean amount spent by the customers per month is 11857 while the amount paid in advanced is 1325. The average probability of full payment is 84.8%. The balance left in account is 5232 and the credit limit is 28500. The minimum amount paid by the customers is 474 while maximum amount spent in one purchase is 5102. **This group represents customers who have low spending power. We can promote cards with offers such as zero annual charges and providing them with benefits such as free coupons or cashback rewards and fee waivers on a variety of places.**
3. For Cluster 2, the mean amount spent by the customers per month is 18495 while the amount paid in advanced is 1620. The average probability of full payment is 88.4%. The balance left in account is 6176 and the credit limit is 36980. The minimum amount paid by the customers is 363 while maximum amount spent in one purchase is 6042. **This group represents customers who have high spending power. We can offer reward points or discounts on their next big transaction to improve their loyalty.**

Problem 2: CART-RF-ANN

Executive Summary

An Insurance firm providing tour insurance is facing higher claim frequency. The management decides to collect data from the past few years. We are assigned the task to make a model which predicts the claim status and provide recommendations to management. Models like CART, RF & ANN are used and the models' performances on train and test sets are compared.

2.1 Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis).

Sample of the Insurance Dataset.

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	C2B	Airlines	No	0.70	Online	7	2.51	Customised Plan	ASIA
1	36	EPX	Travel Agency	No	0.00	Online	34	20.00	Customised Plan	ASIA
2	39	CWT	Travel Agency	No	5.94	Online	3	9.90	Customised Plan	Americas
3	36	EPX	Travel Agency	No	0.00	Online	4	26.00	Cancellation Plan	ASIA
4	33	JZI	Airlines	No	6.30	Online	53	18.00	Bronze Plan	ASIA

Table 7: Sample of the Insurance Dataset.

Data Dictionary

1. **Claimed**: Claim Status (Target)
2. **Agency_Code**: Code of tour firm
3. **Type**: Type of tour insurance firms
4. **Channel**: Distribution channel of tour insurance agencies
5. **Product Name**: Name of the tour insurance products
6. **Duration**: Duration of the tour (in days)
7. **Destination**: Destination of the tour
8. **Sales**: Amount worth of sales per customer in procuring tour insurance policies in rupees (in 100's)
9. **Commision**: The commission received for tour insurance firm (in percentage of sales)
10. **Age**: Age of insured

Checking the types of variables in the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Age           3000 non-null   int64
 1   Agency_Code   3000 non-null   object
 2   Type          3000 non-null   object
 3   Claimed       3000 non-null   object
 4   Commision     3000 non-null   float64
 5   Channel       3000 non-null   object
 6   Duration      3000 non-null   int64
 7   Sales         3000 non-null   float64
 8   Product Name  3000 non-null   object
 9   Destination   3000 non-null   object
dtypes: float64(2), int64(2), object(6)
memory usage: 234.5+ KB
```

From the above output we can see that:

- There are **3000 observations** of different individuals in the data.

- There are **9 independent variables** in the dataset.
- The **Claimed** column is *the target variable*.
- There are *4 numeric columns* and *6 categorical (object) columns*.
- The dataset **does not have any missing values**.

Descriptive Statistics

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
count	3000.000000	3000	3000	3000	3000.000000	3000	3000.000000	3000.000000	3000	3000
unique	NaN	4	2	2	NaN	2	NaN	NaN	5	3
top	NaN	EPX	Travel Agency	No	NaN	Online	NaN	NaN	Customised Plan	ASIA
freq	NaN	1365	1837	2076	NaN	2954	NaN	NaN	1136	2465
mean	38.091000	NaN	NaN	NaN	14.529203	NaN	70.001333	60.249913	NaN	NaN
std	10.463518	NaN	NaN	NaN	25.481455	NaN	134.053313	70.733954	NaN	NaN
min	8.000000	NaN	NaN	NaN	0.000000	NaN	-1.000000	0.000000	NaN	NaN
25%	32.000000	NaN	NaN	NaN	0.000000	NaN	11.000000	20.000000	NaN	NaN
50%	36.000000	NaN	NaN	NaN	4.630000	NaN	26.500000	33.000000	NaN	NaN
75%	42.000000	NaN	NaN	NaN	17.235000	NaN	63.000000	69.000000	NaN	NaN
max	84.000000	NaN	NaN	NaN	210.210000	NaN	4580.000000	539.000000	NaN	NaN

Table 8: Descriptive Statistics of the dataset.

In the Insurance dataset, the average age of the customer is 38.09. The commission received for the tour insurance is 14.52% on the sales amount. The average sales amount per customer is Rs. 6024.99. The average duration of the tour for all customers is 70 days. Also, the minimum duration of travel is -1 day, which is not possible. This seems to be bad data which should be taken care of.

There are total 4 tour agencies of which EPX is the most frequent. Out of the 2 types of agencies, Travel Agency occur the most in the data. Also, there are 2 channels, 5 products and 3 destinations provided by the tour insurance firm in the data.

Distribution of the Age variable.

```
Skewness: 1.1497
```

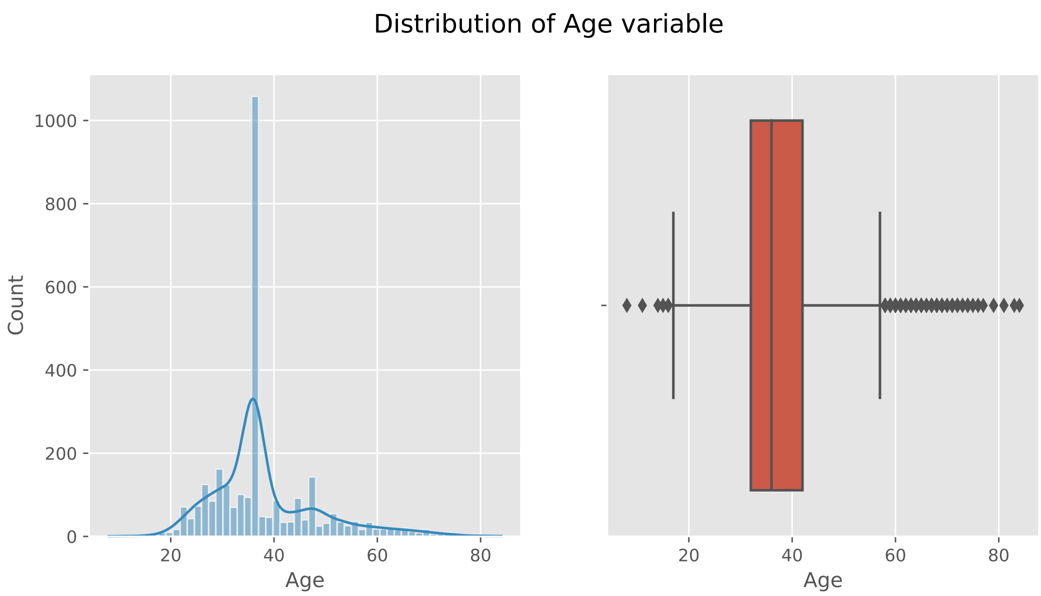

Figure 14: Distribution of the Age variable.

The data in the **Age** variable has *high positive skewness*. There are also **many outliers present** in the variable.

Distribution of the Agency_Code variable.

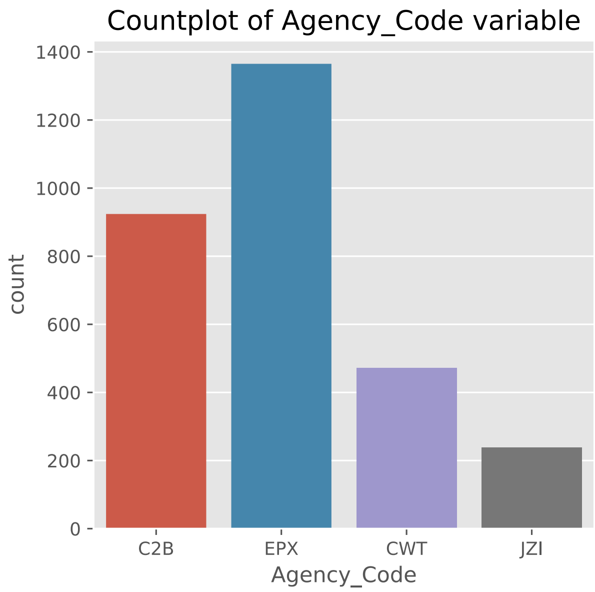

Figure 15: Distribution of the Agency_Code variable.

The agency **EPX has the highest number of customers** while **agency JZI has the lowest number of customers** in the dataset.

Distribution of the Type variable.

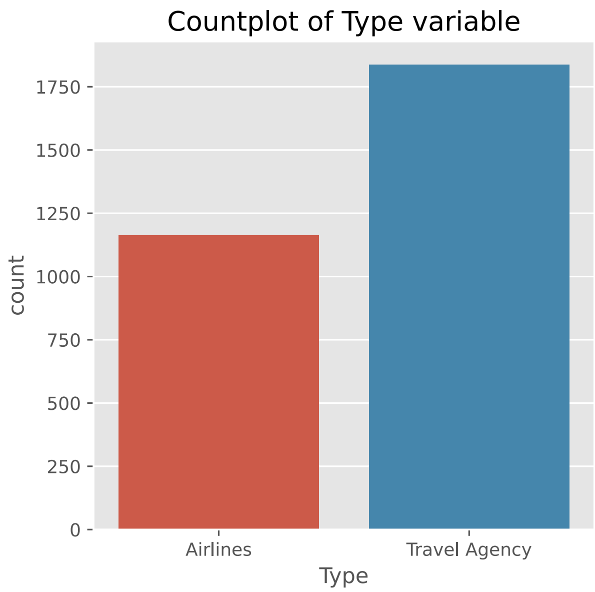

Figure 16: Distribution of the Type variable.

There **are more Travel Agency insurance firms than Airlines travel insurance firms** in the dataset.

Distribution of the Claimed variable.

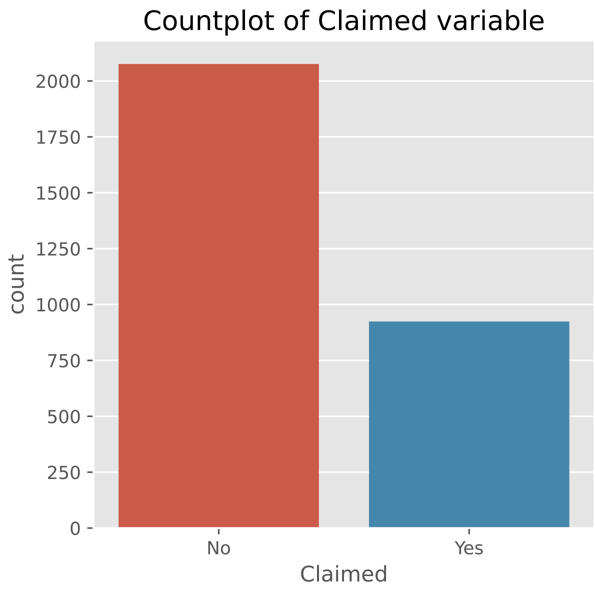

Figure 17: Distribution of the Claimed variable.

In the given dataset, **highest number of customers have not claimed the insurance**. Hence, **the dataset is imbalanced**.

Distribution of the Commision variable.

```
Skewness: 3.1489
```

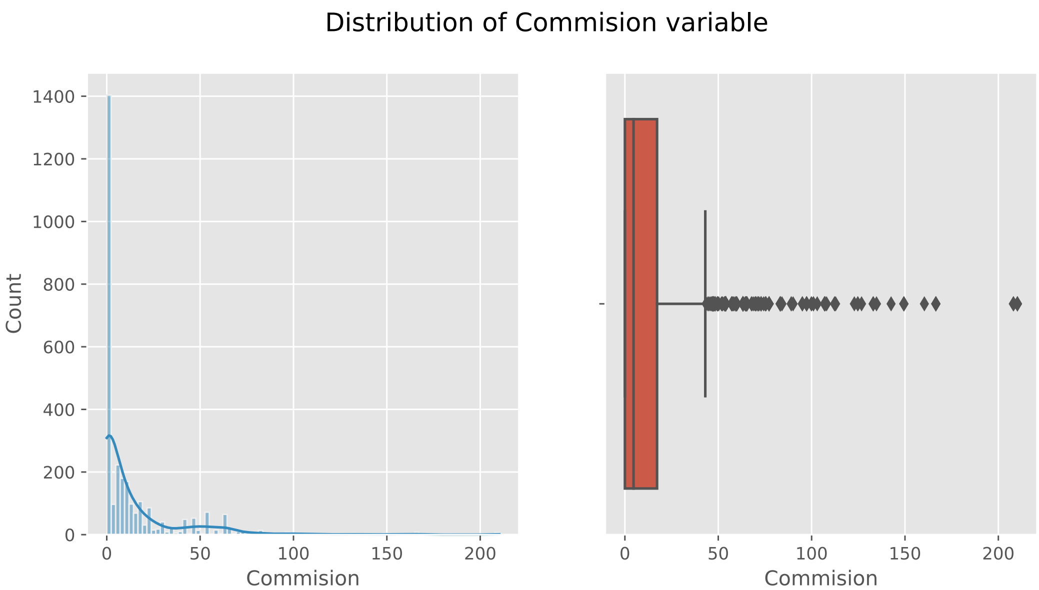

Figure 18: Distribution of the Commision variable.

The data in the **Commision** variable has *very high positive skewness*. There are also **many outliers present** in the variable.

Distribution of the Channel variable.

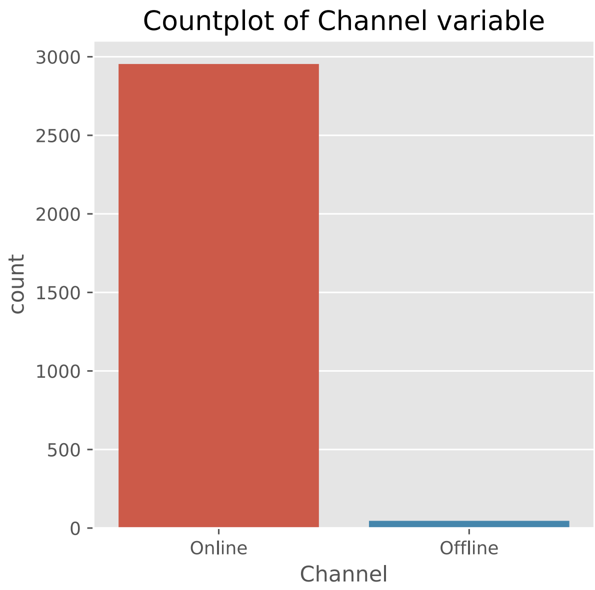

Figure 19: Distribution of the Channel variable.

Almost all of the insurances in the dataset are distributed online. Only few customers have taken the insurance in offline mode.

Distribution of the Duration variable.

```
Skewness: 13.7847
```

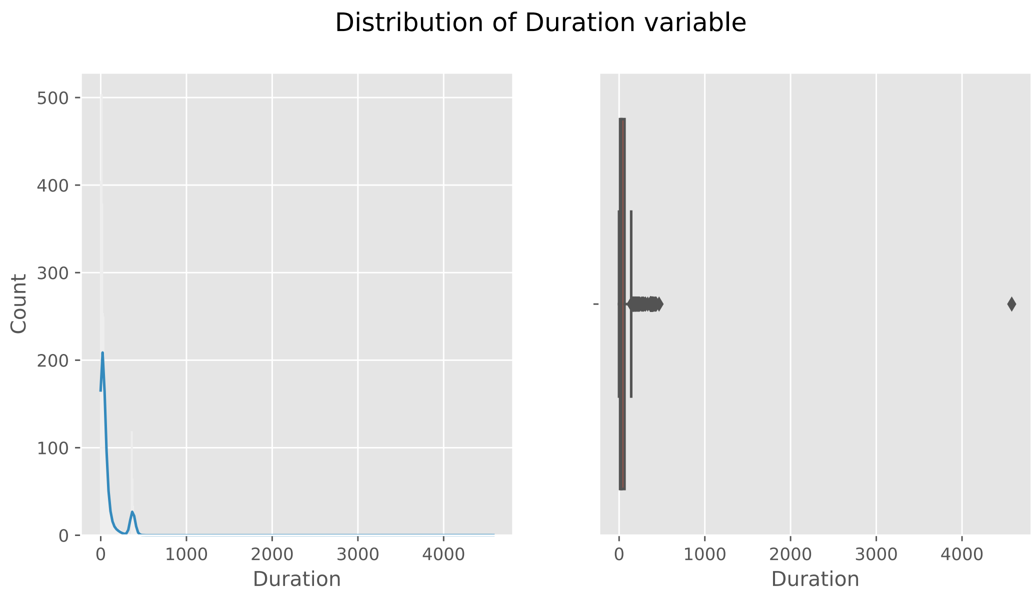

Figure 20: Distribution of the Duration variable.

The data in the **Duration** variable has *very high positive skewness*. There are also **many outliers present** in the variable. **The high skewness is due to one customer who has a travel duration of more than 4000 days**.

Distribution of the Sales variable.

```
Skewness: 2.3811
```

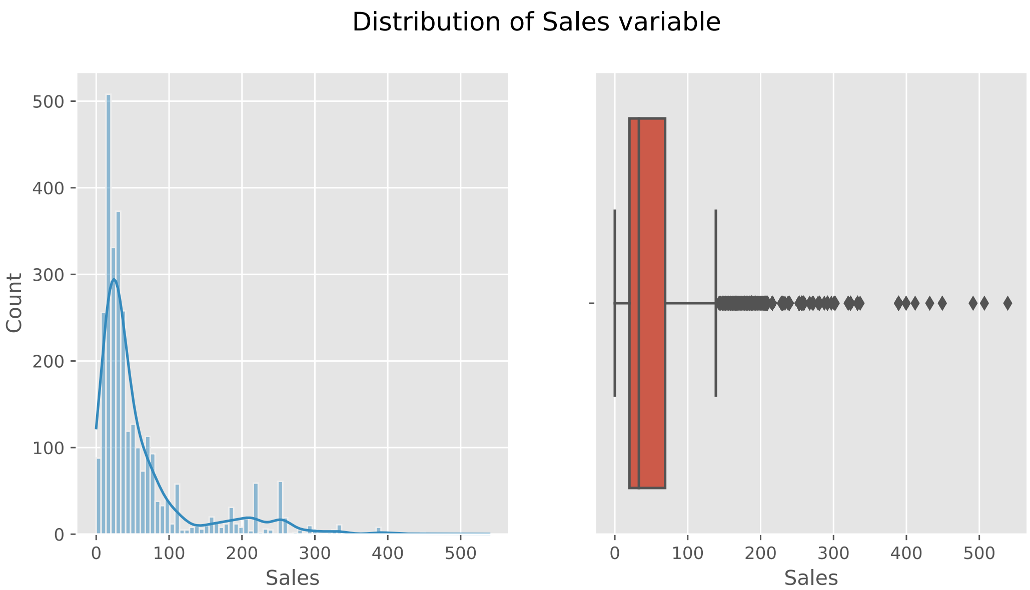

Figure 21: Distribution of the Sales variable.

The data in the **Sales** variable is *positively skewed*. There are also **many outliers present** in the variable.

Distribution of the Product Name variable.

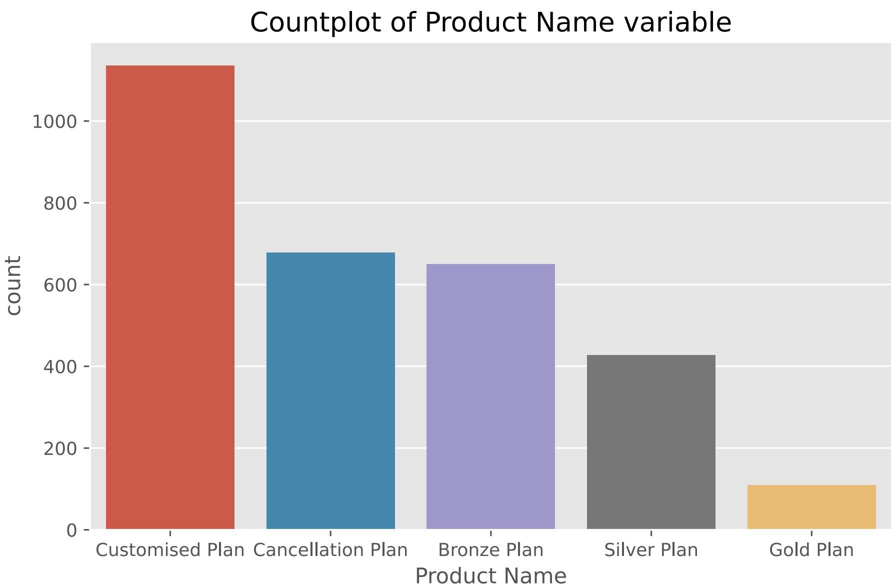

Figure 22: Distribution of the Product Name variable.

From the above figure we see that, **highest number of customers have taken Customised insurance plans**. The **Cancellation plan and the Bronze plan also seem to be popular** among customers. The **Gold plan is the least popular plan**.

Distribution of the Destination variable.

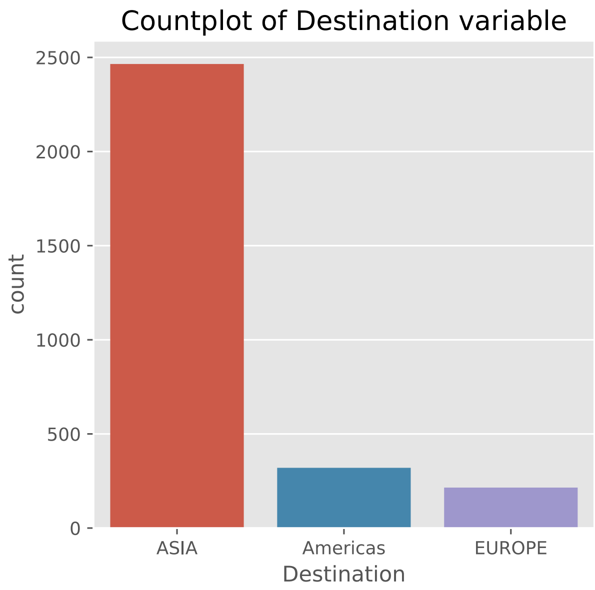

Figure 23: Distribution of the Destination variable.

The **Asia continent seem to be the most popular travel destination** among customers in the dataset.

Bivariate Analysis of Categorical variables with target variable.

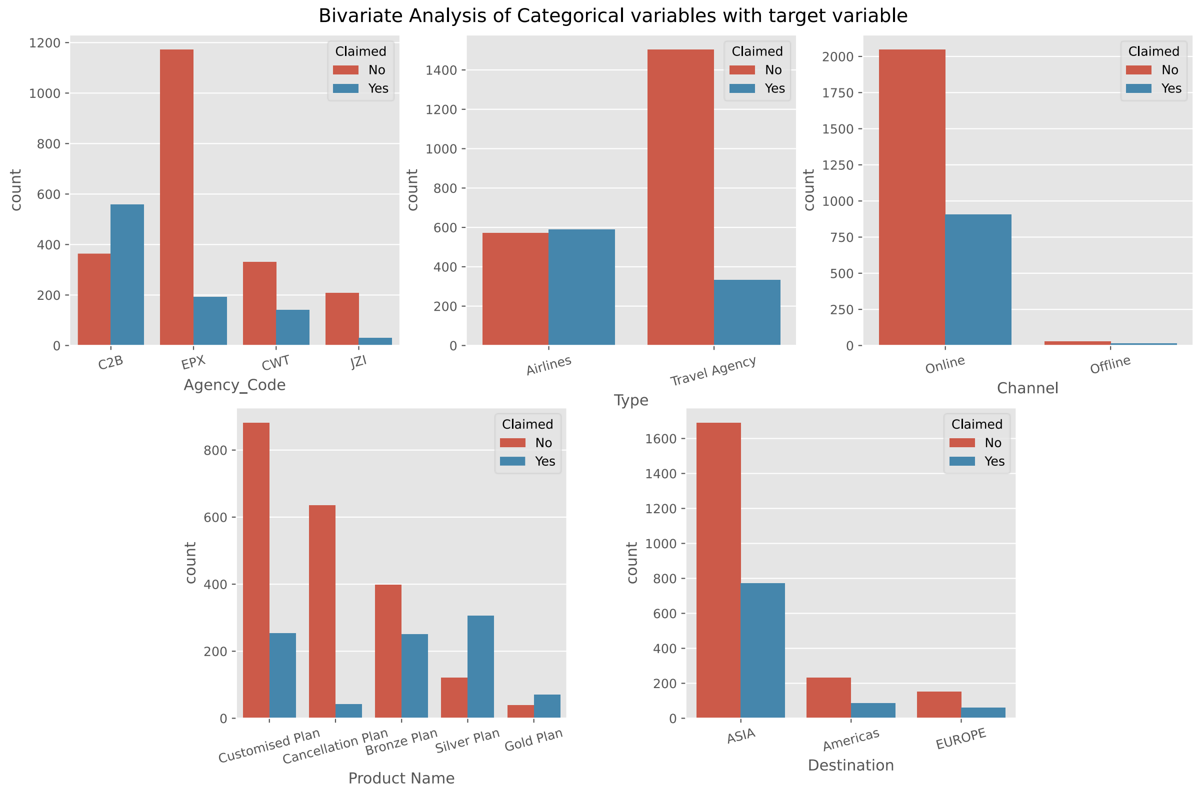

Figure 24: Bivariate Analysis of Categorical variables with target variable.

From the above plot we see that**, for the agency with code C2B, a greater number of customers have claimed their insurance** than those who have not claimed. Also, **for the Airlines travel firms, the number of customers**

who have claimed and who have not claimed are almost the same. Also, **most of the customers who have taken Silver plan have claimed their insurance money.**

Bivariate Analysis of Continuous variables.

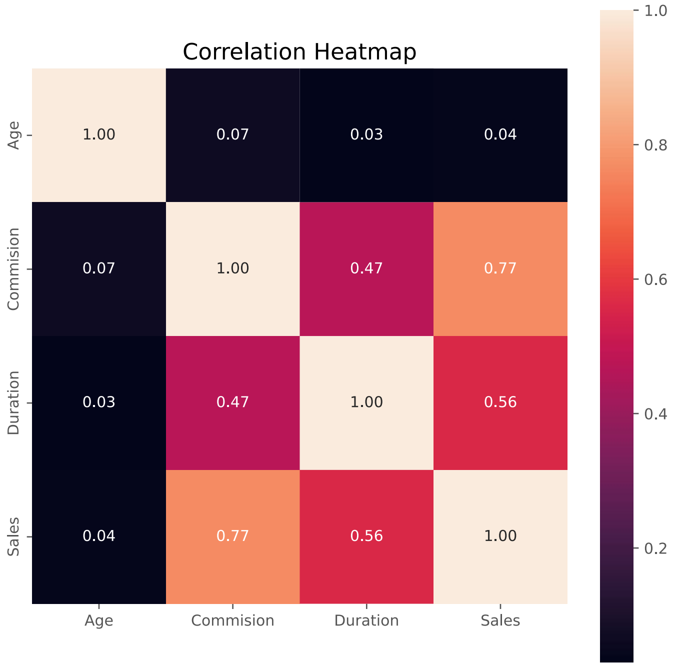

Figure 25: Bivariate Analysis of Continuous variables.

From the above Correlation plot we see that, the **Sales variable is highly correlated with the Commision variable**. The **Duration variable is moderately correlated to the Sales and Commision variables**. Other than that, there is **no correlation between other pairs of variables**.

2.2 Data Split: Split the data into test and train, build classification model CART, Random Forest, Artificial Neural Network

Converting Object variables to Categorical variables.

The object variables in the dataset were converted to integers using label encoding because most of the machine learning models require that the data be of numeric type.

Splitting the data into train and test sets.

The data was then split into the training and test sets. This is required because we have to make sure that the model generalizes well to unseen data. Therefore, we train the model using the training set and then evaluate it on the test set**. For small datasets, the size of the test set is taken to be 20-30% of the dataset. Here we have taken 30% of the data in the test set.** Also, the stratify argument is used to make sure that the proportions of labels in the dependent variable is the same in both the training as well as the test set.

Scaling the data.

As the **variables in the Insurance dataset have different scales, we also scale the data** using StandardScaler function. **Scaling is not required for the CART and Random Forest models but it is necessary for Artificial Neural Networks because it is a weight-based model**.

	Age	Agency_Code	Type	Commision	Channel	Duration	Sales	Product Name	Destination
0	-1.786195	-1.287365	-1.239448	-0.356793	0.133922	-0.208457	-0.522459	1.840389	-0.442595
1	0.072811	0.721901	0.806811	-0.587070	0.133922	-0.345415	-0.687749	-0.525343	-0.442595
2	-0.206040	0.721901	0.806811	-0.587070	0.133922	-0.222153	-0.029332	0.263235	-0.442595
3	2.117717	-1.287365	-1.239448	0.260857	0.133922	0.010676	0.361603	1.840389	-0.442595
4	0.909364	-1.287365	-1.239448	-0.557567	0.133922	0.599597	-0.814494	0.263235	-0.442595

Table 9: Training set after pre-processing

CART Model

Best parameters found for the DecisionTreeClassifier using GridSearchCV after trial and error are:

```
{'max_depth': 4, 'min_samples_leaf': 9, 'min_samples_split': 25}
```

Here, the **max_depth** parameter determines the depth of the tree built by the model, the **min_samples_leaf** parameter means the minimum number of samples that must be present in the leaf nodes and the **min_samples_split** parameter means the minimum number of samples that must be present in the decision nodes for splitting.

	Imp
Agency_Code	0.572302
Sales	0.231088
Product Name	0.095234
Commision	0.058411
Duration	0.042965
Age	0.000000
Type	0.000000
Channel	0.000000
Destination	0.000000

Table 10: Feature importance for the CART model

From the above table we see that, the **Agency_Code variable has the highest importance followed by the Sales variable** in predicting the claimed status of a customer. All **other variables have very low or no importance**, meaning they are not very good predictors of the dependent variable.

Random Forest Model

Best parameters found for the RandomForestClassifier using GridSearchCV after trial and error are:

```
{'max_depth': 7, 'max_features': 3, 'min_samples_leaf': 30, 'min_samples_split': 90,
'n_estimators': 101}
```

Here, the **max_features** parameter means the maximum number of features to use for a tree and the **n_estimators** parameter determines the number of trees to build in the random forest. The remaining parameters are the same as CART model.

	Imp
Agency_Code	0.331945
Product Name	0.199437
Commision	0.155153
Sales	0.150917
Duration	0.068495
Type	0.052348
Age	0.029196
Destination	0.012509
Channel	0.000000

Table 11: Feature importance for the Random Forest model

From the above table we see that, the **Agency_Code variable has the highest importance followed by the Product Name, Commision and Sales variables** in predicting the claimed status of a customer. All **other variables have very low importance**, meaning they are not very good predictors of the dependent variable. The Random Forest model gives importance to more variables than the CART model because of bootstrapping.

ANN Model

Best parameters found for the MLPClassifier using GridSearchCV after trial and error are:

```
{'hidden_layer_sizes': 400, 'max_iter': 300, 'tol': 0.0003}
```

Here, the **hidden_layer_sizes** parameter determines the number of neurons in a hidden layer as well as the number of hidden layers in the ANN. The **max_iter** parameter determines the maximum number of iterations allowed for updating the weights. The **tol** parameter is the tolerance for the loss function for consecutive iterations.

2.3 Performance Metrics: Comment and Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score, classification reports for each model.

Performance Metrics for CART Model.

Accuracy Score for Training and Testing

```
Training Accuracy score for the CART model: 0.797
Testing Accuracy score for the CART model: 0.790
```

From the above output we see that for the **CART model, the training and testing accuracies are very close to each other**. Testing score is slightly lower than the training score, therefore **the model has very slightly overfitted the data**.

Confusion Matrix for Training and Testing

```
Confusion matrix of the train set for the CART model:
      0    1
0  1260  193
1   234  413
```

From the above confusion matrix, we see that out of the total 647 positive examples in the training data, 413 were classified correctly while the other 234 examples were classified as negative. Here the False Negatives are greater than the False Positives, hence we can say that the **Recall is less than the Precision**.

```
Confusion matrix of the test set for the CART model:
     0    1
0  544   79
1  110  167
```

From the above confusion matrix, we see that out of the total 277 positive examples in the testing data, 167 were classified correctly while the other 110 examples were classified as negative. Here the False Negatives are greater than the False Positives, hence we can say that the **Recall is less than the Precision**.

ROC AUC Score and the ROC curves for Training and Testing

```
ROC AUC score of train data for the CART model: 0.836
ROC AUC score of test data for the CART model: 0.781
```

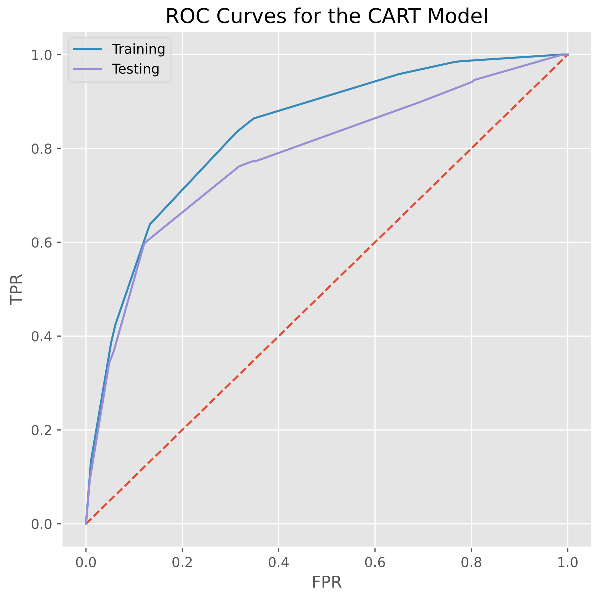

Figure 26: ROC curves for the CART model

From the above output we can see that, the AUC score for the train data is more than the score for the test data; this is also an indicator of slight overfitting. An ideal classifier has the ROC curve closer to top-left corner. The ROC curve for the CART model is far above the diagonal line and near the top-left corner, hence it is a good classifier for the given data.

Classification Reports for Training and Testing

Classification report of the train data for the CART model:

	precision	recall	f1-score	support
0	0.84	0.87	0.86	1453
1	0.68	0.64	0.66	647
accuracy			0.80	2100
macro avg	0.76	0.75	0.76	2100
weighted avg	0.79	0.80	0.79	2100

From the above output we see that, the precision is slightly greater than the recall for the positive class in the training data. Here, it is important that the recall should be large because we need the model to correctly predict as many examples as possible which are actually positive i.e., claimed the insurance. Also, the f1-score value is not very high, hence the overall performance of the model is moderate.

Classification report of the test data for the CART model:

	precision	recall	f1-score	support
0	0.83	0.87	0.85	623
1	0.68	0.60	0.64	277
accuracy			0.79	900
macro avg	0.76	0.74	0.75	900
weighted avg	0.78	0.79	0.79	900

For the test data, the precision is same as that of the training data but the recall has decreased to 0.60. F1-score is very slightly lower than that of the training data. Therefore, we can say that the model generalizes well to unseen data.

Performance Metrics for Random Forest Model.

Accuracy Score for Training and Testing

```
Training Accuracy score for the Random Forest model: 0.794
Testing Accuracy score for the Random Forest model: 0.782
```

From the above output we see that for the **Random Forest model, the training and testing accuracies are close to each other**. Testing score is slightly lower than the training score, therefore **the model has slightly overfitted the data**.

Confusion Matrix for Training and Testing

```
Confusion matrix of the train set for the Random Forest model:
      0    1
0  1301  152
1   280  367
```

From the above confusion matrix, we see that out of the total 647 positive examples in the training data, 367 were classified correctly while the other 280 examples were classified as negative. Here the False Negatives are greater than the False Positives, hence we can say that the **Recall is less than the Precision**.

```
Confusion matrix of the test set for the Random Forest model:
     0    1
0  562   61
1  135  142
```

From the above confusion matrix, we see that out of the total 277 positive examples in the testing data, 142 were classified correctly while the other 135 examples were classified as negative. Here the False Negatives are greater than the False Positives, hence we can say that the **Recall is less than the Precision**.

ROC AUC Score and the ROC curves for Training and Testing

```
ROC AUC score of train data for the Random Forest model: 0.846
ROC AUC score of test data for the Random Forest model: 0.802
```

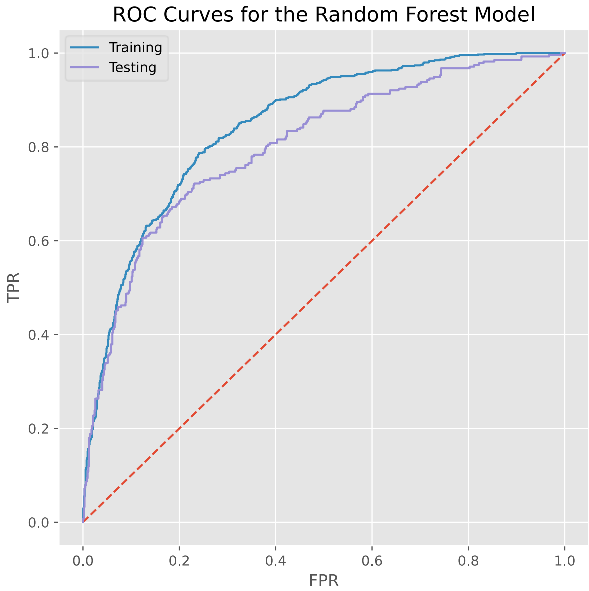

Figure 27: ROC Curves for the Random Forest Model

From the above output we can see that, the AUC score for the train data is more than the score for the test data; this is also an indicator of overfitting. The ROC curve for the Random Forest model is far above the diagonal line and near the top-left corner, hence it is a good classifier for the given data.

Classification Reports for Training and Testing

```
Classification report of the train data for the Random Forest model:
              precision    recall  f1-score   support

           0       0.82      0.90      0.86      1453
           1       0.71      0.57      0.63       647

    accuracy                           0.79      2100
   macro avg       0.77      0.73      0.74      2100
weighted avg       0.79      0.79      0.79      2100
```

From the above output we see that, the precision is greater than the recall for the positive class in the training data. Here, it is important that the recall should be large because we need the model to correctly predict as many examples as possible which are actually positive i.e., claimed the insurance. Also, the f1-score value is not very high, hence the overall performance of the model is moderate.

```
Classification report of the test data for the Random Forest model:
              precision    recall  f1-score   support

           0       0.81      0.90      0.85       623
           1       0.70      0.51      0.59       277

    accuracy                           0.78       900
   macro avg       0.75      0.71      0.72       900
weighted avg       0.77      0.78      0.77       900
```

For the test data, the precision is almost same as that of the training data but the recall has decreased to 0.51. F1-score is slightly lower than that of the training data. Therefore, we can say that the model generalizes well to unseen data.

Performance Metrics for ANN Model.

Accuracy Score for Training and Testing

```
Training Accuracy score for the ANN model: 0.812
Testing Accuracy score for the ANN model: 0.773
```

From the above output we see that for the **ANN model, the training and testing accuracies are different from each other**. Testing score is lower than the training score, therefore **the model has overfitted the data**.

Confusion Matrix for Training and Testing

```
Confusion matrix of the train set for the ANN model:
      0    1
0  1291  162
1   233  414
```

From the above confusion matrix, we see that out of the total 647 positive examples in the training data, 414 were classified correctly while the other 233 examples were classified as negative. Here the False Negatives are greater than the False Positives, hence we can say that the **Recall is less than the Precision**.

```
Confusion matrix of the test set for the ANN model:
     0    1
0  555   68
1  136  141
```

From the above confusion matrix, we see that out of the total 277 positive examples in the testing data, 141 were classified correctly while the other 136 examples were classified as negative. Here the False Negatives are greater than the False Positives, hence we can say that the **Recall is less than the Precision**.

ROC AUC Score and the ROC curves for Training and Testing

```
ROC AUC score of train data for the ANN model: 0.869
ROC AUC score of test data for the ANN model: 0.797
```

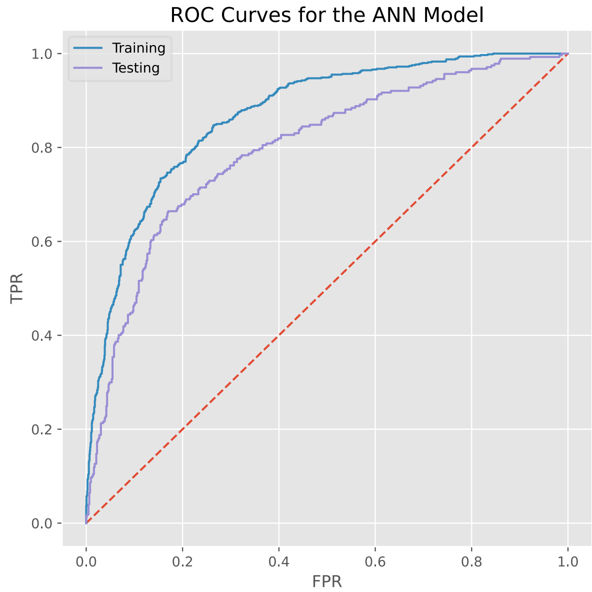

Figure 28: ROC Curves for the ANN Model

From the above output we can see that, the AUC score for the train data is more than the score for the test data; this is also an indicator of overfitting. The ROC curve for the ANN model is far above the diagonal line and near the top-left corner, hence it is a good classifier for the given data. The gap between the training and testing curve larger than the other models.

Classification Reports for Training and Testing

```
Classification report of the train data for the ANN model:
              precision    recall  f1-score   support

           0       0.85      0.89      0.87      1453
           1       0.72      0.64      0.68       647

    accuracy                           0.81      2100
   macro avg       0.78      0.76      0.77      2100
weighted avg       0.81      0.81      0.81      2100
```

From the above output we see that, the precision is greater than the recall for the positive class in the training data. Here, it is important that the recall should be large because we need the model to correctly predict as many examples as possible which are actually positive i.e., claimed the insurance. Also, the f1-score value is not very high, hence the overall performance of the model is moderate.

```
Classification report of the test data for the ANN model:
              precision    recall  f1-score   support

           0       0.80      0.89      0.84       623
           1       0.67      0.51      0.58       277

    accuracy                           0.77       900
   macro avg       0.74      0.70      0.71       900
weighted avg       0.76      0.77      0.76       900
```

For the test data, the precision has reduced to 0.67 and the recall has decreased to 0.51. F1-score is lower than that of the training data. Therefore, we can say that the model does not generalizes well to unseen data. This can be resolved by acquiring more data samples.

2.4 Final Model: Compare all the models and write an inference which model is best/optimized.

	Accuracy	Precision	Recall	F1-score	AUC-score
CART Model	0.797	0.682	0.638	0.659	0.836
Random Forest Model	0.794	0.707	0.567	0.630	0.846
ANN Model	0.812	0.719	0.640	0.677	0.869

Table 12: Performance Metrics for Training data

	Accuracy	Precision	Recall	F1-score	AUC-score
CART Model	0.790	0.679	0.603	0.639	0.781
Random Forest Model	0.782	0.700	0.513	0.592	0.802
ANN Model	0.773	0.675	0.509	0.580	0.797

Table 13: Performance Metrics for Testing data

From the above two tables we can see that:

1. The ANN model has the highest accuracy on the training data but lowest accuracy on the test data. The CART and Random Forest model have both the accuracies close to each other. Hence, they have better performance compared to ANN model with respect to accuracy.
2. The Precision for the Random Forest model is most consistent for both training and testing. The ANN model has the highest precision on the training data but lowest precision on the test data.
3. The CART model has the most consistent recall for both the training and testing data. ANN model has the highest recall for training but lowest on testing data.
4. The difference between the f1-score for training and testing for the ANN model is large even though it has the highest score for training data. Again, the CART model has the most consistent f1-score for both the training and testing data.
5. The AUC scores for all the models are high. But the difference between the AUC score for training and testing for the ANN model is large indicating overfitting.

As the recall is the most important metric for this case study and also considering all the other metrics, the CART and the Random Forest model are the best options for model selection. **Random Forest model has the highest AUC score on the test data and it will also exhibit less variation on unseen data** as it is an ensemble method compared to the CART and ANN model. Hence, **Random Forest is the best model for this particular case study.**

2.5 Inference: Based on the whole Analysis, what are the business insights and recommendations

For the business problem of a Tour Insurance firm, we have built various models for predicting whether a customer will claim their insurance or not. Models like CART, Random Forest and ANN were evaluated on the training and test datasets and their performances were compared using various performance metrics like Accuracy, Precision, Recall, AUC score, etc.

After comparing the models, it was seen that all of them have performed well on the given data. The ANN model highly overfitted the data. The CART and the Random Forest model performed almost similarly. But as the Random Forest model is an ensemble of many CART models, it's predictions will be less variable on new data. Therefore, the Random Forest model was determined to be the best model for the business problem.

The Tour insurance firm can make use of the above models as:

1. For **Fraud detection** – By predicting whether a customer will file for insurance claim or not, the firm can **carefully scrutinize the customers who are going to file for claim and detect fraud beforehand**.

2. By predicting whether a customer will file for insurance claim or not, the **processing of the claims can be made faster and accurate by implementing automation. This will increase customer satisfaction**.
3. **Referral programs can be implemented for the customers who are going to claim the insurance by providing gift vouchers or providing discounts on their next travel for every successful referral**.