Problem 2: CART-RF-ANN

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

Attribute Information:

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

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

Table 1: Head of the dataset showing the first 5 records

	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

The dataset was loaded and the head of the dataset was checked. Table 1 shows the first 5 records of the dataset. From this table, we can see the different variables or columns of the dataset.

Table 2: Information of 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
```

The dataset has 10 columns and 3000 records which is seen in Table 2. There are 3000 non-null records in all the 10 columns meaning there are no missing records based on this initial analysis that was done.

Table 3: Data type of the columns in the dataset

```
Age              int64
Agency_Code     object
Type            object
Claimed         object
Commision      float64
Channel         object
Duration         int64
Sales          float64
Product Name    object
Destination     object
dtype: object
```

From Table 3, it is seen that the variable 'Age' is of integer data type. The data type is float64 for 'Co mmision' and 'Sales' variables and object for all the other variables. There are 9 independent variabl es and one target variable – 'Clamied'.

The shape of the data is (3000, 10) meaning the dataset has 3000 rows and 10 columns.

Table 4: Missing value of the columns in the dataset

Age	0
Agency_Code	0
Type	0
Claimed	0
Commision	0
Channel	0
Duration	0
Sales	0
Product Name	0
Destination	0
dtype: int64	

The dataset was further checked for missing values and it is seen from Table 4 that there are no missing values in the dataset.

Table 5: Description of the dataset

	Age	Commision	Duration	Sales
count	3000.000000	3000.000000	3000.000000	3000.000000
mean	38.091000	14.529203	70.001333	60.249913
std	10.463518	25.481455	134.053313	70.733954
min	8.000000	0.000000	-1.000000	0.000000
25%	32.000000	0.000000	11.000000	20.000000
50%	36.000000	4.630000	26.500000	33.000000
75%	42.000000	17.235000	63.000000	69.000000
max	84.000000	210.210000	4580.000000	539.000000

Table 5 shows the description or the summary of the numerical columns of the dataset. It can be seen that there are 10 different columns in this dataframe and all of them have 3000 values. By looking at Table 5, we are able to deduce that ‘Duration’ has the highest mean value while ‘Commision’ has the lowest mean value. ‘Duration’ has the highest standard deviation value while ‘Age’ has the lowest standard deviation value.

Univariate Analysis:

Numerical variables:

Table 6: Head of the numerical dataset showing the first 5 records

	Age	Commision	Duration	Sales
0	48	0.70	7	2.51
1	36	0.00	34	20.00
2	39	5.94	3	9.90
3	36	0.00	4	26.00
4	33	6.30	53	18.00

Table 6 shows the first 5 records of the numerical dataset. The numerical dataset was used to calculate the skewness, plot the univariate distribution and the boxplot.

Table 7: Skewness of the variables of the dataset

	Skewness
Age	1.149138
Commision	3.147283
Duration	13.777788
Sales	2.379958

1. **Age:**

Table 8: Description of 'Age'

```
Description of Age
-----------------------------------------------------
count    3000.000000
mean       38.091000
std        10.463518
min         8.000000
25%        32.000000
50%        36.000000
75%        42.000000
max        84.000000
Name: Age, dtype: float64 Distribution of Age
```

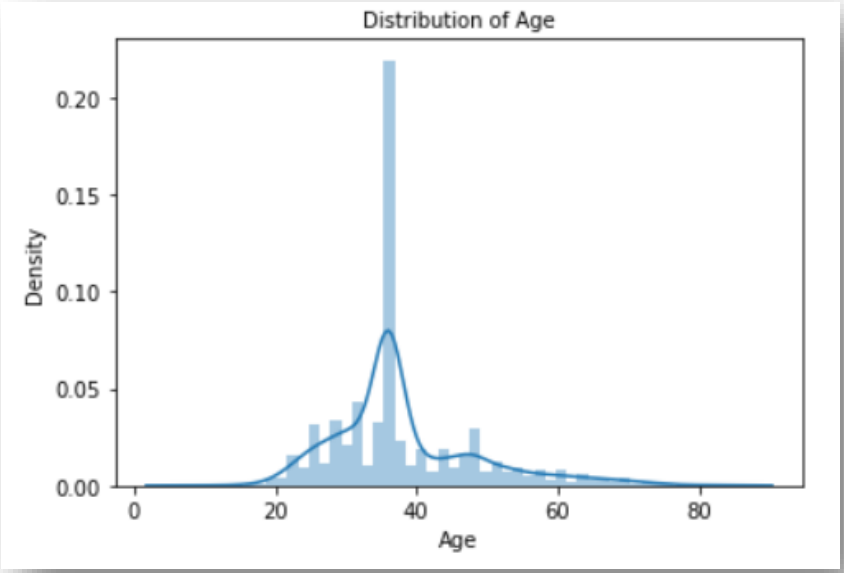

Figure 1: Univariate distribution of 'Age'

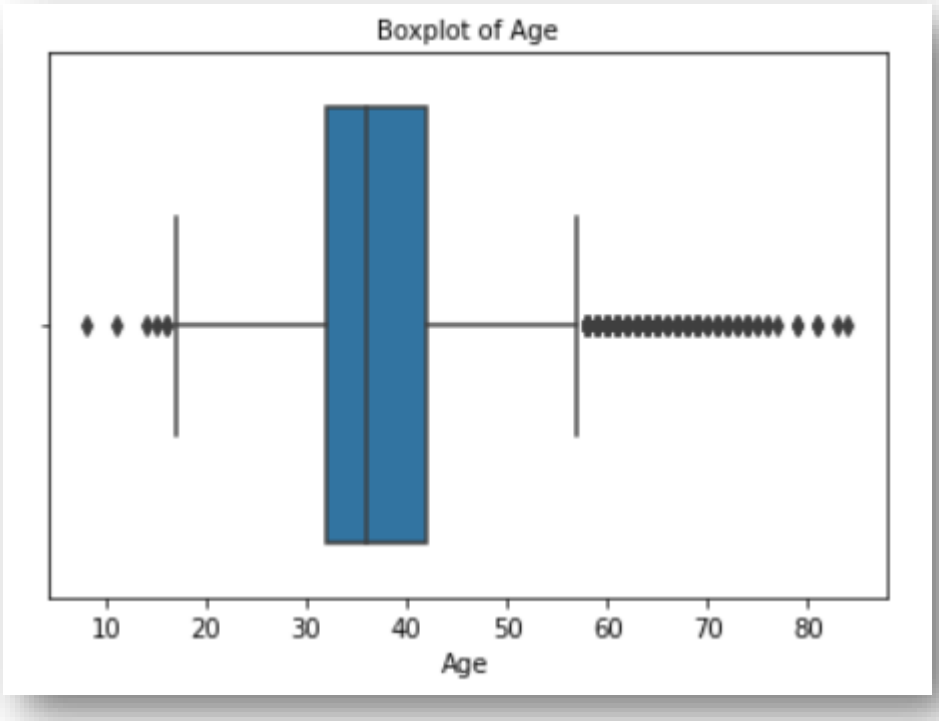

Figure 2: Boxplot showing the distribution of 'Age'

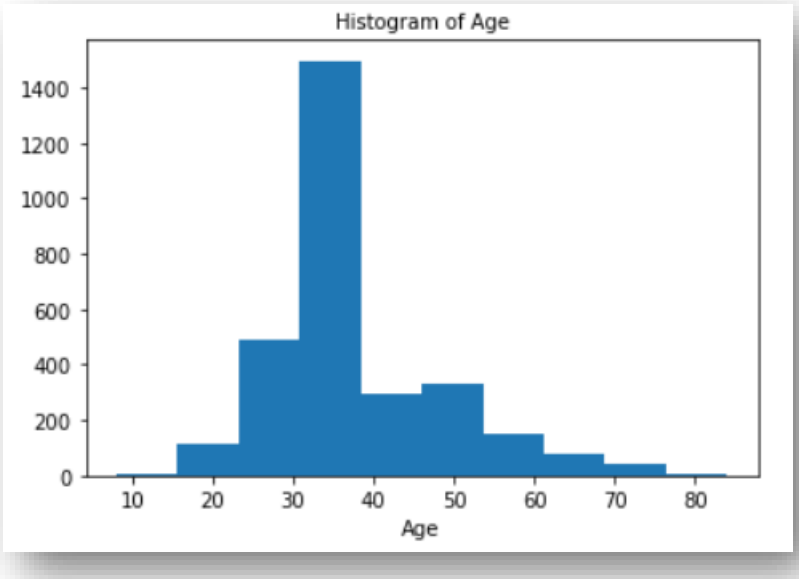

Figure 3: Histogram of 'Age'

Univariate analysis of ‘Age’ is done to understand the patterns and distribution of the data. From Figure 2, we can see that the Box plot of ‘Age’ variable has many outliers. The distribution of the data is moderately right skewed which is seen in Figure 1. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘Age’ variable is 1.149138. From Table 8, it is seen that the mean of the data is 38.09 meaning many people were of the age 38. The minimum age is 8 and the maximum age is 84.

2. Commision:

Table 9: Description of 'Commision'

```
Description of Commision
------------------------------------------------------------
count    3000.000000
mean       14.529203
std        25.481455
min         0.000000
25%         0.000000
50%         4.630000
75%        17.235000
max       210.210000
Name: Commision, dtype: float64 Distribution of Commision
```

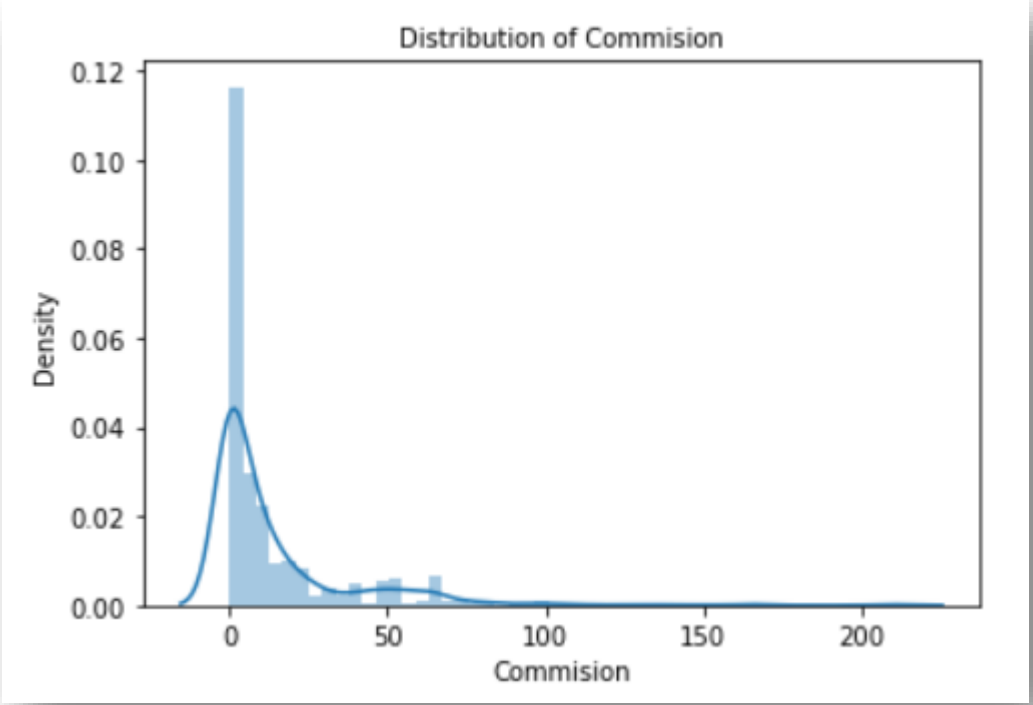

Figure 4: Univariate distribution of 'Commision'

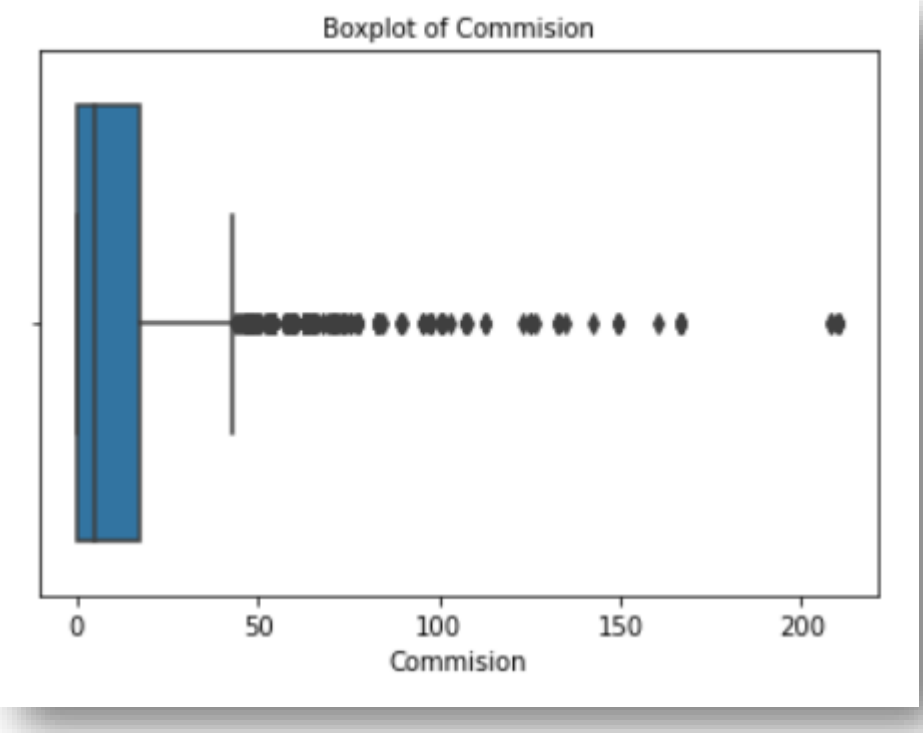

Figure 5: Boxplot showing the distribution of 'Commision'

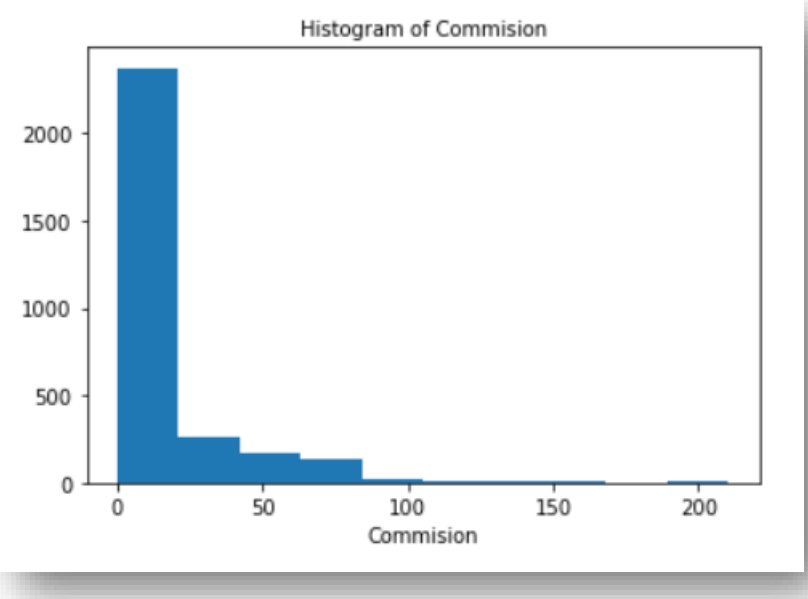

Figure 6: Histogram of 'Commision'

Univariate analysis of ‘Commision’ is done to understand the patterns and distribution of the data. From Figure 5, we can see that the Box plot of ‘Commision’ variable has many outliers. The distribution of the data is right skewed which is seen in Figure 4. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘Commision’ variable is 3.147283. From Table

9, it is seen that the mean of the data is 14.53 meaning the commission received for tour insurance firm is 14.53% on average. The minimum commission received is 0%.

3. Duration:

Table 10: Description of 'Duration'

```
Description of Duration
-----------------------------------------------------------
count    3000.000000
mean       70.001333
std       134.053313
min        -1.000000
25%        11.000000
50%        26.500000
75%        63.000000
max      4580.000000
Name: Duration, dtype: float64 Distribution of Duration
```

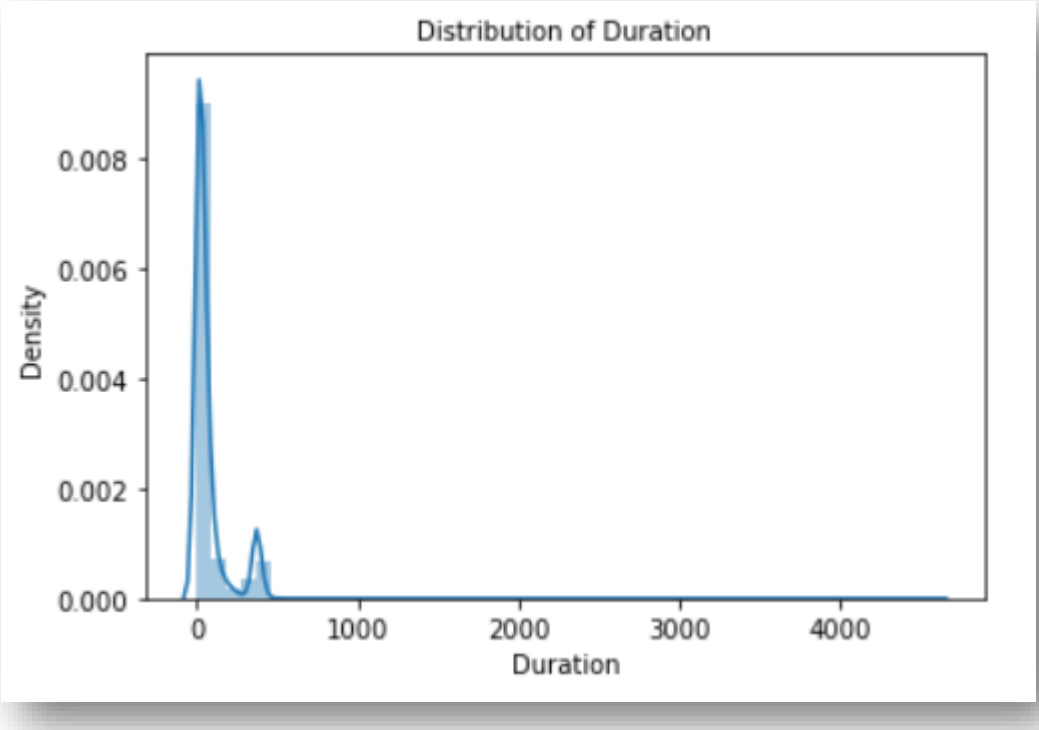

Figure 7: Univariate distribution of 'Duration'

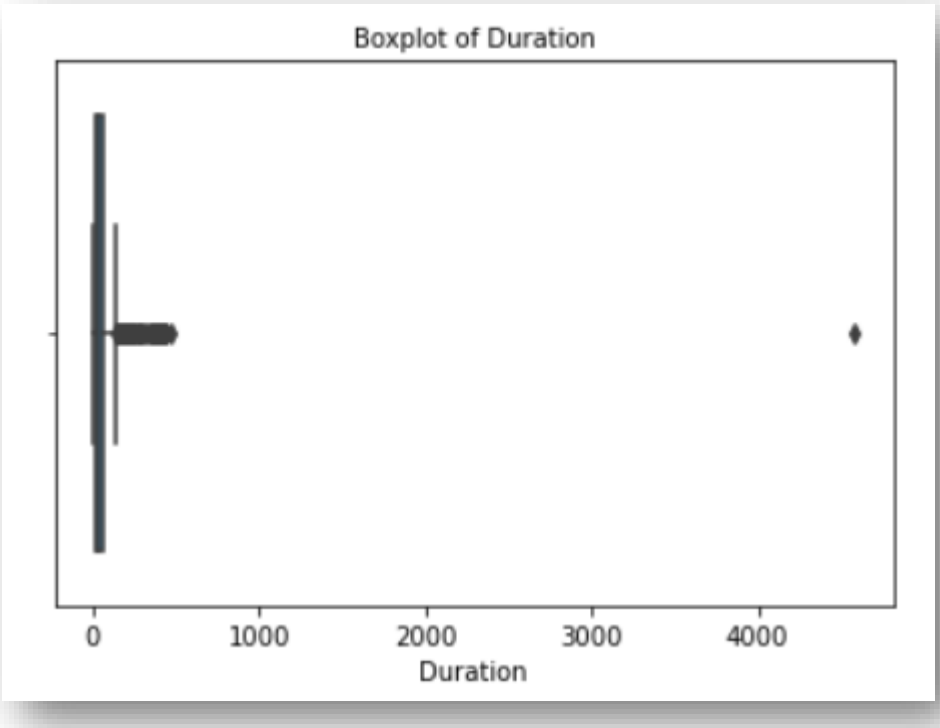

Figure 8: Boxplot showing the distribution of 'Duration'

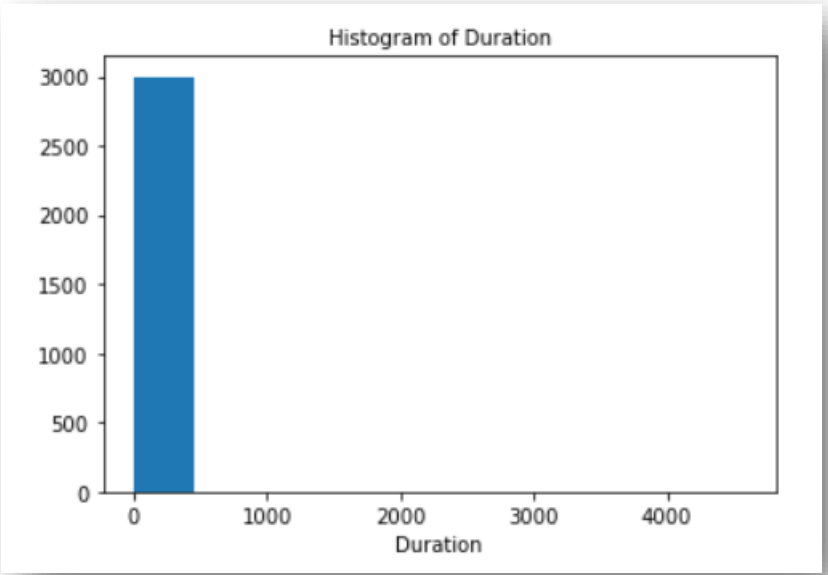

Figure 9: Histogram of 'Duration'

Univariate analysis of ‘Duration’ is done to understand the patterns and distribution of the data. From Figure 8, we can see that the Box plot of ‘Duration’ variable has many outliers. The distribution of the data is right skewed which is seen in Figure 7. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘Duration’ variable is 13.777788. From Table 10, it is seen that the mean of the data is 70.00 meaning the duration of the tour is 70 days on average.

4. Sales:

Table 11: Description of 'Sales'

```
Description of Sales
-----------------------------------------------------
count    3000.000000
mean       60.249913
std        70.733954
min         0.000000
25%        20.000000
50%        33.000000
75%        69.000000
max       539.000000
Name: Sales, dtype: float64 Distribution of Sales
```

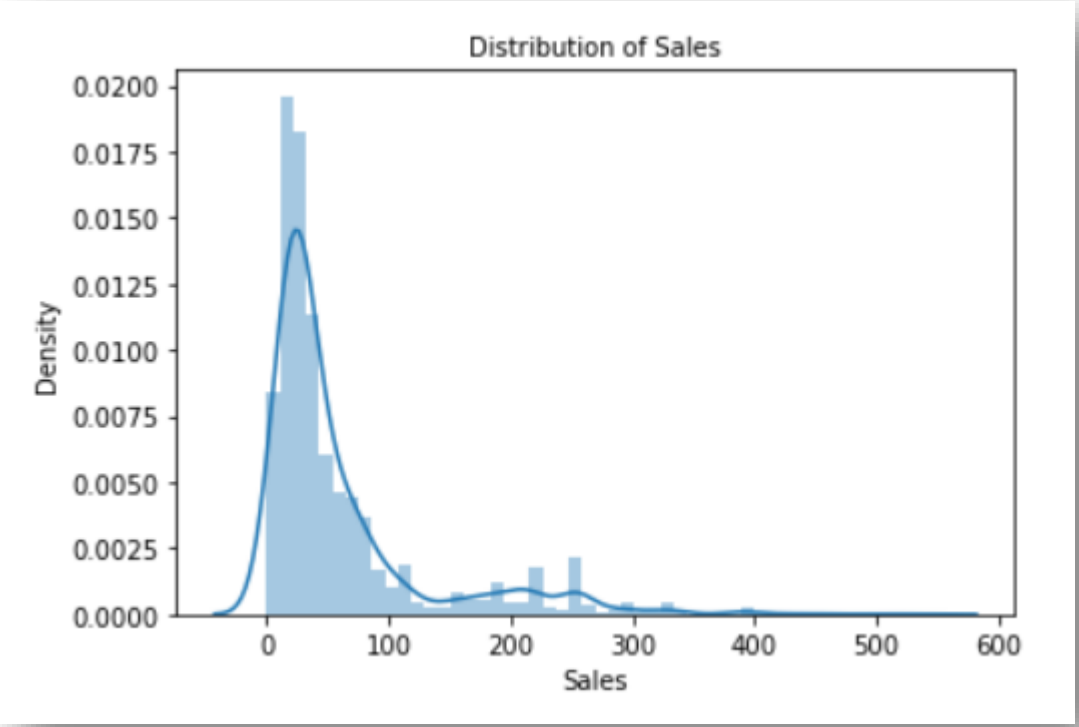

Figure 10: Univariate distribution of 'Sales'

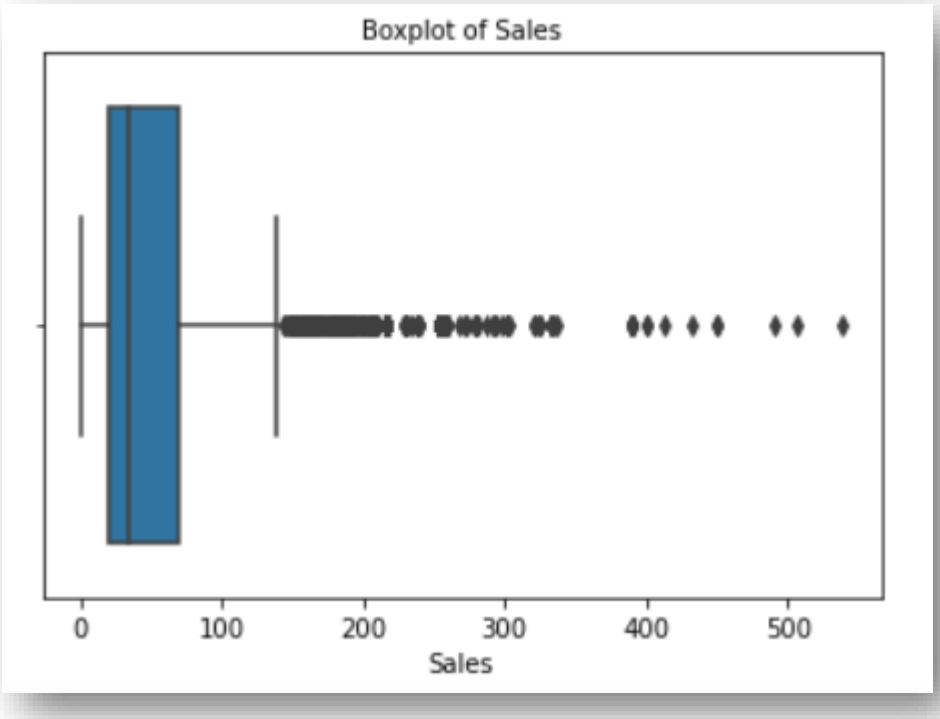

Figure 11: Boxplot showing the distribution of 'Sales'

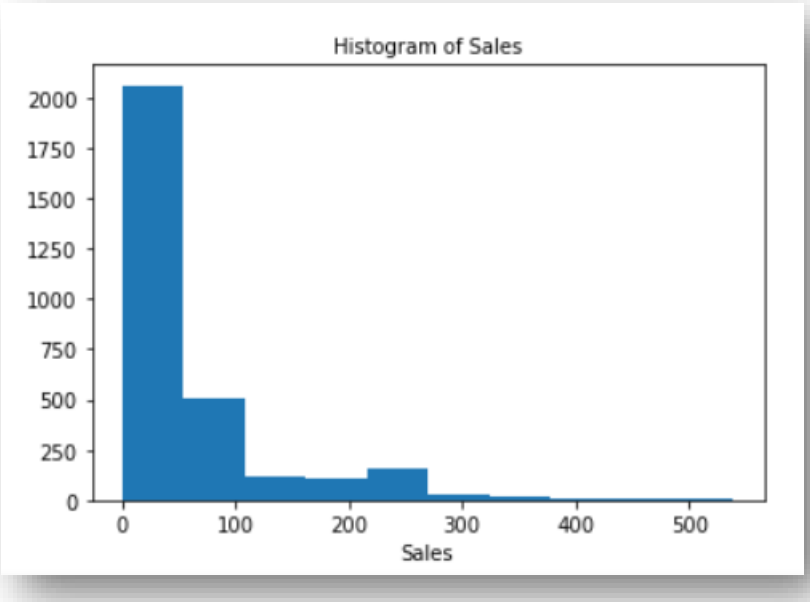

Figure 12: Histogram of 'Sales'

Univariate analysis of ‘Sales’ is done to understand the patterns and distribution of the data. From Figure 11, we can see that the Box plot of ‘Sales’ variable has many outliers. The distribution of the data is moderately right skewed which is seen in Figure 10. This is also seen in Table 7 where the skewness values are given. The skewness value of ‘Sales’ variable is 2.379958. From Table 11, it is

seen that the mean of the data is 60.25 meaning the amount worth of sales per customer in procuring tour insurance policies is 6025 rupees (the data is in 100's) on average.

Categorical variables:

5. **Agency Code:**

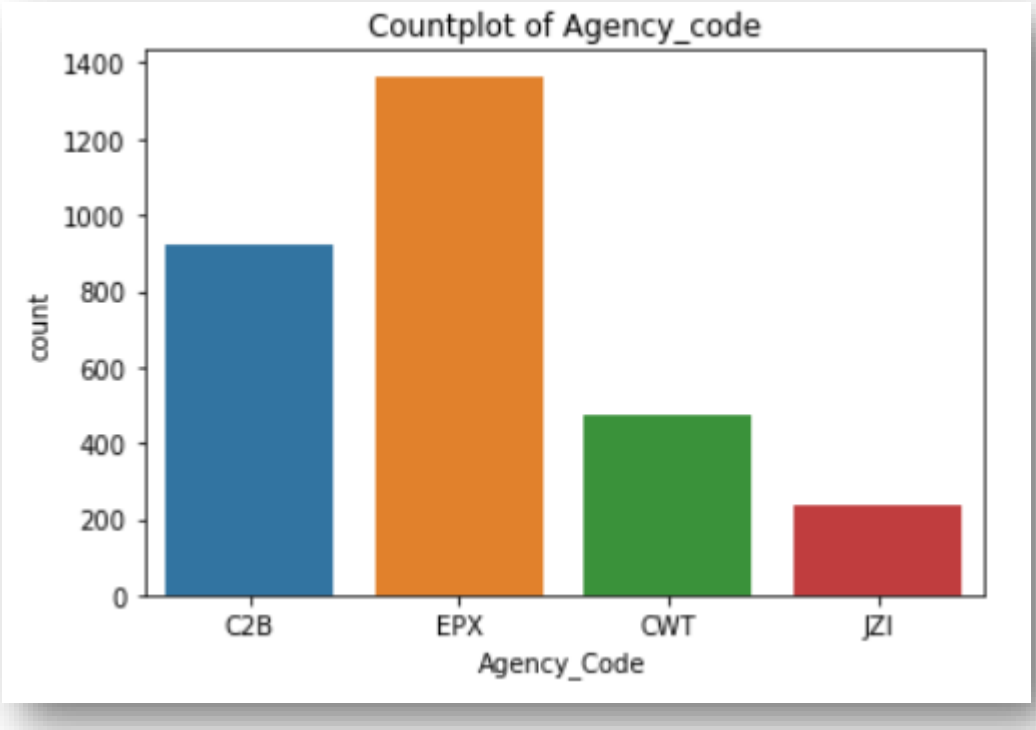

Figure 13: Countplot of 'Agency_Code'

Univariate analysis of 'Agency_Code' is done to understand the patterns and distribution of the data. From Figure 13, we can interpret that majority of the 'Agency_Code' is 'EPX'.

6. Type:

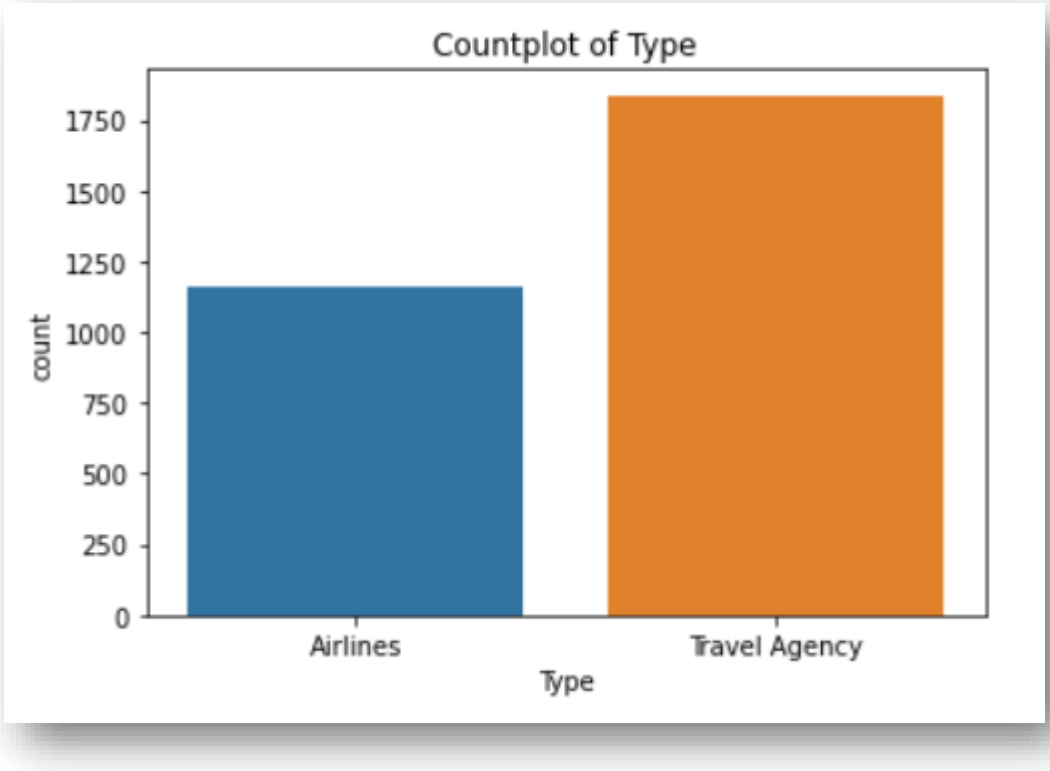

Figure 14: Countplot of 'Type'

Univariate analysis of ‘Type’ is done to understand the patterns and distribution of the data. From Figure 14, we can interpret that the majority of the type of tour insurance firms is ‘Travel Agency’.

7. **Claimed:**

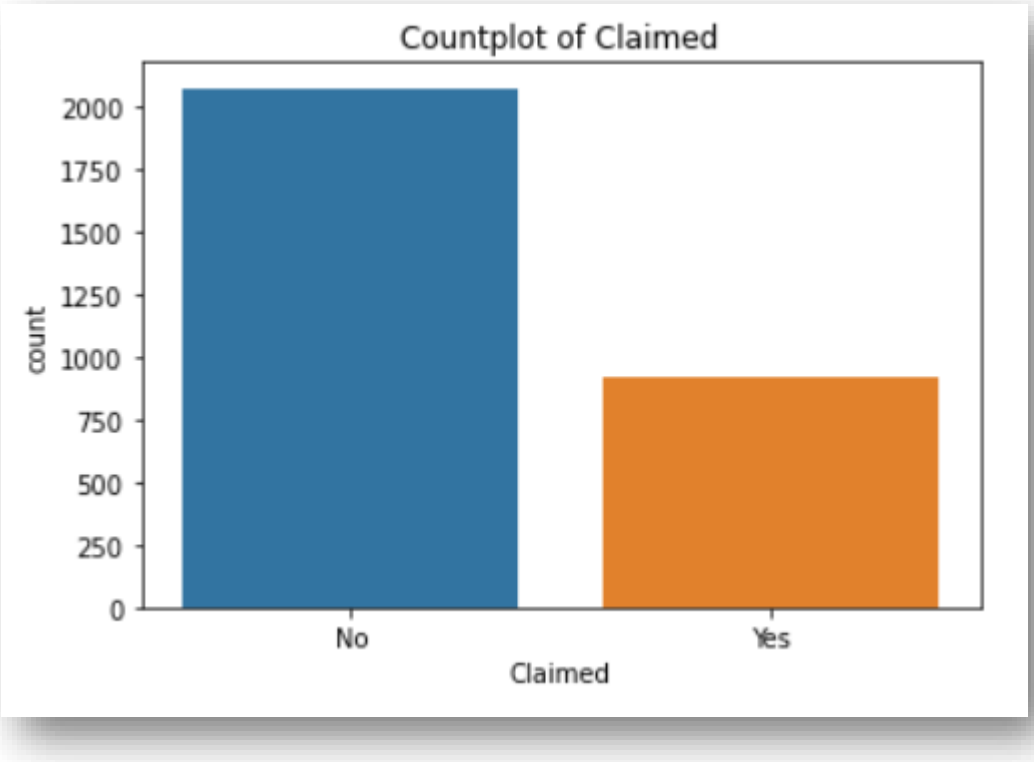

Figure 15: Countplot of 'Claimed'

Univariate analysis of 'Claimed' is done to understand the patterns and distribution of the data. From Figure 15, we can interpret that the majority of the people have not claimed the tour insurance.

8. Channel:

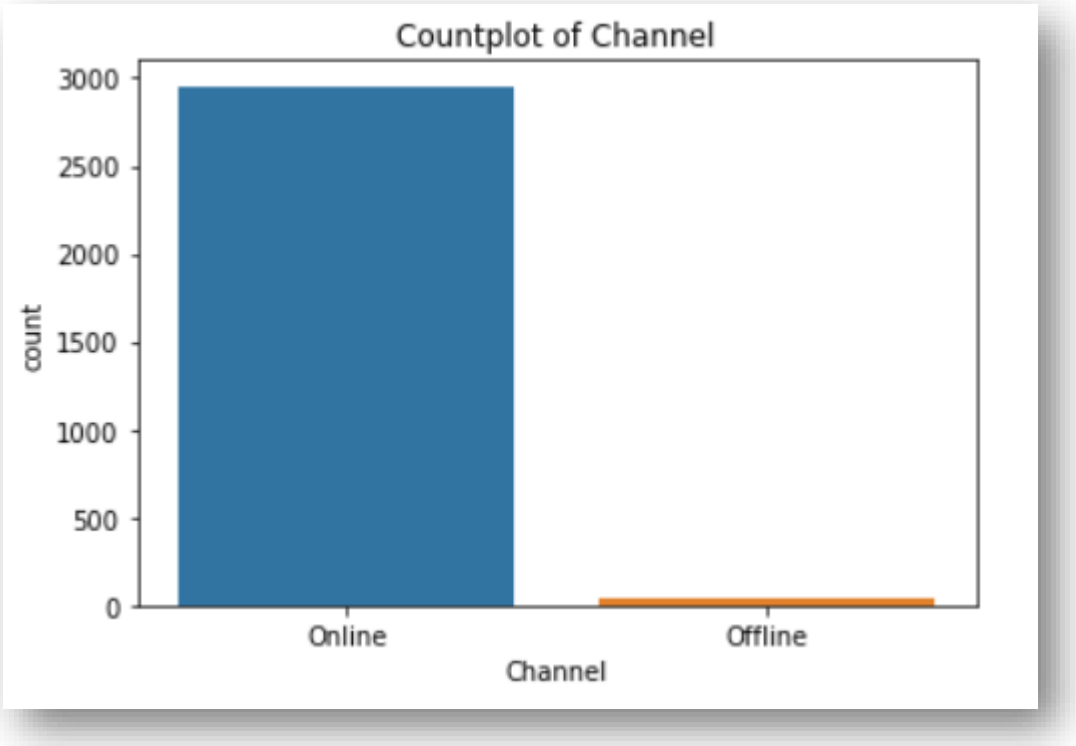

Figure 16: Countplot of 'Channel'

Univariate analysis of ‘Channel’ is done to understand the patterns and distribution of the data. From Figure 16, we can understand that ‘Online’ is the most used distribution channel of tour insurance agencies.

9. Product Name:

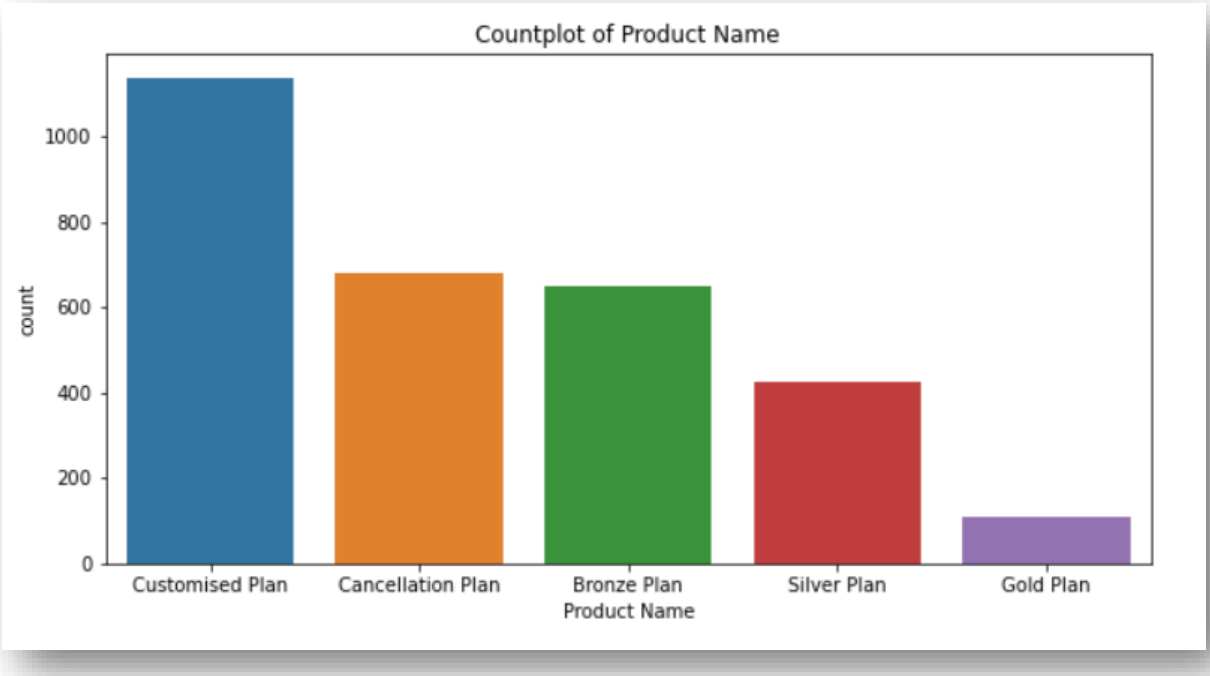

Figure 17: Countplot of 'Product Name'

Univariate analysis of 'Product Name' is done to understand the patterns and distribution of the data. From Figure 17, we can understand that 'Customised Plan' is the most frequently used name of the tour insurance products.

10. Destination:

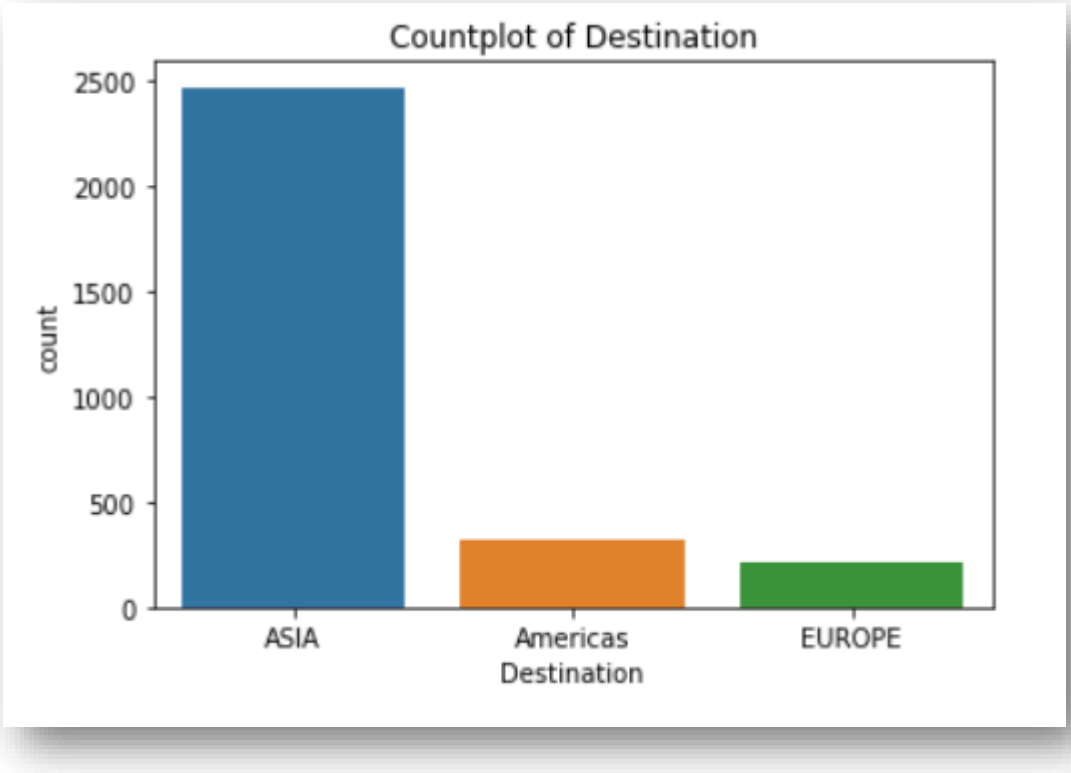

Figure 18: Countplot of 'Destination'

Univariate analysis of ‘Destination’ is done to understand the patterns and distribution of the data. From Figure 18, we can understand that ‘Asia’ tour is the most chosen and common destination.

Bivariate Analysis:

Numerical variables:

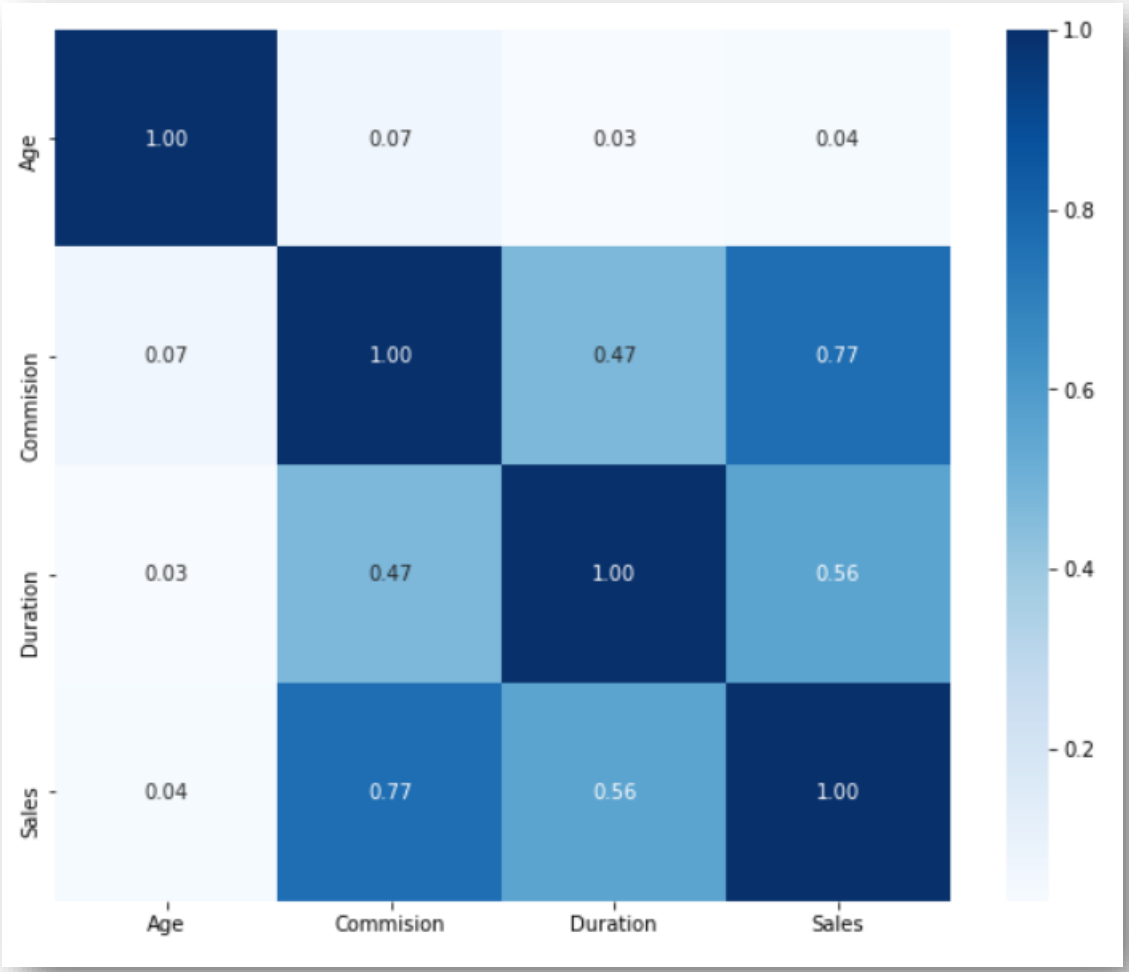

Figure 19: Heat map showing the bivariate analysis of the dataset

Bivariate analysis is done using the help of a heat map. A heat map is used to understand the correlation between two numerical values in a dataset. Figure 19 shows the heat map of the dataset.

Categorical variables:

1. **Agency Code:**

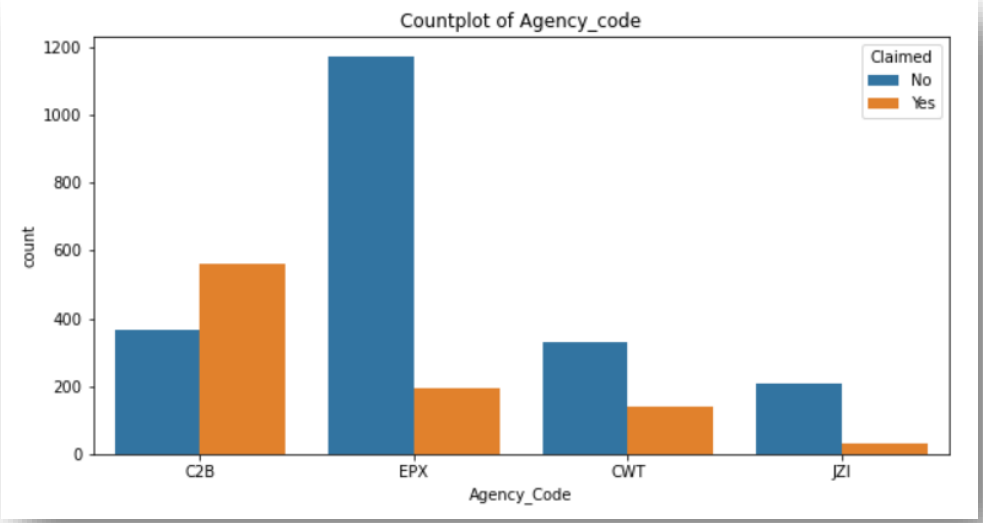

Figure 20: Countplot of 'Agency_Code' and 'Claimed'

From the Figure 20, we can interpret that majority of the 'Agency_code' is 'EPX' and this is true for people who do not claim insurance whereas 'C2B' is the common 'Agency_Code' for people who claim the insurance.

2. **Type:**

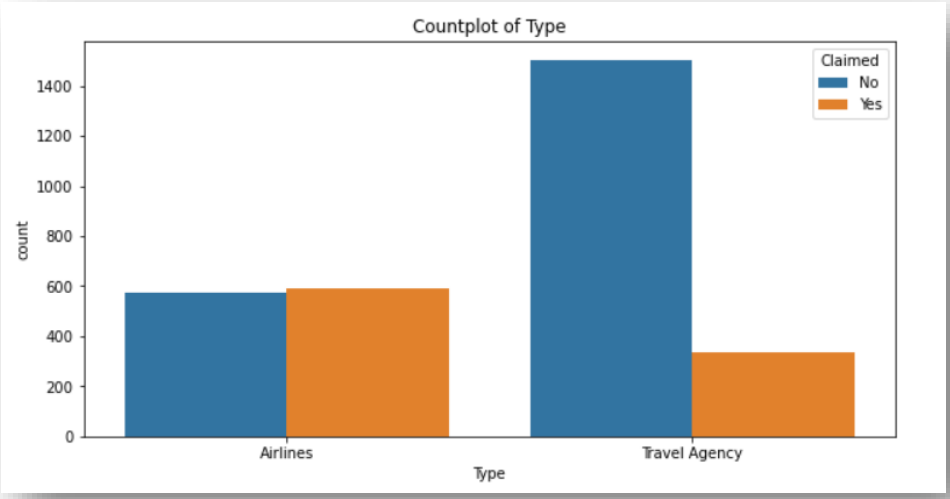

Figure 21: Countplot of 'Type' and 'Claimed'

From the Figure 21, we can interpret that the majority of the type of tour insurance firms is 'Travel Agency' and this is true for people who do not claim insurance.

3. **Channel:**

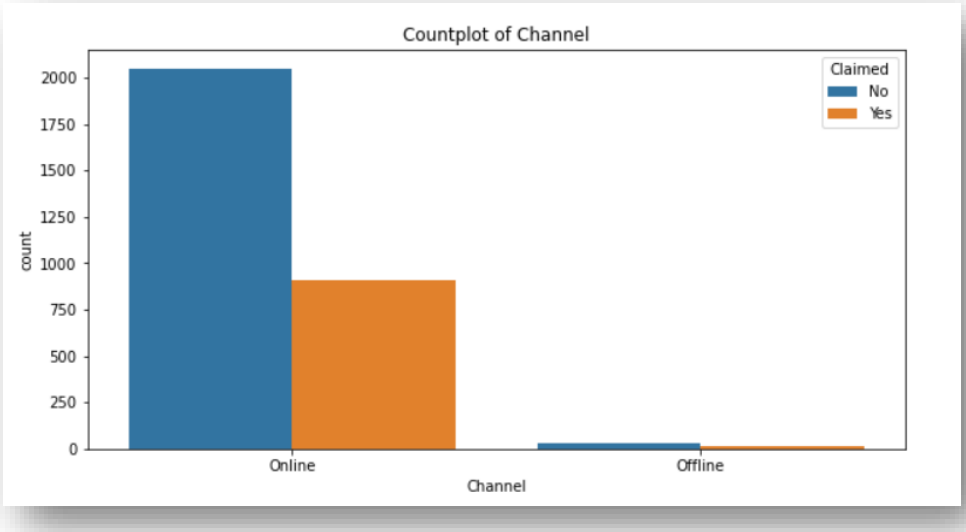

Figure 22: Countplot of 'Channel' and 'Claimed'

From the Figure 22, we can interpret that we can understand that 'Online' is the most used distribution channel of tour insurance agencies and this is true for both, people who do not claim insurance and also people who claim the insurance.

4. **Product Name:**

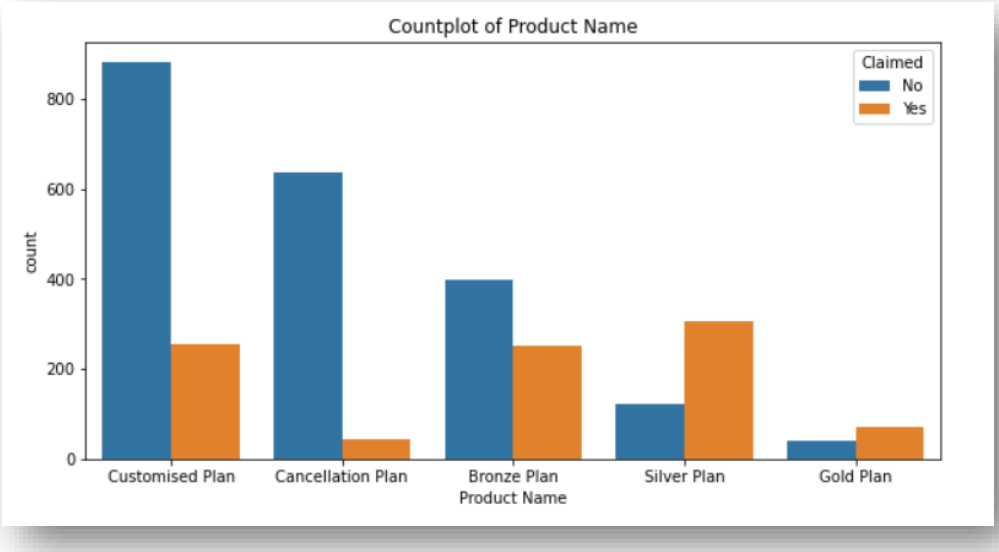

Figure 23: Countplot of 'Product Name' and 'Claimed'

From the Figure 23, we can interpret that we can understand that 'Customised Plan' is the most frequently used name of the tour insurance products and this is true for people who do not claim insurance.

5. Destination:

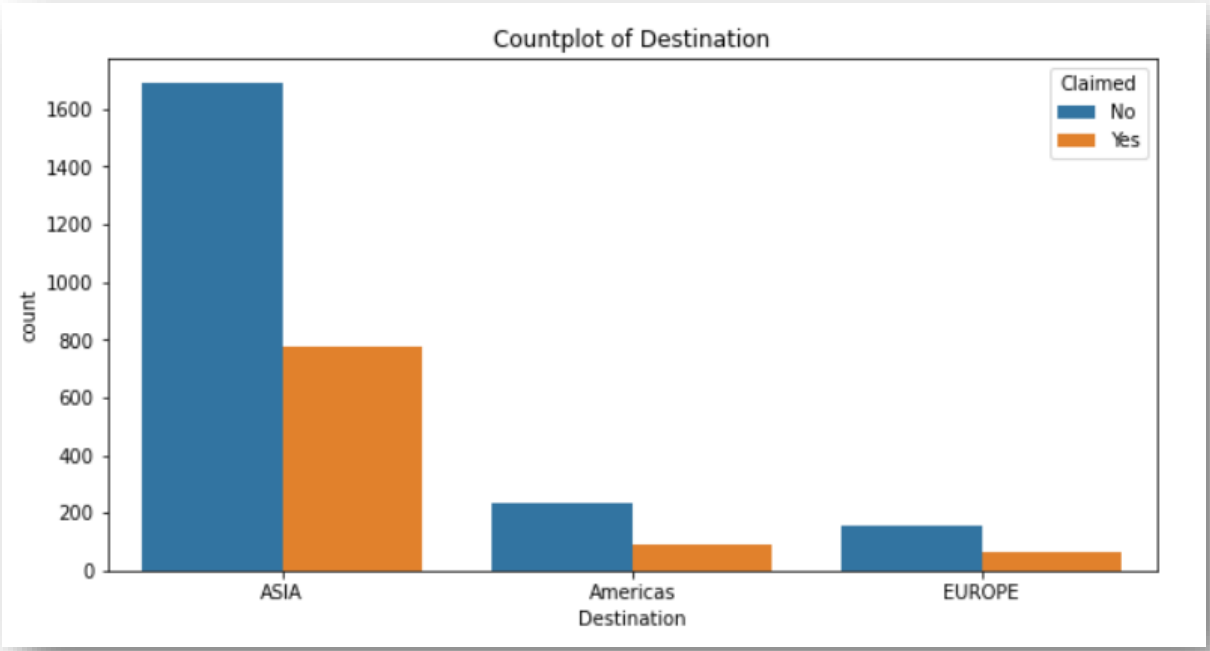

Figure 24: Countplot of 'Destination' and 'Claimed'

From the Figure 24, we can understand that ‘Asia’ tour is the most chosen and common destination and this is true for both, people who do not claim insurance and people who claim the insurance.

Multivariate Analysis:

Numerical variables:

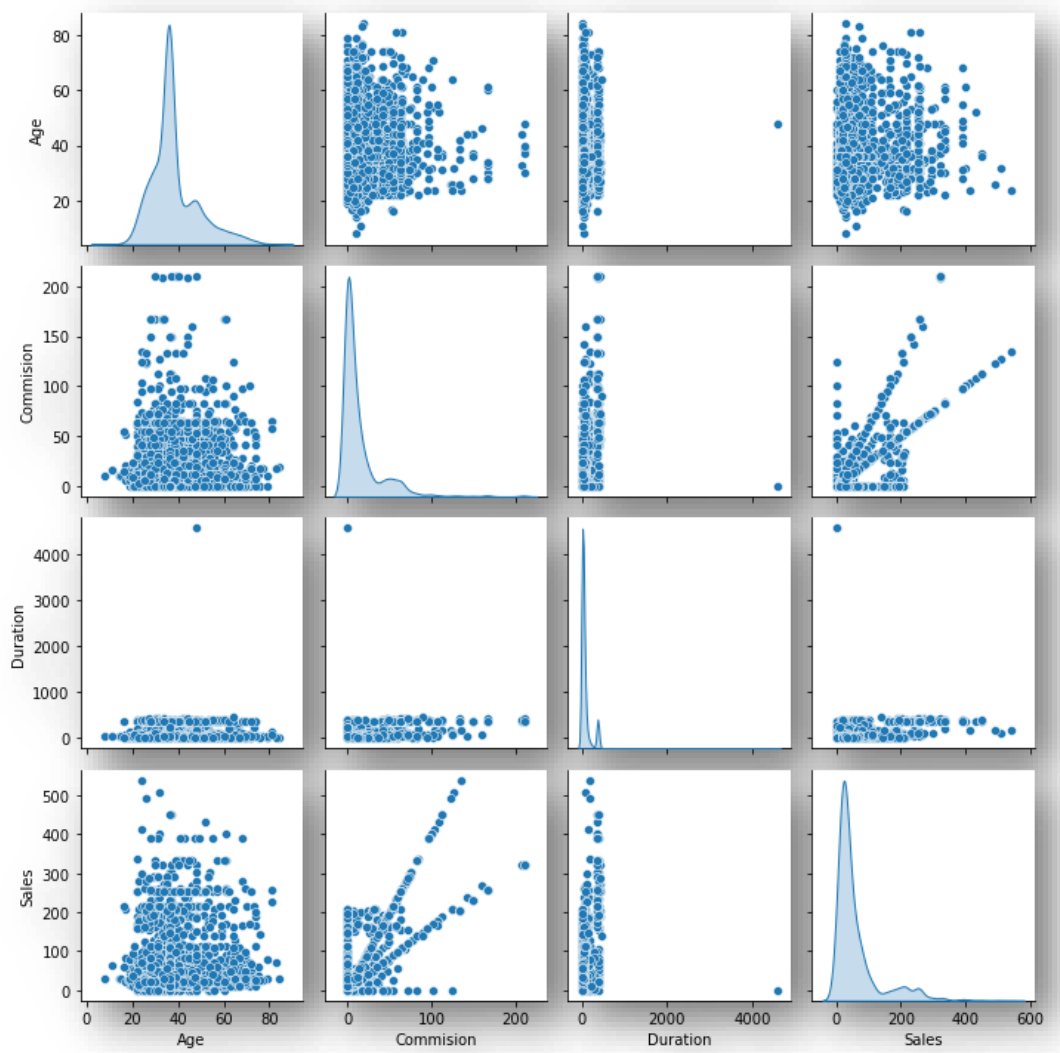

Figure 25: Pair plot showing the multivariate analysis of the numerical variables in the dataset

Multivariate analysis is done using the help of a pair plot to understand the relationship between all the numerical values in the dataset. Pair plot can be used to compare all the variables with each other to understand the patterns or trends in the dataset. Figure 25 shows the pair plot of the dataset.

Observations:

From Figure 19 and Figure 25 we can observe that there are weak correlations between all the numerical variables in the dataset.

Categorical variables:

1. **Agency Code:**

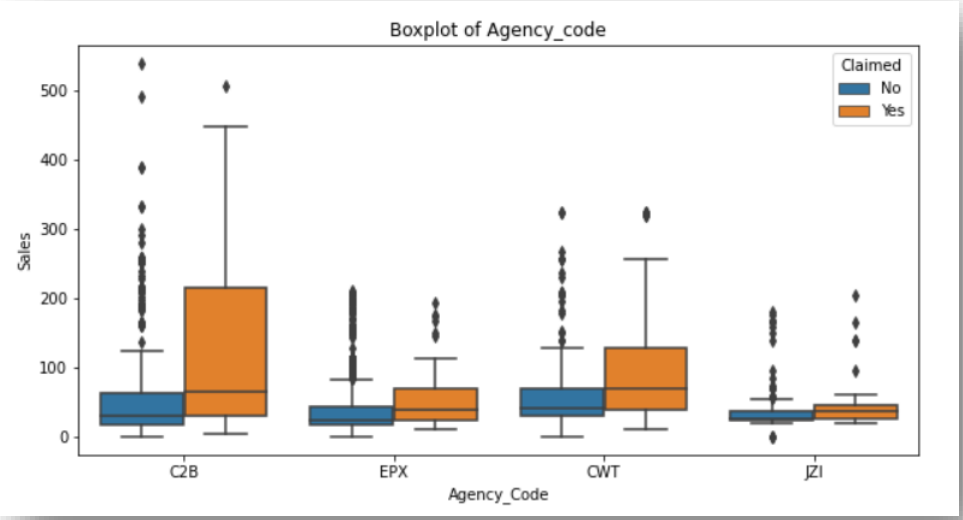

Figure 26: Boxplot of 'Agency_Code'

From Figure 26, we can see that the Box plot of 'Agency_Code' variable has many outliers. The distribution of the data is moderately right skewed or positively skewed which is also seen in Figure 26.

2. **Type:**

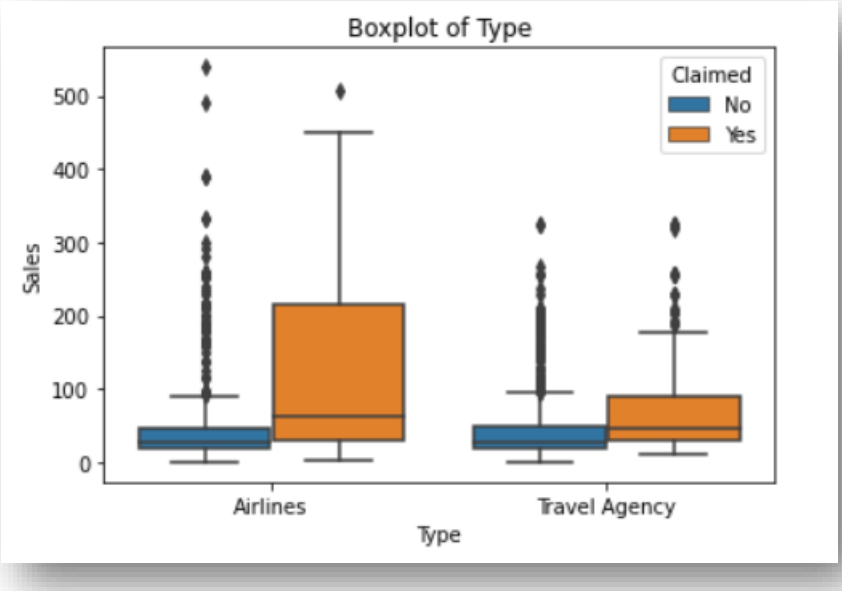

Figure 27: Boxplot of 'Type'

From Figure 27, we can see that the Box plot of 'Type' variable has many outliers. The distribution of the data is moderately right skewed or positively skewed which is also seen in Figure 27.

3. **Claimed:**

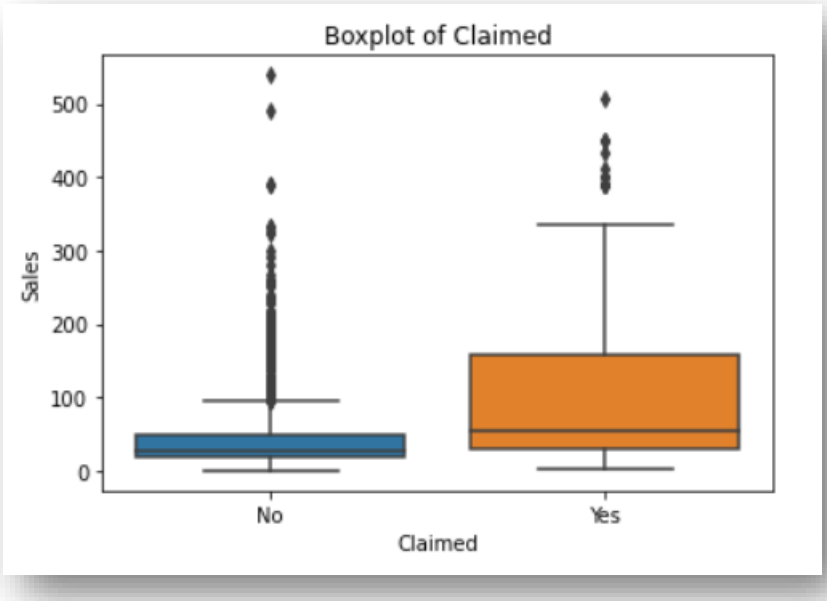

Figure 28: Boxplot of Claimed'

From Figure 28, we can see that the Box plot of 'Claimed' variable has many outliers. The distribution of the data is moderately right skewed or positively skewed which is also seen in Figure 28.

4. **Channel:**

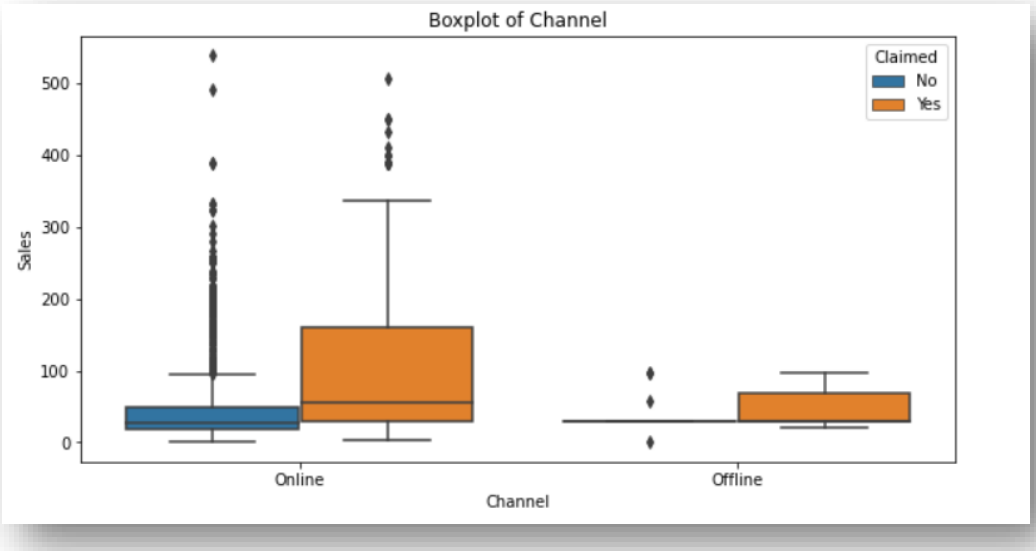

Figure 29: Boxplot of 'Channel'

From Figure 29, we can see that the Box plot of 'Channel' variable has many outliers. The distribution of the data is moderately right skewed or positively skewed which is also seen in Figure 29. Almost all the offline business has a claimed associated.

5. **Product Name:**

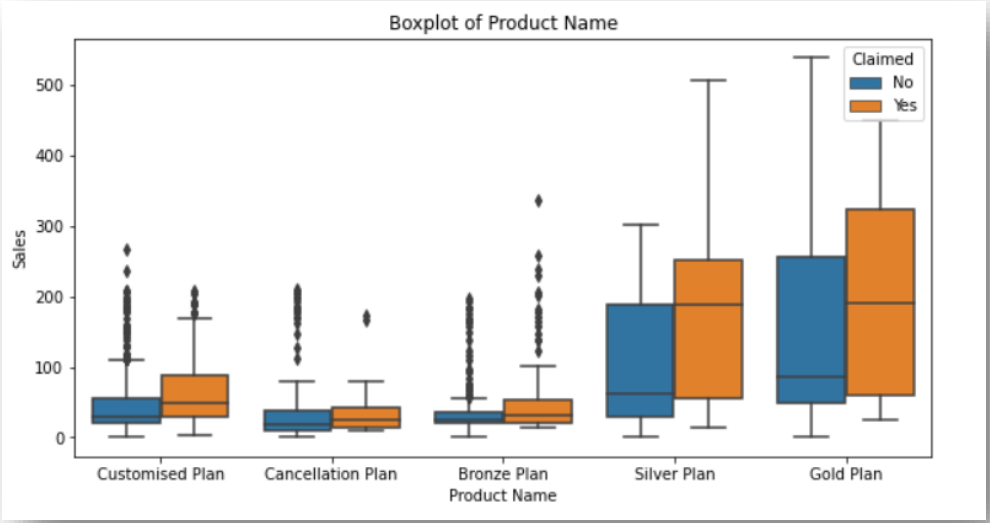

Figure 30: Boxplot of 'Product Name'

From Figure 30, we can see that the Box plot of 'Product Name' variable has outliers. Within this variable, however, 'Silver Plan' and 'Gold Plan' do not have any outliers. The distribution of the data is moderately right skewed or positively skewed which is also seen in Figure 30. However the distribution of the data is left skewed for claim status 'Yes' in the 'Silver Plan' and normal distribution for claim status 'Yes' in the 'Gold Plan'.

6. **Destination:**

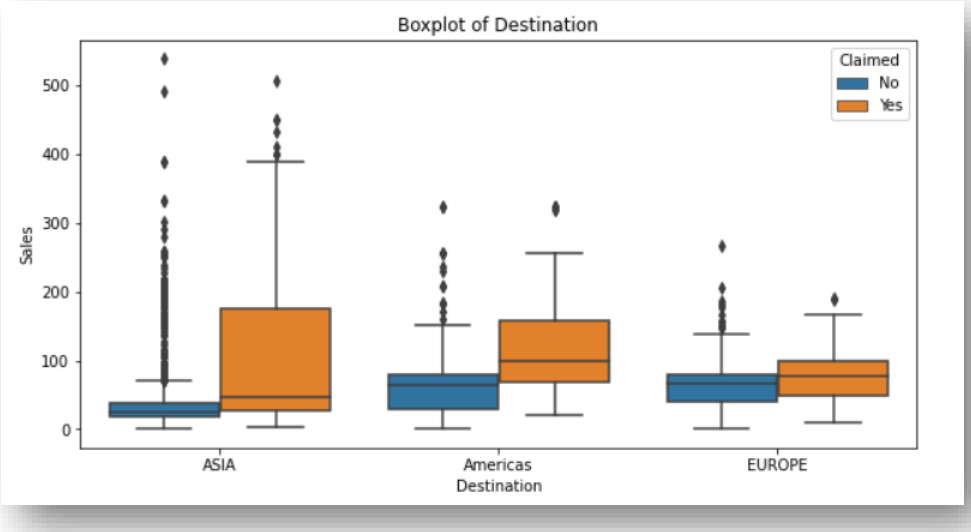

Figure 31: Boxplot of 'Destination'

From Figure 31, we can see that the Box plot of 'Destination' variable has many outliers. The distribution of the data is moderately right skewed or positively skewed which is also seen in Figure 31.

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

```
X = df2.drop("Claimed", axis=1)
y = df2.pop("Claimed")
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, train_labels, test_labels = train_test_split(X, y, test_size=.30, random_state=1)
```

Figure 32: Splitting the data into train and test

The data is first split into train and test data as shown in Figure 32 before building the decision tree, random forest and the artificial neural network model. There is no fixed rule for separation training and testing data sets. Most of the researchers use **70:30 ratio** for separation data sets. The same ratio was used in this dataset to split the data into train and test. The random state was set to be 1.

Scaling is a way of representing a dataset. Scaling needs to be done as a dataset has features or variables with different 'weights' for each feature. In such cases, it is suggested to transform the features so that all the features are in the same 'scale'. This is called scaling. Scaling also makes it easier to compare the features now that the weightage on each feature is the same.

Scaling can be done by Z-score method and Min-max method. Z-score method is used when you want to centralize the data (weight-based). Example: principal component analysis (PCA), neural network etc., Min-max method is used when the data is distance based. Example: Clustering, KNN etc., Models have to be implemented only after scaling is done.

The data is scaled using Z-score method before building the different models. The scaled head of the dataset is shown in Table 12.

Table 12: Head of the scaled dataset by Z-score method

	Age	Agency_Code	Type	Commision	Channel	Duration	Sales	Product Name	Destination
0	0.947162	-1.314358	-1.256796	-0.542807	0.124788	-0.470051	-0.816433	0.268835	-0.434646
1	-0.199870	0.697928	0.795674	-0.570282	0.124788	-0.268605	-0.569127	0.268835	-0.434646
2	0.086888	-0.308215	0.795674	-0.337133	0.124788	-0.499894	-0.711940	0.268835	1.303937
3	-0.199870	0.697928	0.795674	-0.570282	0.124788	-0.492433	-0.484288	-0.525751	-0.434646
4	-0.486629	1.704071	-1.256796	-0.323003	0.124788	-0.126846	-0.597407	-1.320338	-0.434646

Decision Tree Classifier

```
{'criterion': 'gini', 'max_depth': 7, 'min_samples_leaf': 19, 'min_samples_split': 139}

DecisionTreeClassifier(max_depth=7, min_samples_leaf=19, min_samples_split=139,
                       random_state=1)
```

Figure 33: Best parameters for Decision Tree Classifier Model

Decision Tree model is supervised learning algorithm which can be used for classification and regression type of problems. Many combinations of parameters were tried using Grid Search Cross Validation or the Grid Search CV function. Multiple values were passed for the different parameters. Figure 33 shows the best parameters for decision tree classifier model.

- The **'criterion'** used for the Decision Tree Classifier is **'gini'**.
- The **'max_depth'** parameter is used to prune the trees. It is the number of decision trees or the level of the trees. The optimum 'max_depth' parameter is found to be **7** meaning the depth level of the decision tree is 7.
- The **'min_samples_leaf'** is a parameter that determines how many observations should be in each leaf node or terminal. In this data, the 'min_samples_leaf' was determined to be **19**.
- The parameter **'min_samples_split'** determines how many observations a node should have for it to be split into left and right child nodes. **139** was found to be the optimum 'min_samples_split' value.

Table 13: Feature Importances of Decision Tree Classifier

	Imp
Agency_Code	0.561444
Sales	0.247335
Product Name	0.071714
Age	0.046122
Duration	0.045311
Commision	0.021128
Type	0.006946
Channel	0.000000
Destination	0.000000

Feature importance is a measure of the features used in the split and the contribution of the different features in the split. A value of 0 means that that particular independent variable was never used in splitting the nodes. These values are a relative measure of feature importance and not absolute. From Table 13, we can see that 'Agency_Code' was the most important feature in splitting the nodes whereas 'Channel' and 'Destination' was never used in split.

Random Forest Classifier

```
{'max_depth': 10, 'max_features': 3, 'min_samples_leaf': 3, 'min_samples_split': 25, 'n_estimators': 300}

RandomForestClassifier(max_depth=10, max_features=3, min_samples_leaf=3,
                       min_samples_split=25, n_estimators=300, random_state=1)
```

Figure 34: Best parameters for Random Forest Classifier Model

Random Forest technique is an ensemble technique wherein we construct multiple models and take the average output of all the models to take a final decision or make a prediction. Many combinations of parameters were tried using Grid Search Cross Validation or the Grid Search CV function. Multiple values were passed for the different parameters. Figure 34 shows the best parameters for random forest classifier model.

- The **'max_depth'** parameter is used to prune the trees. It is the number of decision trees or the level of the trees. The optimum 'max_depth' parameter is found to be 10 meaning the depth level of the decision tree is **10**.
- The **'max_features'** parameter determines how many number of the independent variables or features a random forest classifier uses for evaluating and splitting the decision nodes. The 'max_features' was found to be **3** in this dataset.
- The **'min_samples_leaf'** is a parameter that determines how many observations need to be present in each of the terminal nodes or the leaf nodes in all the decision trees in the random forest. In this data, the 'min_samples_leaf' was determined to be **3**.
- The parameter **'min_samples_split'** determines how many observations a node should have for it to be split into left and right child nodes. **25** was found to be the optimum 'min_samples_split' value.
- The **'n_estimators'** is the number of trees you want to build in a random forest and the optimum value was found to be **300**.

Table 14: Feature Importances of Random Forest Classifier

	Imp
Agency_Code	0.214320
Sales	0.182379
Product Name	0.177333
Commision	0.142495
Duration	0.121317
Age	0.085214
Type	0.056292
Destination	0.015263
Channel	0.005386

Feature importance is a measure of the features used in the split and the contribution of the different features in the split. A value of 0 means that that particular independent variable was never used in splitting the nodes. These values are a relative measure of feature importance and not

absolute. From Table 14, we can see that 'Agency_Code' was the most important feature in splitting the nodes whereas 'Destination' and 'Channel' were the least important feature used in split.

Artificial Neural Network

```
{'hidden_layer_sizes': 100, 'max_iter': 40, 'solver': 'adam', 'tol': 0.01}

MLPClassifier(hidden_layer_sizes=100, max_iter=40, random_state=1, tol=0.01)
```

Figure 35: Best parameters for Artificial Neural Network Model

Artificial Neural Network is a machine learning algorithm that is modelled around what is currently known about how the human brain functions. Similar to a biological neural network, an artificial neural network has the ability to learn, generalize and adapt. It is made of 3 layers – Input, Hidden and Output layer. Many combinations of parameters were tried using Grid Search Cross Validation or the Grid Search CV function. Multiple values were passed for the different parameters. Figure 35 shows the best parameters for artificial neural network model.

- The **'hidden_layer_sizes'** was found to be **100** meaning there is one layer with 100 neurons.
- The **'max_iter'** parameter is the maximum number of iterations to update the synaptic weights of neurons. The optimum 'max_iter' was found to be **40**.
- The **'solver'** is the algorithm used in the process of backpropagation to calculate the weights of the neural network. The 'solver' used is **'adam'** solver.
- The **'tol'** parameter is the threshold level. The lower the threshold, higher the accuracy and lesser the number of times the model will execute and vice versa. The optimum threshold value was found to be **'0.01'**.

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.

Model Performance Metrics:

Confusion matrix:

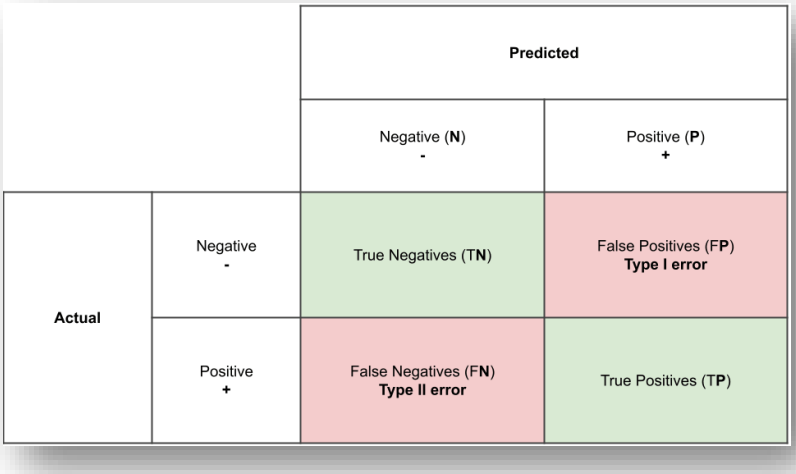

Figure 36: Confusion Matrix

A confusion matrix is a 2x2 tabular structure reflecting the performance of the model in four blocks. Figure 36 shows how a confusion matrix will look like. True Positive (TP) and True Negative (TN) are correct predictions. False Positive (FP) and False Negative (FN) are incorrect predictions.

Table 15: Confusion Matrix formulas

Metric Name	Formula from Confusion Matrix
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\frac{TP}{TP+FP}$
Recall, Sensitivity, TPR	$\frac{TP}{TP+FN}$
Specificity, 1-FPR	$\frac{TN}{TN+FP}$
F1	$\frac{2 * precision * recall}{precision + recall}$

- **Accuracy** – it is a measure of how accurately or cleanly the model classifies the data points. Lesser the false predictions, more the accuracy. In a classification problem, the best score is **100%** accuracy.

- **Precision** – it is a measure of how many among the points identified as positive by the model, are really positive. A precision score of 1.0 means that all the points are identified as positive by the model. **1.0** is a perfect precision score. However it does not indicate about the number of observations that were not labelled correctly.
- **Recall or sensitivity** – is the ratio of correctly predicted positive observations to the all observations in actual class. A perfect recall score is **1.0** but a recall score above 0.5 is considered as a good recall score. However the recall score does not indicate about how many observations are incorrectly predicted.
- **Specificity** – is a measure of how many of the actual negative data points are identified as negative by the model.
- **F1** – it is the measure of the model's accuracy on a dataset. The F-score is a way of combining the precision and recall of the model and it is defined as the harmonic mean of the model's precision and recall. An F1 score is considered perfect when it's **1**, while the model is a total failure when the score is 0.

All these can be calculated from the confusion matrix by using the formulas given in Table 15.

Classification report:

A Classification report is used to measure the quality of predictions from a classification algorithm. The classification report shows the main classification metrics and their scores. The metrics are precision, recall, f1-score and accuracy for the actual and predicted data. The metrics are calculated by using true and false positives, true and false negatives from the confusion matrix.

ROC Curve:

Receiver Operating Characteristics (ROC) Curve is a technique for visualizing classifier performance. It is a graph between true positive (TP) rate and false positive (FP) rate.

$$\text{TP rate} = \frac{TP}{total\ positive}$$

$$\text{FP rate} = \frac{FP}{total\ negative}$$

ROC graph is a trade-off between benefits (TP) and costs (FP). The steeper the ROC Curve, the stronger the model will be and vice versa.

ROC_AUC score:

Area under the ROC Curve (AUC) is the measure of the area under the ROC Curve. The ROC_AUC score gives us the value of the area under the ROC Curve. The larger the area under the curve, the better the model.

Decision Tree Classifier

Training Data

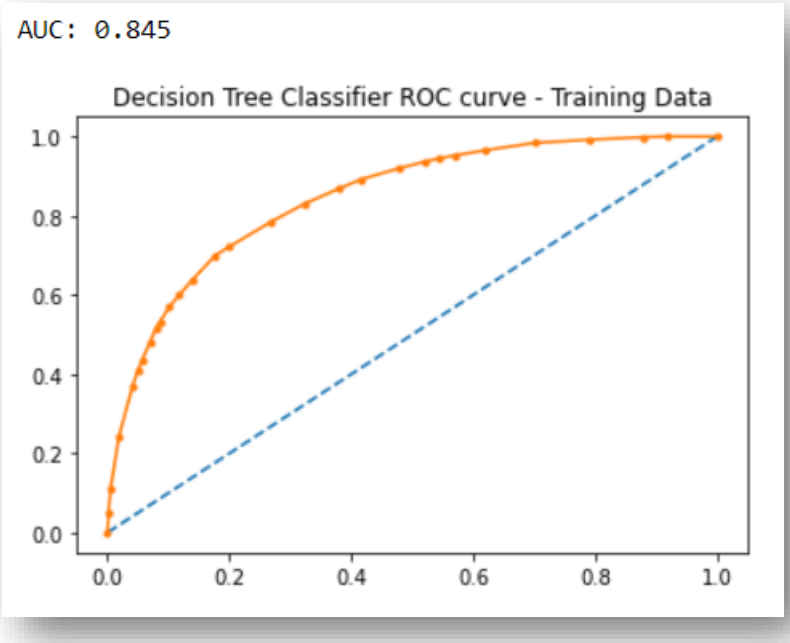

Figure 37: Decision Tree Classifier ROC Curve for Training Data

The ROC_AUC score for the training data was calculated to be 0.845. The ROC curve was plotted and is shown in Figure 37.

```
array([[1322,  149],
       [ 271,  358]], dtype=int64)
```

Figure 38: Decision Tree Classifier Confusion Matrix for Training Data

```
dtcl_train_acc=best_grid_dtcl.score(X_train,train_labels)
dtcl_train_acc
```

```
0.8
```

Figure 39: Decision Tree Classifier Accuracy score for Training Data

The confusion matrix was calculated and shown in Figure 38. The accuracy score for the training data was found to be 0.8 which is shown in Figure 39.

Table 16: Decision Tree Classifier Classification Report for Training Data

	precision	recall	f1-score	support
0	0.83	0.90	0.86	1471
1	0.71	0.57	0.63	629
accuracy			0.80	2100
macro avg	0.77	0.73	0.75	2100
weighted avg	0.79	0.80	0.79	2100

The classification report for the decision tree classifier model with the scores for the different model performance measures is calculated and shown in Table 16. From this table we can see that the precision of the model is 0.71 means 71% of the data points identified as positive by the model, are really positive .The f1-score is 0.63 means the model is 63% accurate on this data set. Both these scores are low. However the model has 80% accuracy. The recall score is 0.57 which means 57% of the positive observations are correctly predicted.

Test Data

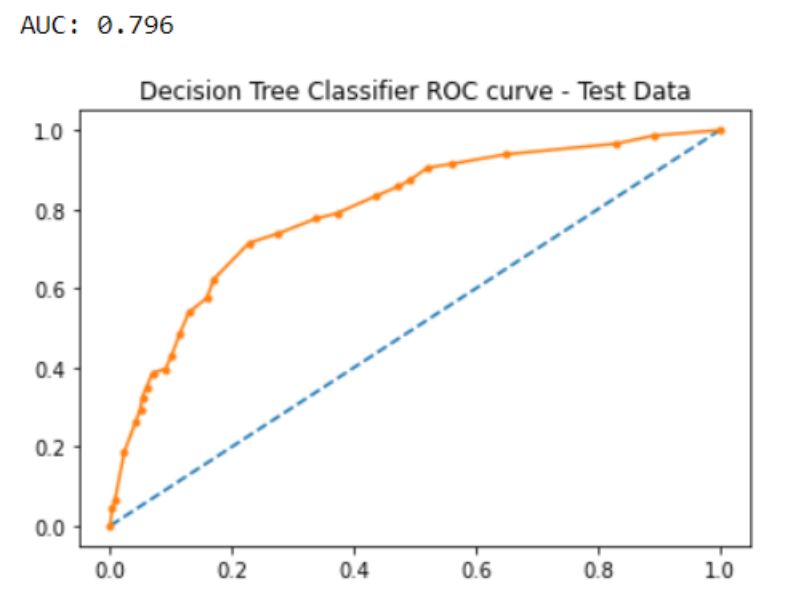

Figure 40: Decision Tree Classifier ROC Curve for Test Data

The ROC_AUC score for the test data was calculated to be 0.796. The ROC curve was plotted and is shown in Figure 40.

```
array([[544,  61],
       [168, 127]], dtype=int64)
```

Figure 41: Decision Tree Classifier Confusion Matrix for Test Data

```
dtcl_test_acc=best_grid_dtcl.score(X_test,test_labels)
dtcl_test_acc
```

```
0.7455555555555555
```

Figure 42: Decision Tree Classifier Accuracy score for Test Data

The confusion matrix was calculated and shown in Figure 41. The accuracy score for the test data was found to be 0.746 which is shown in Figure 42.

Table 17: Decision Tree Classifier Classification Report for Test Data

	precision	recall	f1-score	support
0	0.76	0.90	0.83	605
1	0.68	0.43	0.53	295
accuracy			0.75	900
macro avg	0.72	0.66	0.68	900
weighted avg	0.74	0.75	0.73	900

The classification report for the decision tree classifier model with the scores for the different model performance measures is calculated and shown in Table 17. From this table we can see that the precision of the model is 0.68 means 68% of the data points identified as positive by the model, are really positive .The f1-score is 0.53 means the model is 53% accurate on this data set. Both these

scores are low. However the model has 75% accuracy. The recall score is 0.43 which means 43% of the positive observations are correctly predicted.

Random Forest Classifier

Training Data

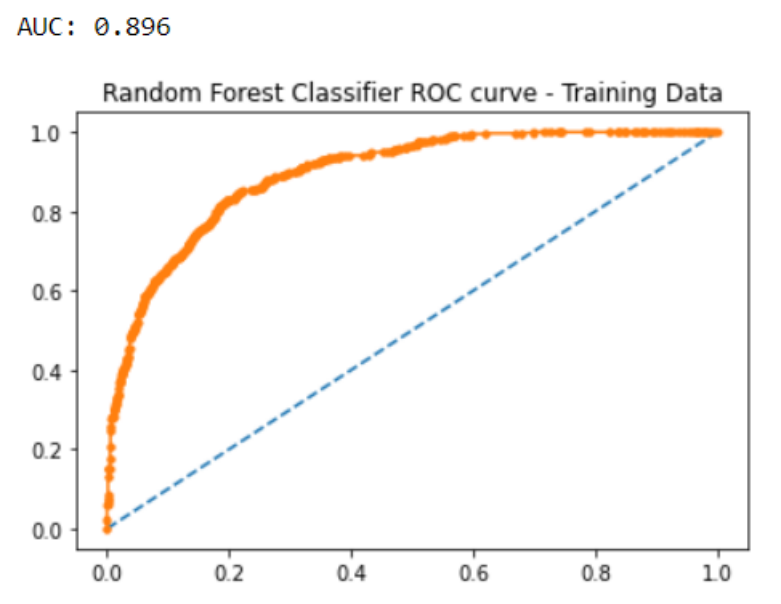

Figure 43: Random Forest Classifier ROC Curve for Training Data

The ROC_AUC score for the training data was calculated to be 0.896. The ROC curve was plotted and is shown in Figure 43.

```
array([[1347,  124],
       [ 231,  398]], dtype=int64)
```

Figure 44: Random Forest Classifier Confusion Matrix for Training Data

```
rfcl_train_acc=best_grid_rfcl.score(X_train,train_labels)
rfcl_train_acc
```

```
0.830952380952381
```

Figure 45: Random Forest Classifier Accuracy score for Training Data

The confusion matrix was calculated and shown in Figure 44. The accuracy score for the training data was found to be 0.831 which is shown in Figure 45.

Table 18: Random Forest Classifier Classification Report for Training Data

	precision	recall	f1-score	support
0	0.85	0.92	0.88	1471
1	0.76	0.63	0.69	629
accuracy			0.83	2100
macro avg	0.81	0.77	0.79	2100
weighted avg	0.83	0.83	0.83	2100

The classification report for the random forest classifier model with the scores for the different model performance measures is calculated and shown in Table 18. From this table we can see that the precision of the model is 0.76 means 76% of the data points identified as positive by the model, are really positive .The f1-score is 0.69 means the model is 69% accurate on this data set. Both these scores are higher than the decision tree model which is good comparatively. The model has 83% accuracy which is also better than the decision tree model. The recall score is 0.63 which means 63% of the positive observations are correctly predicted.

Test Data

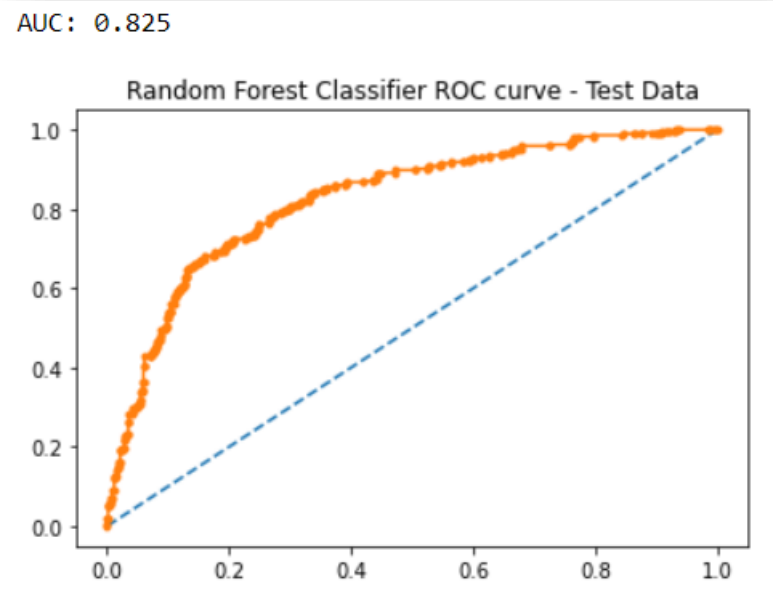

Figure 46: Random Forest Classifier ROC Curve for Test Data

The ROC_AUC score for the test data was calculated to be 0.825. The ROC curve was plotted and is shown in Figure 46.

```
array([[552,  53],
       [149, 146]], dtype=int64)
```

Figure 47: Random Forest Classifier Confusion Matrix for Test Data

```
rfcl_test_acc=best_grid_rfcl.score(X_test,test_labels)
rfcl_test_acc
```

```
0.7755555555555556
```

Figure 48: Random Forest Classifier Accuracy score for Test Data

The confusion matrix was calculated and shown in Figure 47. The accuracy score for the test data was found to be 0.775 which is shown in Figure 48.

Table 19: Random Forest Classifier Classification Report for Test Data

	precision	recall	f1-score	support
0	0.79	0.91	0.85	605
1	0.73	0.49	0.59	295
accuracy			0.78	900
macro avg	0.76	0.70	0.72	900
weighted avg	0.77	0.78	0.76	900

The classification report for the random forest classifier model with the scores for the different model performance measures is calculated and shown in Table 19. From this table we can see that the precision of the model is 0.73 means 73% of the data points identified as positive by the model, are really positive .The f1-score is 0.59 means the model is 59% accurate on this data set. Both these scores are higher than the decision tree model which is good comparatively. The model has 78% accuracy which is also better than the decision tree model. The recall score is 0.49 which means 49%

of the positive observations are correctly predicted. The recall score is low but the other measures have good scores which makes it a better model.

Artificial Neural Network

Training Data

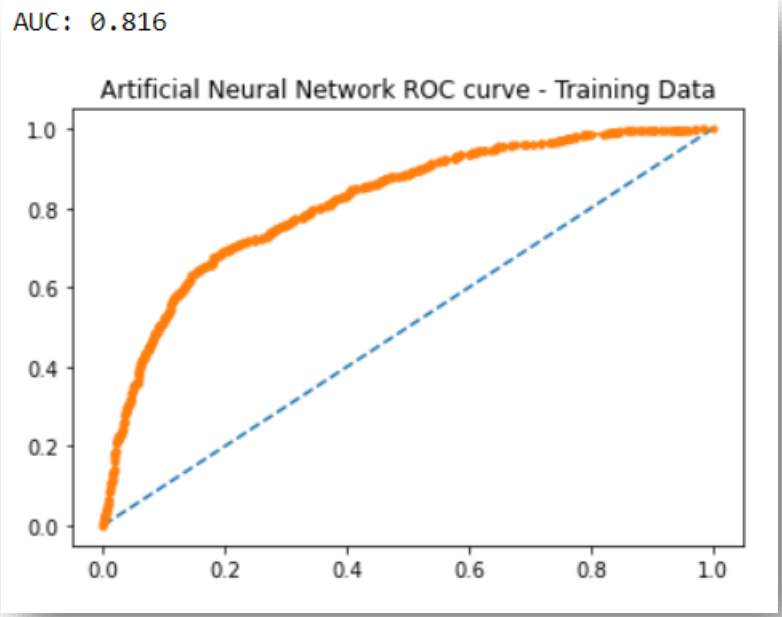

Figure 49: Artificial Neural Network ROC Curve for Training Data

The ROC_AUC score for the training data was calculated to be 0.816. The ROC curve was plotted and is shown in Figure 49.

```
array([[1289,  182],
       [ 262,  367]], dtype=int64)
```

Figure 50: Artificial Neural Network Confusion Matrix for Training Data

```
nncl_train_acc=best_grid_nncl.score(X_train,train_labels)
nncl_train_acc
```

```
0.7885714285714286
```

Figure 51: Artificial Neural Network Accuracy score for Training Data

The confusion matrix was calculated and shown in Figure 50. The accuracy score for the training data was found to be 0.788 which is shown in Figure 51.

Table 20: Artificial Neural Network Classification Report for Training Data

	precision	recall	f1-score	support
0	0.83	0.88	0.85	1471
1	0.67	0.58	0.62	629
accuracy			0.79	2100
macro avg	0.75	0.73	0.74	2100
weighted avg	0.78	0.79	0.78	2100

The classification report for the artificial neural network model with the scores for the different model performance measures is calculated and shown in Table 20. From this table we can see that the precision of the model is 0.67 means 67% of the data points identified as positive by the model, are really positive .The f1-score is 0.62 means the model is 62% accurate on this data set. Both these scores are higher than the decision tree model but lower than the random forest model. The model has 79% accuracy. The recall score is 0.58 which means 58% of the positive observations are correctly predicted. The recall score is low.

Test Data

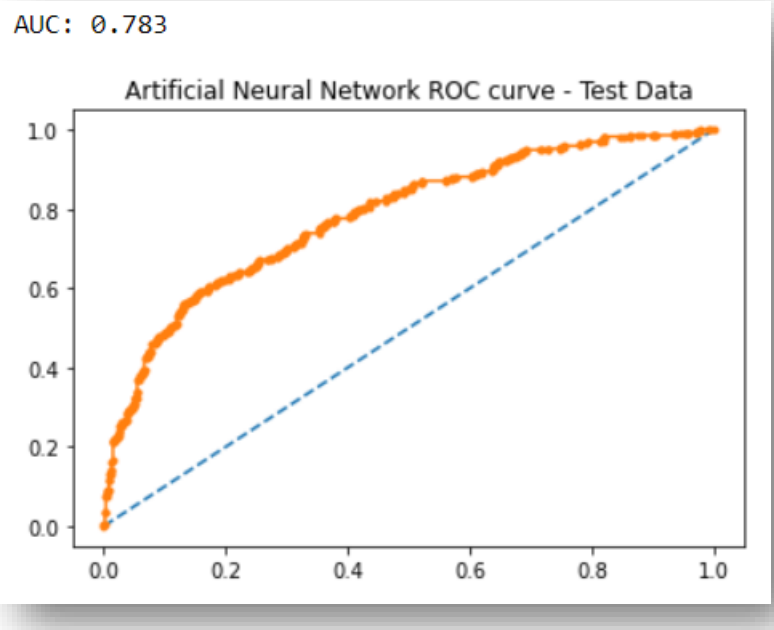

Figure 52: Artificial Neural Network ROC Curve for Test Data

The ROC_AUC score for the test data was calculated to be 0.783. The ROC curve was plotted and is shown in Figure 52.

```
array([[547,  58],
       [154, 141]], dtype=int64)
```

Figure 53: Artificial Neural Network Confusion Matrix for Test Data

```
nncl_test_acc=best_grid_nncl.score(X_test,test_labels)
nncl_test_acc
```

```
0.7644444444444445
```

Figure 54: Artificial Neural Network Accuracy score for Test Data

The confusion matrix was calculated and shown in Figure 53. The accuracy score for the test data was found to be 0.764 which is shown in Figure 54.

Table 21: Artificial Neural Network Classification Report for Test Data

	precision	recall	f1-score	support
0	0.78	0.90	0.84	605
1	0.71	0.48	0.57	295
accuracy			0.76	900
macro avg	0.74	0.69	0.70	900
weighted avg	0.76	0.76	0.75	900

The classification report for the artificial neural network model with the scores for the different model performance measures is calculated and shown in Table 21. From this table we can see that the precision of the model is 0.71 means 71% of the data points identified as positive by the model, are really positive .The f1-score is 0.57 means the model is 57% accurate on this data set. The model has 76% accuracy. The recall score is 0.48 which means 48% of the positive observations are correctly predicted. The recall score is low.

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

A comparison of all the three models was done to understand which model is the best suited for our case study. The model performance measures of all the three models were tabulated and it is shown in Table 22.

Table 22: Comparison of all the three models

	Decision Tree Train	Decision Tree Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.800	0.746	0.831	0.776	0.789	0.764
AUC	0.845	0.796	0.896	0.825	0.816	0.783
Recall	0.570	0.570	0.630	0.490	0.580	0.480
Precision	0.710	0.710	0.760	0.730	0.670	0.710
F1 Score	0.630	0.630	0.690	0.590	0.620	0.570

From Table 22, we are able to understand that the ROC_AUC score for the random forest model is the highest compared to the other two models. The larger the area under the curve, the better the model. This is also true in case of accuracy where the score is highest in the random forest model compared to the other two models. The precision and f1 score of the random forest train data is the highest. The recall score for the random forest train data is the highest but low for the test data. However since all the other model performance measures are good for the random forest model, it is chosen as the optimized model for our problem.

This is further analysed with the help of the ROC Curve.

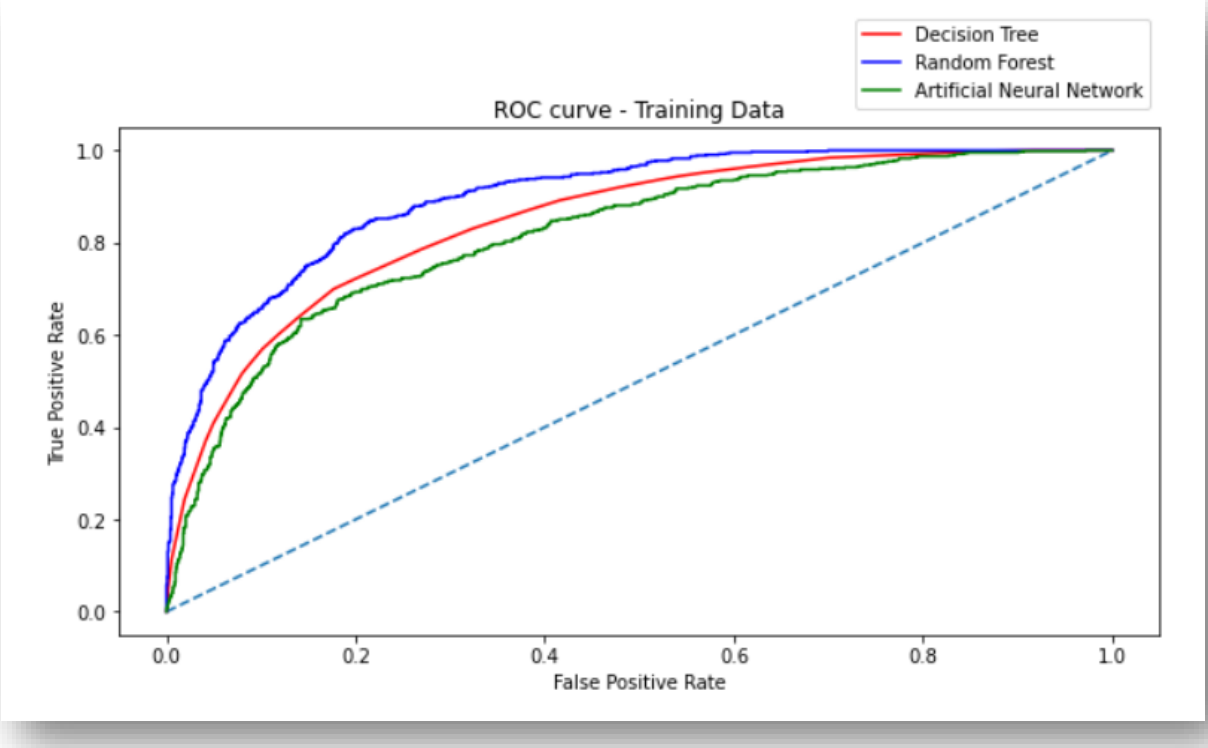

Figure 55: ROC Curve for all 3 models - Training Data

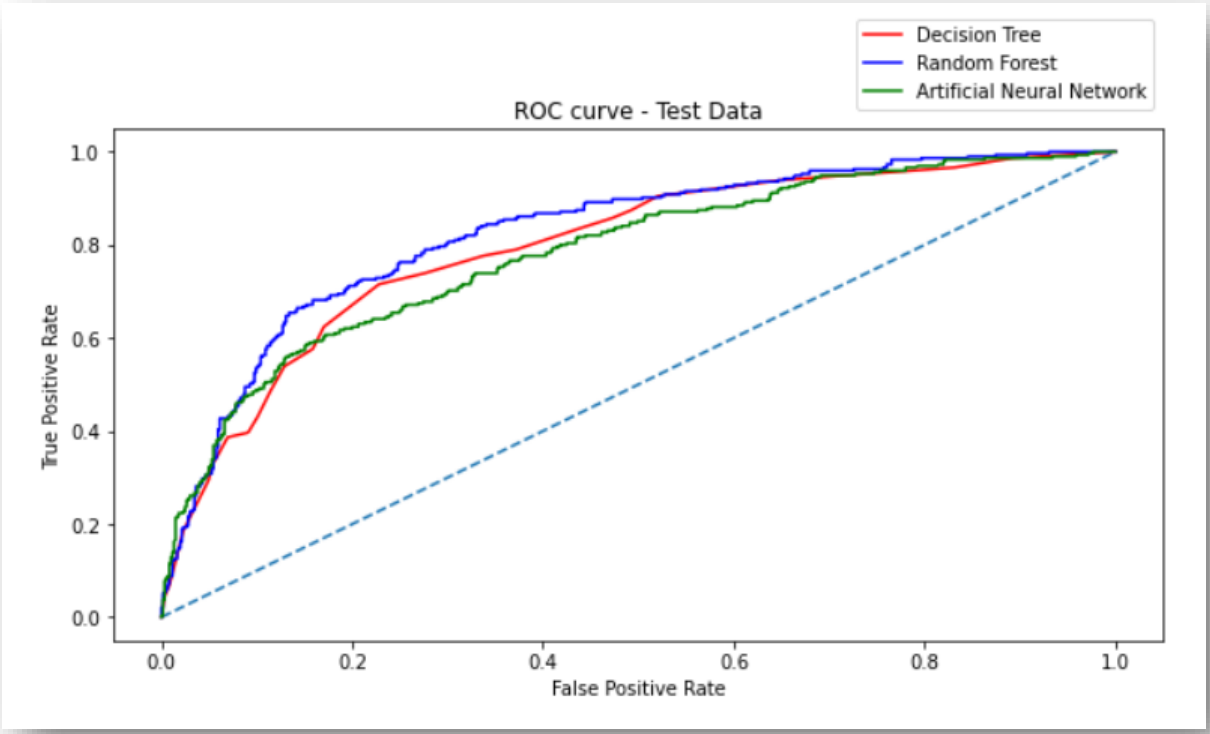

Figure 56: ROC Curve for all 3 models - Test Data

From Figure 55 and Figure 56, we are able to see that the Random Forest Classifier model is the most optimum for our case study as the curve for the Random Forest Classifier model is the steepest. The steeper the ROC Curve, the stronger the model.

Therefore **Random Forest Classifier** is selected as the best model for our problem as it has better accuracy, precision, recall and f1 score compared to decision tree and artificial neural network model.

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

The recommendations for this problem is that more real time unstructured data and past data should be collected. Relations between different variables such as day of the incident, time, and age group should be associated with other external information such as location, behaviour patterns, weather information, airline/vehicle types, etc., can be done to figure out the relations between the variables.

The JZI agency resources have to improve their sales. They can improve the sales by advertising, conducting promotional marketing campaigns etc. They can also improve their sales by having a tie up with other agencies.

According to the observations that were made in the data, 90% of insurance is done by online channel. Therefore this can be concentrated more. More options to digitalize the process can be done. Making the process online has benefitted the customers as well as the business. This has

profited the business as the cost on paper and other expenses is saved. Almost all the offline business have a claim associated with it. A study can be performed to understand why this is the scenario for offline businesses.

Asia is the most common destination of the tour and has a claim associated with it. Other destinations can also be encouraged by advertising the tourist attractions and offers to those destinations.

Sales should be improved. This can be done by offering the customers a lesser claim cycle time, increasing customer satisfaction, optimizing the claim recovery, combating fraud, etc., Tour insurance ads can be made to pop up when a customer is looking for planning a tour or booking tickets.

The majority of the people who took the insurance were of the age 38. Young people travel a lot. People are most likely to explore the world when they are young. They can be encouraged to take tour insurance. Targeting young group can improve the sales of the business.

From the data, we can see that more sales happen via Agency than Airlines and the trend shows the claims are processed more at Airlines. So the insurance company can conduct researches as to why the sales happen more via agency and understand the process better.

Insights gained from this model can be used to improve the business. Young people can be targeted, improvements can be made in the existing insurance policies, new policies can be designed focusing on damage claim etc.