PROJECT – DATA MINING

ADITHYA MANIVANNAN

PGP – DSBA

Contents

PROBLEM 1: CLUSTERING

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.

Introduction

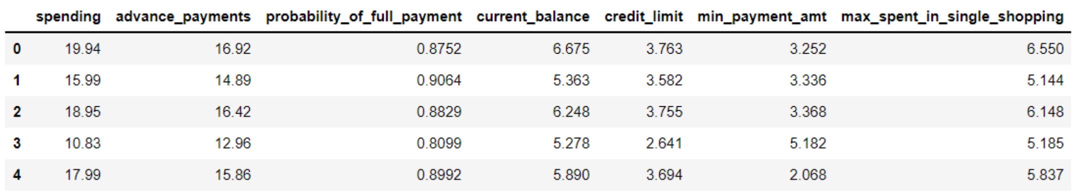
The purpose of this whole exercise is to explore the dataset. Do the exploratory data analysis. Explore the dataset using central tendency and other parameters. The data consists of activities of different customers based on their credit card usage. We will perform exploratory data analysis to understand what the given data has to say and then use clustering techniques to develop a customer segmentation so that the bank can give promotional offers to its customers based on the clusters we have identified.

Data Dictionary for Market Segmentation:

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 (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)

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

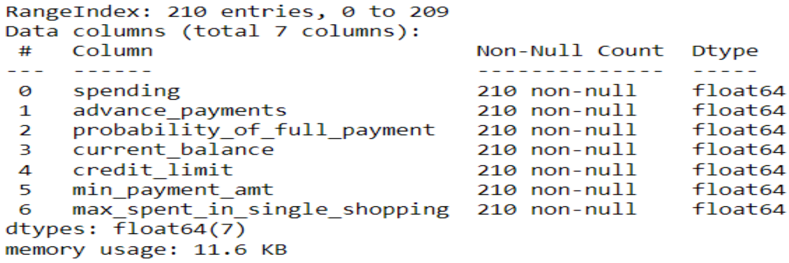
Sample of the dataset:



	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

The dataset has 7 different variables that define the spending of the credit card users.

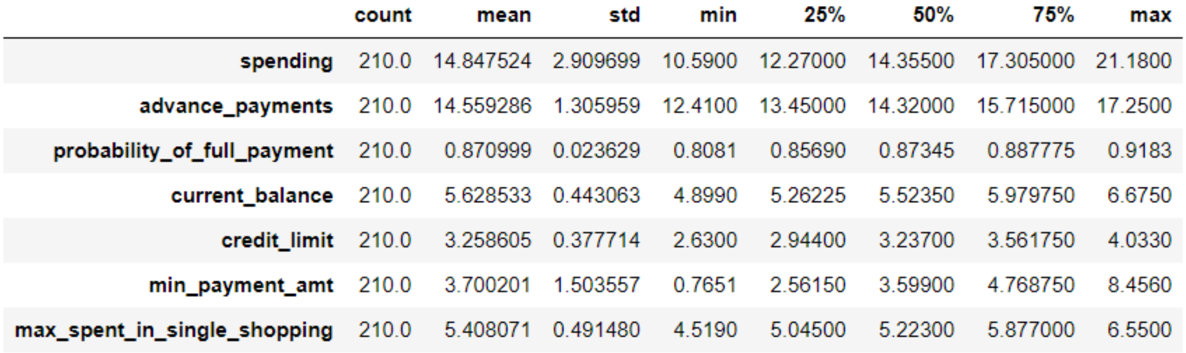
Exploratory Data Analysis:



```
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 diagram, we can understand that there are 7 columns with 210 rows each. Each variable is of the Float data type and there are no null values in any of the columns.

5 Point Summary:



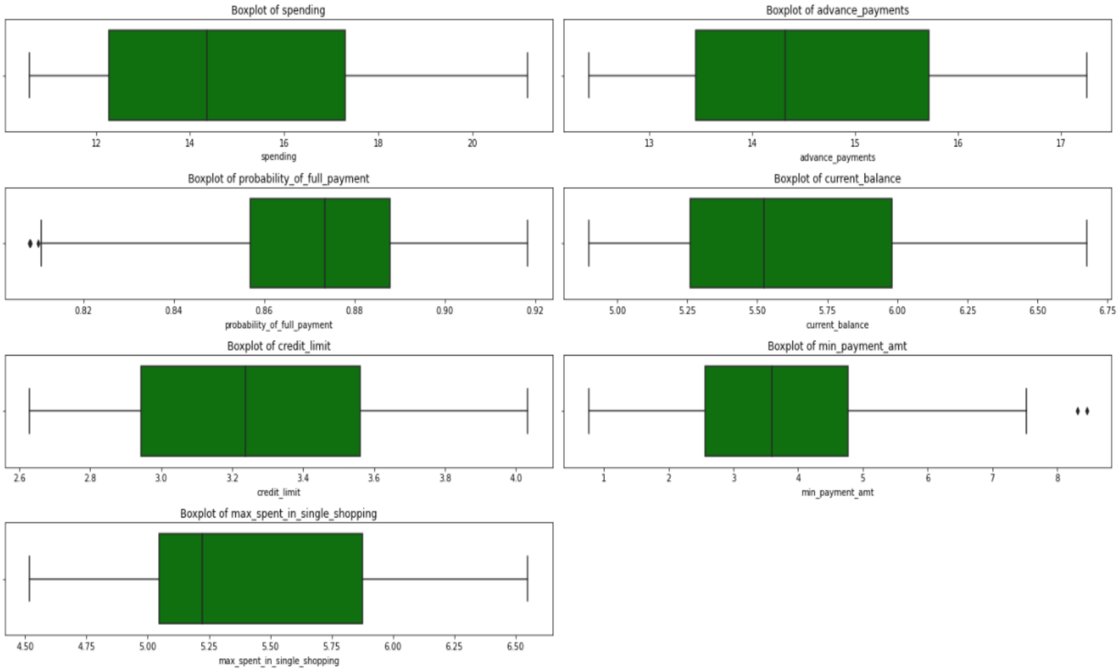
	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.847524	2.909699	10.5900	12.27000	14.35500	17.305000	21.1800
advance_payments	210.0	14.559286	1.305959	12.4100	13.45000	14.32000	15.715000	17.2500
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.85690	0.87345	0.887775	0.9183
current_balance	210.0	5.628533	0.443063	4.8990	5.26225	5.52350	5.979750	6.6750
credit_limit	210.0	3.258605	0.377714	2.6300	2.94400	3.23700	3.561750	4.0330
min_payment_amt	210.0	3.700201	1.503557	0.7651	2.56150	3.59900	4.768750	8.4560
max_spent_in_single_shopping	210.0	5.408071	0.491480	4.5190	5.04500	5.22300	5.877000	6.5500

From the 5-point summary, we can understand the following:

- It looks like all the variables are normally distributed since the mean and median are almost the same.
- The average advance payments is roughly 10% of the average spending.
- The average probability of full payment is 87.09%
- The minimum of min_payment_amt paid is 76.51. The maximum of min_payment_amt paid is 845.60. This suggests the data is widely spread for this variable and might have outliers

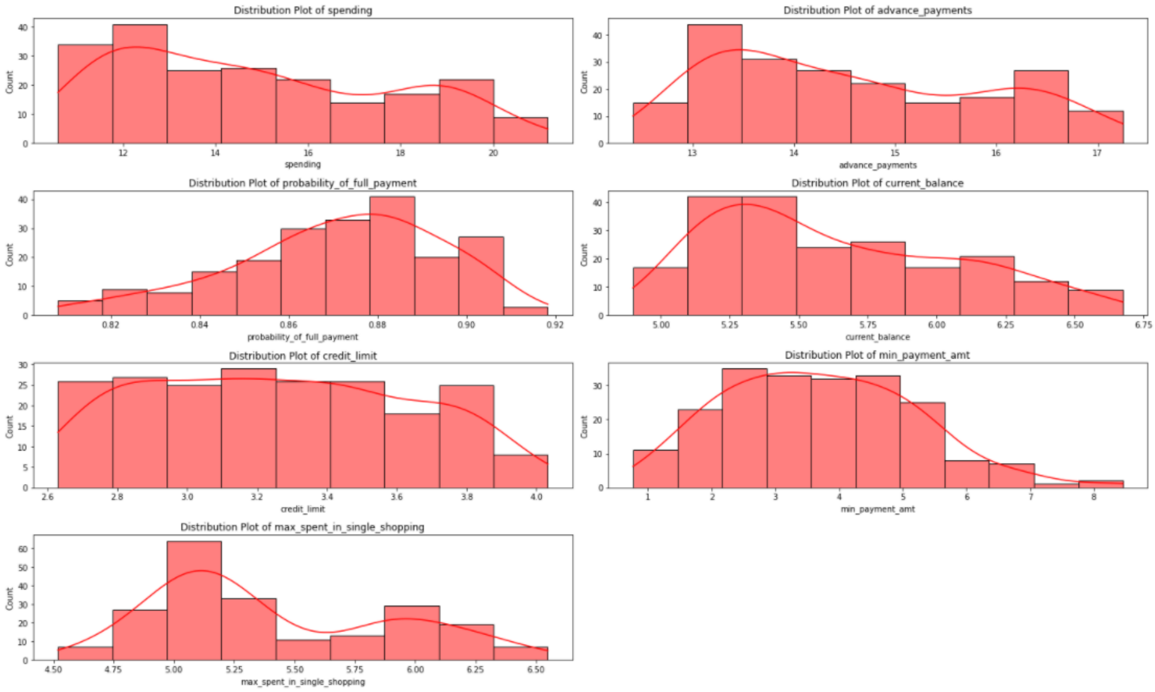
The average of max_spent_in_single_shopping is 5408.07. The maximum of max_spent_in_single_shopping is 6550.00.

Univariate Analysis:

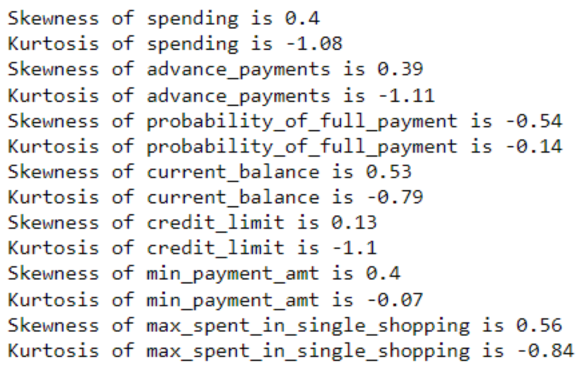


From the boxplot, we can understand that the variables "probability_of_full_payment" and "min_payment_amt" have outliers while none of the other variables have their outliers.

Distribution Plot



Skewness and Kurtosis



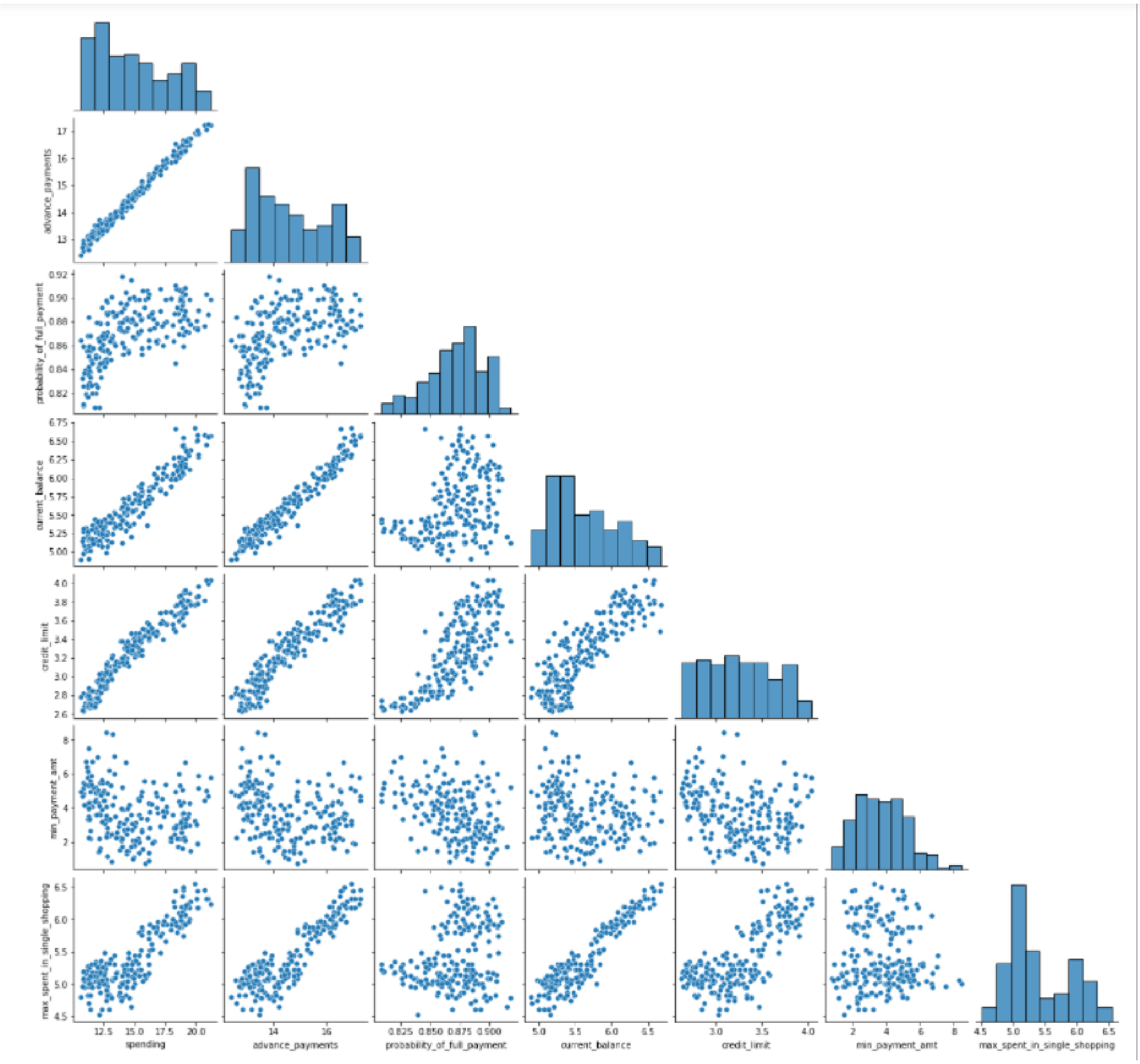
```
Skewness of spending is 0.4
Kurtosis of spending is -1.08
Skewness of advance_payments is 0.39
Kurtosis of advance_payments is -1.11
Skewness of probability_of_full_payment is -0.54
Kurtosis of probability_of_full_payment is -0.14
Skewness of current_balance is 0.53
Kurtosis of current_balance is -0.79
Skewness of credit_limit is 0.13
Kurtosis of credit_limit is -1.1
Skewness of min_payment_amt is 0.4
Kurtosis of min_payment_amt is -0.07
Skewness of max_spent_in_single_shopping is 0.56
Kurtosis of max_spent_in_single_shopping is -0.84
```

The skewness is a measure of symmetry or asymmetry of data distribution, and kurtosis measures whether data is heavy-tailed or light-tailed in a normal distribution. Data can be positive-skewed (data-pushed towards the right side) or negative-skewed (data-pushed towards the left side).

We can see that the skewness is between 0 and 1 for all variables which shows that its is slightly skewed.

Bivariate Analysis

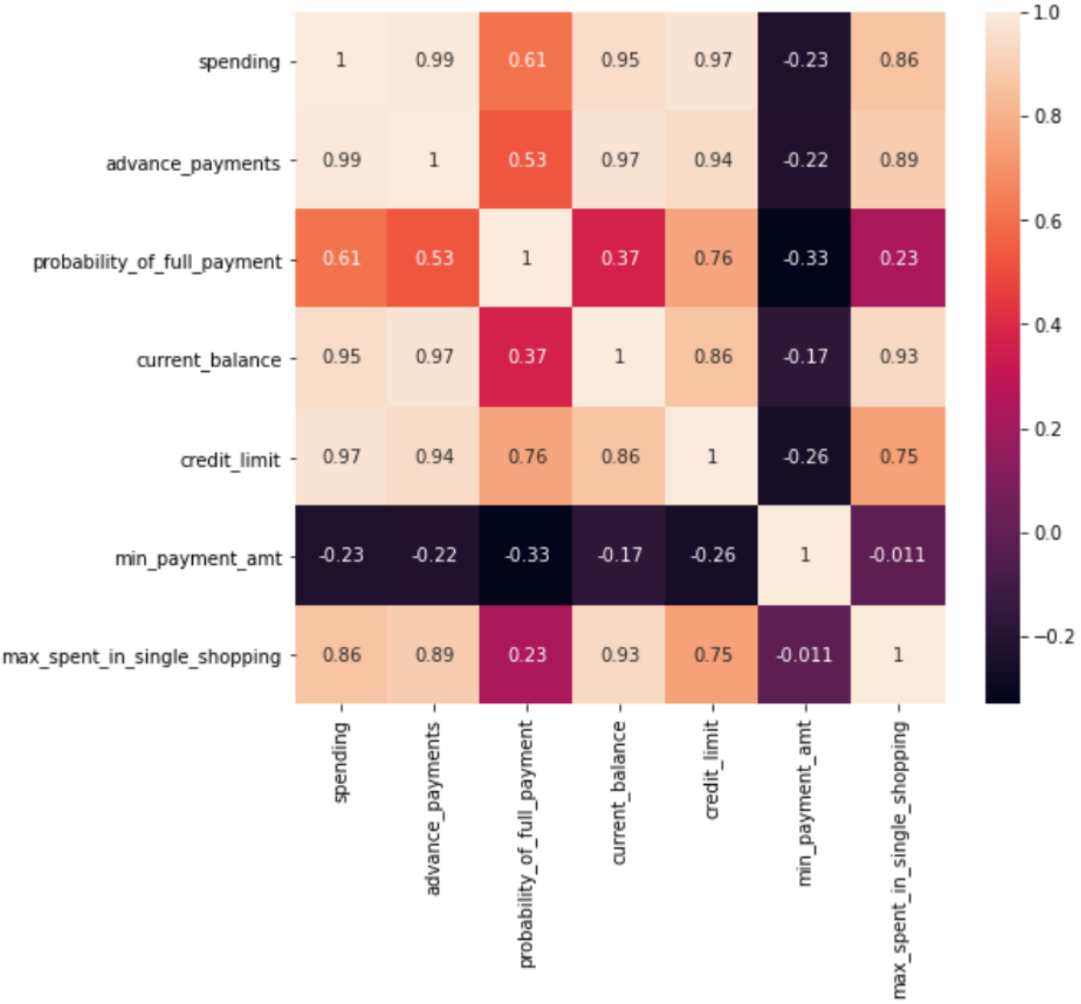
Pairplot



The pairplot shows that there is a linear relationship between a lot of the variables. We can see a positive linearity between advance_payments and spending, balance, credit_limit. Similarly, even between credit_limit and spending.

Let us know find the correlation using a correlation heatmap.

Correlation heatmap:



From the above correlation plot we can see those various aspects of credit card usage have high positive correlation with each other.

Correlation values are always between 1 and -1. Those which are closer to 1 are positively correlated and those which near -1 are negatively correlated. Values near to 0 have no correlation. These correlation support our assumptions made from the pairplot.

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

Clustering is essentially "grouping close things together and distant things separate". If we don't normalize the features, we will end up giving more weight to some features than others leading to incorrect clustering.

Let us see the variances between variables in the provided dataset.

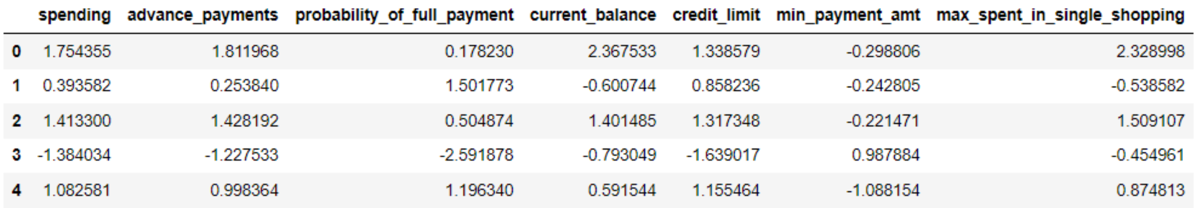
```
spending                        8.466351
advance_payments                1.705528
probability_of_full_payment     0.000558
current_balance                 0.196305
```

```
credit_limit                      0.142668
min_payment_amt                   2.260684
max_spent_in_single_shopping      0.241553
```

From the above table, though there is not much variance between most of the variables, our target variable spending has a variance of 8.46 whereas other variables variance lies between 0 and 3. Hence scaling is necessary.

We will be using the Standard Scaler method for scaling our data. This method will calculate the z-score for each data point and then scale the data such that mean = 0 and variance/standard deviation = 1.

After scaling the data, below is the head of the data.



	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

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

Hierarchical Clustering

Hierarchical clustering is a technique of cluster analysis which performs hierarchy of clusters. There are mainly two types of hierarchical clustering:

1. Divisive: This is a top-to-down approach. It starts with all of the observations in the same cluster and then splits into smaller clusters.

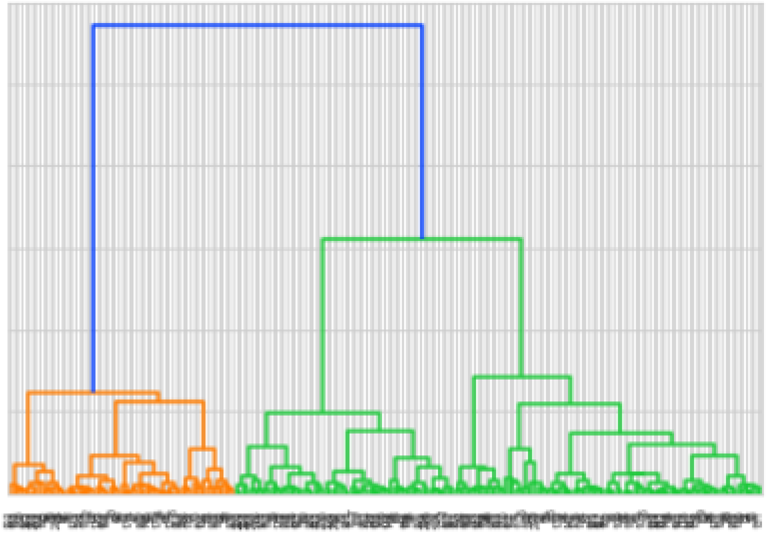
2. Agglomerative: This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

Concept of Linkage

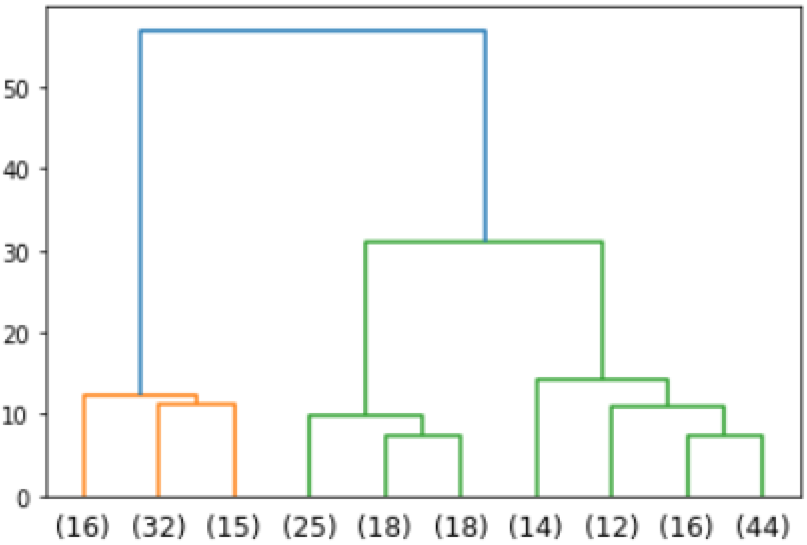
Once the clusters are formed, the distance between clusters are calculated using different linkage methods. The different linkage types are: 1. Single linkage 2. Complete linkage 3. Average linkage 4. Centroid linkage 5. Ward’s method.

For our dataset we will only look at Ward’s method linkages

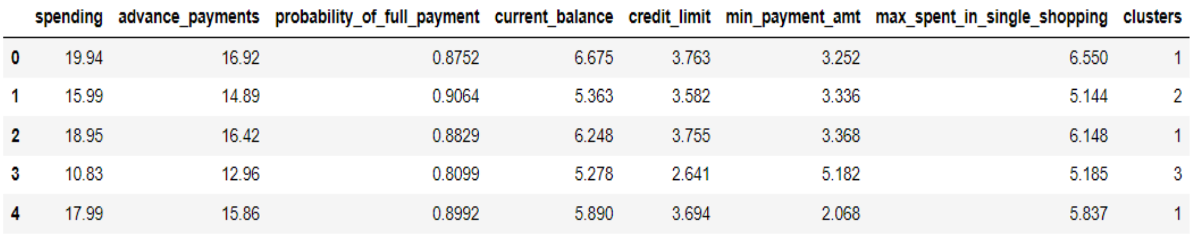
Ward’s method: joins records and clusters together progressively to produce larger and larger clusters. We will use the scipy package for dendrogram and linkage.



From the above diagrams we can see that the data has been segregated into three clusters by color (blue, orange and green) using Ward's method. However, since we are not able to see the data points, we have truncated the dendrogram as below.

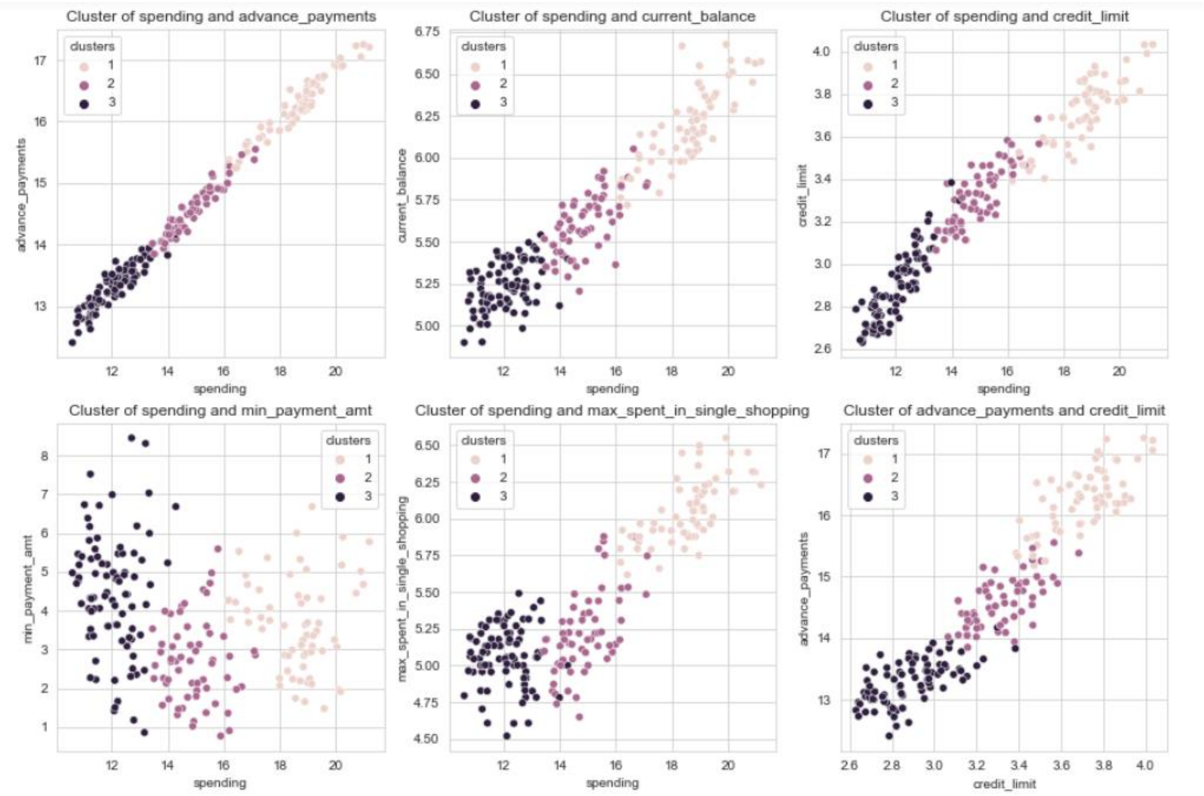


Head of our dataset after merging the clusters:



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

Let us now see the relationship between clusters and each variable using scatterplots.



We can see 3 different clusters that are formed and the relationship between these and the variables

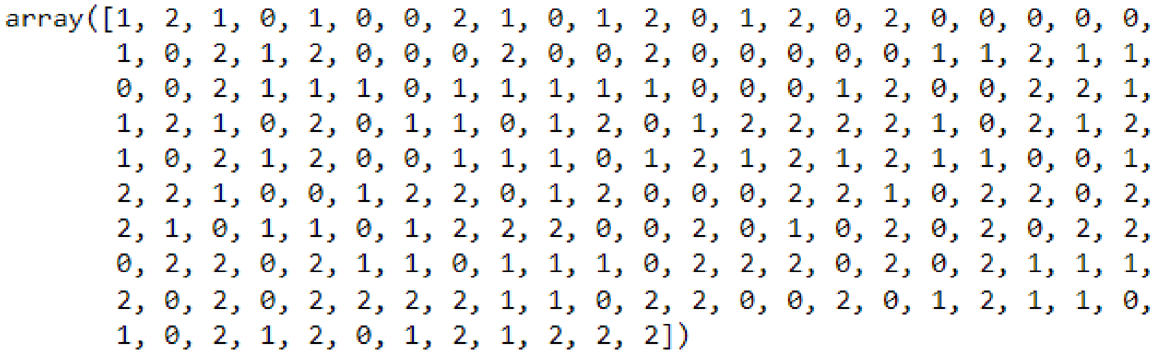
1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Explain the results properly. Interpret and write inferences on the finalized clusters.

K-Means clustering is a unsupervised learning algorithm which tries to find groups or form clusters on the basis of their similarity.

Parameter K: k is the target variable which refers to the number of centroids in the given dataset.
Means: in K-Means clustering, ‘Means’ refers to the averaging of data to find the centroid in a cluster.
There are techniques to find the optimal number of k values as below:
1. The Elbow method
2. The Silhouette method

Let us now apply K-means clustering on the scaled data followed by calculating WSS scores and plotting them to check the optimal k value using Elbow method. We will be using sklearn.cluster package to use Kmeans.

After we scaled the data, we will first try applying k-means clustering with number of clusters as 3. Below are the labels after applying k-means clustering.



```
array([1, 2, 1, 0, 1, 0, 0, 2, 1, 0, 1, 2, 0, 1, 2, 0, 2, 0, 0, 0, 0, 0,
       1, 0, 2, 1, 2, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 1, 1, 2, 1, 1,
       0, 0, 2, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 2, 0, 0, 2, 2, 1,
       1, 2, 1, 0, 2, 0, 1, 1, 0, 1, 2, 0, 1, 2, 2, 2, 2, 1, 0, 2, 1, 2,
       1, 0, 2, 1, 2, 0, 0, 1, 1, 1, 0, 1, 2, 1, 2, 1, 2, 1, 1, 0, 0, 1,
       2, 2, 1, 0, 0, 1, 2, 2, 0, 1, 2, 0, 0, 0, 2, 2, 1, 0, 2, 2, 0, 2,
       2, 1, 0, 1, 1, 0, 1, 2, 2, 2, 0, 0, 2, 0, 1, 0, 2, 0, 2, 0, 2, 2,
       0, 2, 2, 0, 2, 1, 1, 0, 1, 1, 1, 0, 2, 2, 2, 0, 2, 0, 2, 1, 1, 1,
       2, 0, 2, 0, 2, 2, 2, 2, 1, 1, 0, 2, 2, 0, 0, 2, 0, 1, 2, 1, 1, 0,
       1, 0, 2, 1, 2, 0, 1, 2, 1, 2, 2, 2])
```

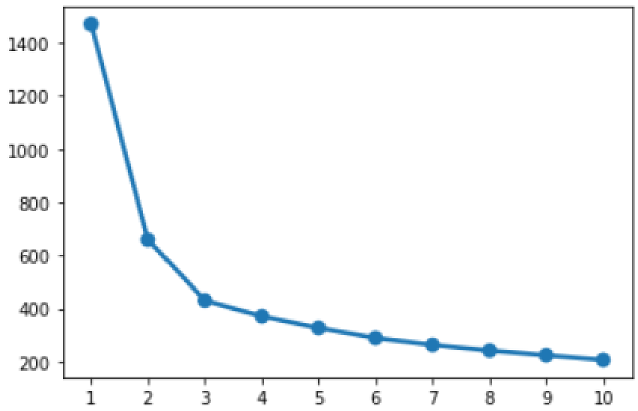
However, when we try applying k-means clustering with k value starting from 1 to 10, we have the below inertia/WSS

1469.9999999999995,
659.1717544870411,
430.65897315130064,
371.38509060801107,
327.2127816566134,
289.315995389595,
262.98186570162267,
241.8189465608603,
223.91254221002728,
206.3961218478669

From the above WSS scores, we can see that for cluster 1 the score is 1469.99 and the score for 2 clusters dropped to 659.17 which is a significant difference in the scores. For cluster 3, the score is 430.65 whereas from cluster 3 to cluster 10 we see that there is no significant decrease in the scores. Hence, we arrive at the optimum no of clusters as 3. Let us now see the Elbow method visually to understand the scores better.

Elbow Method

For a given number of clusters, the total within cluster-sum of squares (WSS) is computed. That value of k is chosen to be optimum, where addition of one more cluster does not lower the value of total WSS appreciably

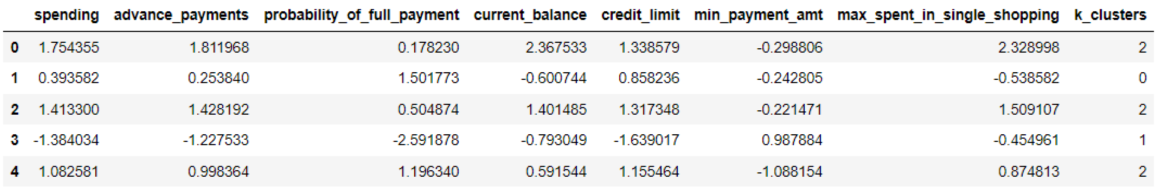


We then calculate the silhouette scores for 3 and 4 clusters to see the optimal K.

When the clusters were 3, we obtained a value of 0.40072705527512986 and when the clusters was changed to 4, we obtained a silhouette score of 0.3276547677266192.

Hence, since the silhouette score of 3 clusters was better, it was chosen as the optimal k.

Let us now append the original dataframe with the k_means clusters.

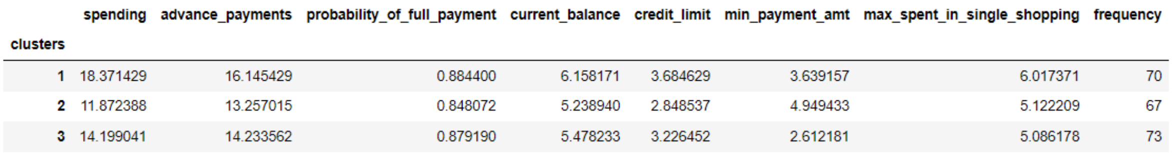


	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	k_clusters
0	1.754355	1.811968	0.178230	2.367533	1.338579	-0.298806	2.328998	2
1	0.393582	0.253840	1.501773	-0.600744	0.858236	-0.242805	-0.538582	0
2	1.413300	1.428192	0.504874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

This is the head of the data with the clusters mentioned.

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

Recommendations for Hierarchical clustering



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	frequency
clusters								
1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371	70
2	11.872388	13.257015	0.848072	5.238940	2.848537	4.949433	5.122209	67
3	14.199041	14.233562	0.879190	5.478233	3.226452	2.612181	5.086178	73

When we look at the final clusters merged with original dataset and take the average values for the variables, below are the recommendations for each cluster profile.

Cluster 1: Platinum customers

Cluster 3: Gold customers

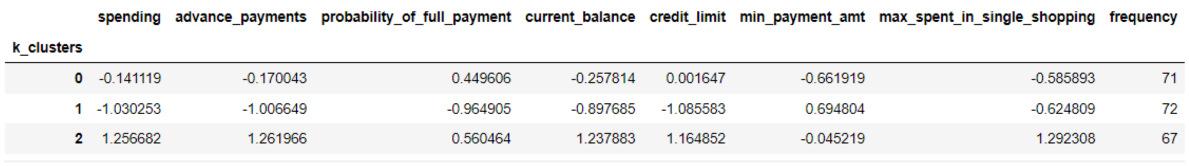
Cluster 2: Silver customers

Customers under cluster 1 have a high spending, current balance, credit_limit and max_spent_in_single_shopping which clearly shows that they are premium high-net worth customers who make expensive purchases on their credit cards. Provide offers on high expense purchases such as luxury products, flight tickets etc. Increase rewards for loyalty programs.

Customers under cluster 3 have a relatively lesser spending, current balance, credit_limit and max_spent_in_single_shopping which indicate that they are upper middle-class customers. The bank can provide promotional offers to this segment such that they increase their spending and are potential customers who can move into premium segments. Offers can be provided on general groceries and things that lower and middle class families will purchase frequently in a month.

Customers under cluster 2 have the least spending and credit_limits compared to other clusters. This signifies that they are customers who have recently bought credit cards or youths who have started working recently. Bank can provide customized offers to this segment to promote more spending on credit cards. They can have a check on the credit_limit but also provide lower interests to encourage more usage.

Recommendations for K-Means clustering



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	frequency
k_clusters								
0	-0.141119	-0.170043	0.449606	-0.257814	0.001647	-0.661919	-0.585893	71
1	-1.030253	-1.006649	-0.964905	-0.897685	-1.085583	0.694804	-0.624809	72
2	1.256682	1.261966	0.560464	1.237883	1.164852	-0.045219	1.292308	67

K-Means Clusters When we look at the final clusters merged with original dataset and take the average values for the variables, below are the recommendations for each cluster profile.

Cluster 0: Platinum customers

Cluster 2: Gold Customers

Cluster 1: Silver Customers

Customers under cluster 0 have a high spending, current balance, credit_limit and max_spent_in_single_shopping which clearly shows that they are premium high-net worth customers who make expensive purchases on their credit cards. Provide offers on high expense purchases such as luxury products, flight tickets etc. Increase rewards for loyalty programs.

Customers under cluster 2 have a relatively lesser spending, current balance, credit_limit and max_spent_in_single_shopping which indicate that they are upper middle-class customers. The bank can provide promotional offers to this segment such that they increase their spending and are potential customers who can move into premium segments. Offers can be provided on general groceries and things that lower and middle class families will purchase frequently in a month.

Customers under cluster 1 have the least spending and credit_limits compared to other clusters. This signifies that they are customers who have recently bought credit cards or youths who have started working recently. Bank can provide customized offers to this segment to promote more spending on credit cards. They can have a check on the credit_limit but also provide lower interests to encourage more usage.

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.

Introduction

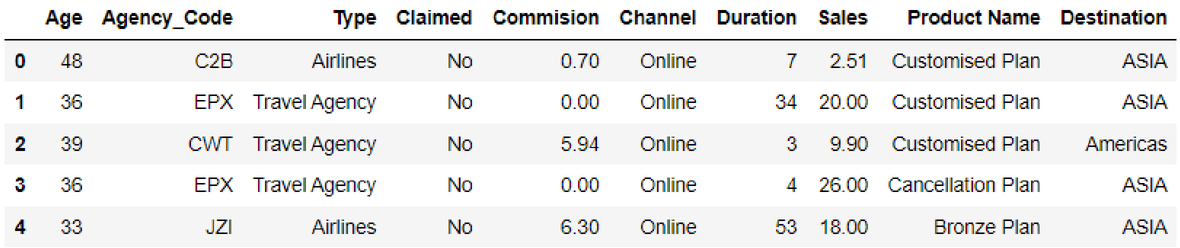
The purpose of this whole exercise is to explore the dataset. Do the exploratory data analysis. Explore the dataset using central tendency and other parameters. The data consists of customer details who have availed tour insurance. We will perform exploratory data analysis to understand what the given data has to say and then use Decision Trees, Random Forest and Artificial Neural Network to build a model which predicts the claim status and provide recommendations to management with the business insights gained.

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).

Sample Dataset:

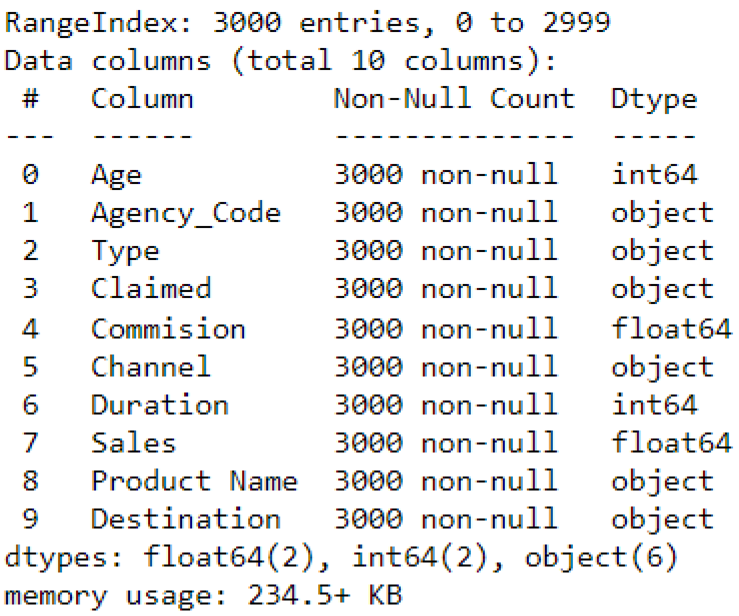


	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	C2B	Airlines	No	0.70	Online	7	2.51	Customised Plan	ASIA
1	36	EPX	Travel Agency	No	0.00	Online	34	20.00	Customised Plan	ASIA
2	39	CWT	Travel Agency	No	5.94	Online	3	9.90	Customised Plan	Americas
3	36	EPX	Travel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

We can see that there are 10 variables and from the data dictionary, we understand that "Claimed" is the target variable. Hence, this will be the dependant variable while building the model.

Exploratory Data Analysis:

Information of dataset:

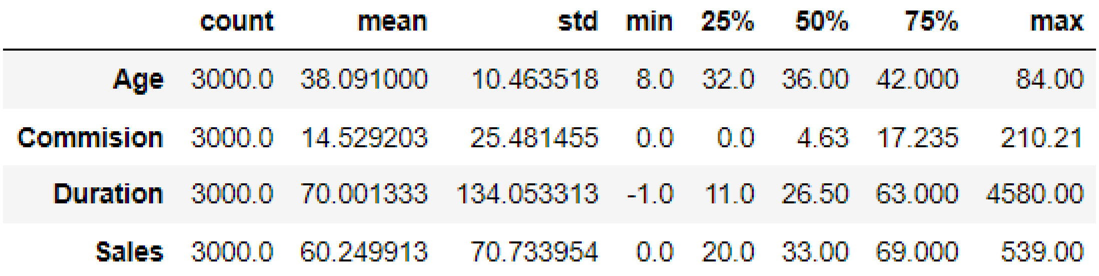


```
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Age           3000 non-null   int64  
 1   Agency_Code   3000 non-null   object 
 2   Type          3000 non-null   object 
 3   Claimed       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
```

We can see that there are 3000 rows and 10 columns. Of the 10 columns, 6 are of object data type while 2 are of float and the other 2 are int. There are no null values.

However, there are 139 rows of duplicates in the dataset. Hence, we will be removing the duplicates from the dataset.

Descriptive Statistics

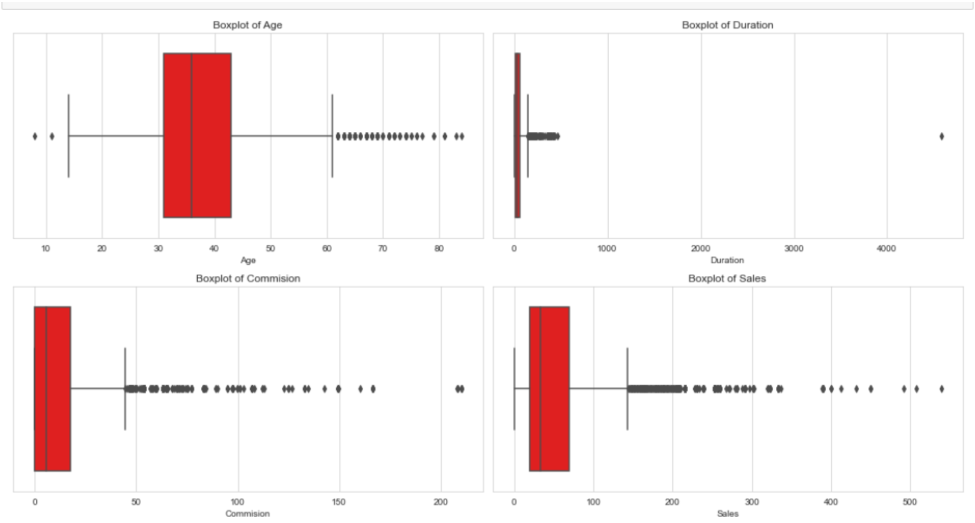


	count	mean	std	min	25%	50%	75%	max
Age	3000.0	38.091000	10.463518	8.0	32.0	36.00	42.000	84.00
Commision	3000.0	14.529203	25.481455	0.0	0.0	4.63	17.235	210.21
Duration	3000.0	70.001333	134.053313	-1.0	11.0	26.50	63.000	4580.00
Sales	3000.0	60.249913	70.733954	0.0	20.0	33.00	69.000	539.00

From the above table below are the observations:

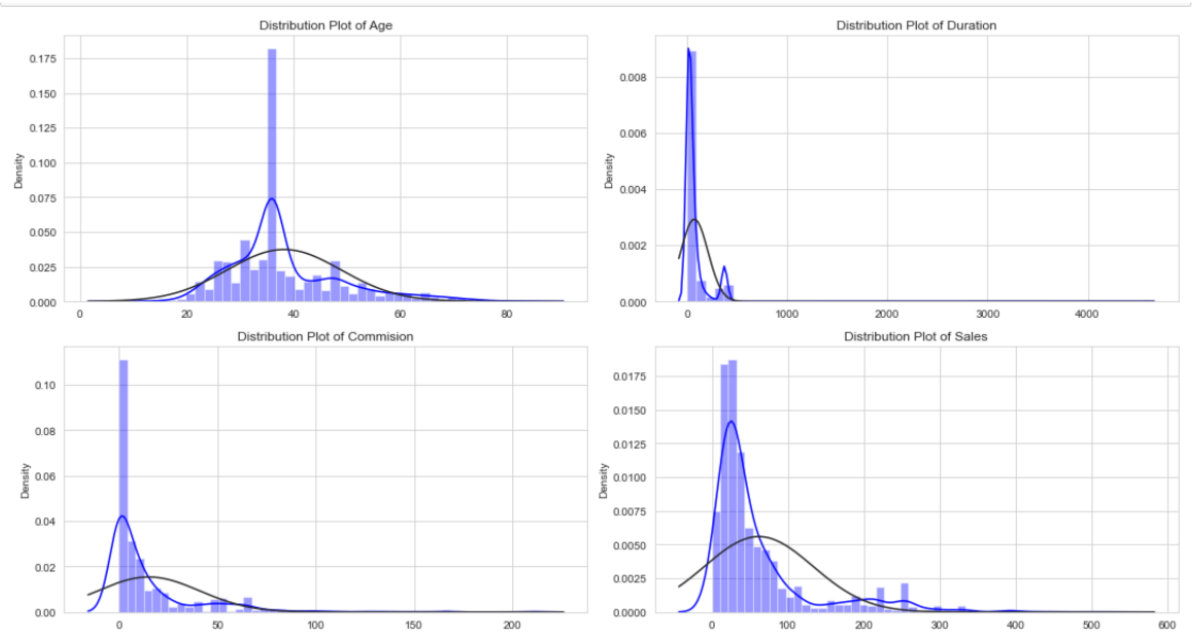
1. Age seems to be normally distributed. The minimum age is 8 yrs and max age is 84 yrs which shows that we might have the correct data on Age. Average age of customers in the dataset is 38 yrs.
2. 2. Commission received for tour insurance is widely spread. The average commission is 14.52 however the minimum commission received is 0 and maximum commission received is 210.21 and 95% of data lies within 63.21 which indicates that there might be outliers.
3. Duration of the tour ranges from 134 days to 4580 days which means there are outliers. 95% of data lies within 367 days whereas maximum days shows 4580 days which is close to 12 years. It is surprising to see that someone would go for a tour for 12 years. Need to consult the business to validate the data. For now we will go with further analysis as is. The minimum duration shows -1 which cannot be the case in reality. Hence we might drop this row or impute it mean value to treat bad data.
4. Sales figures ranges from 0 to 539 (in 100's). The average sales is amounted to 60.24. There might be outliers as well.
5. Except for Age the other variables seem to be skewed with outliers present.

Univariate Analysis

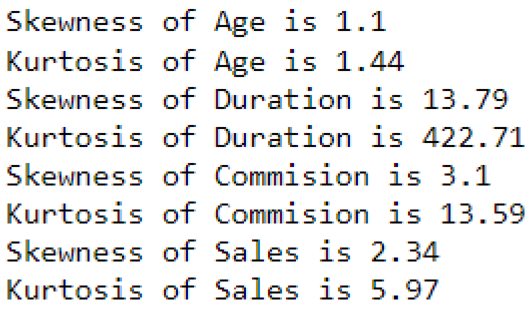


From the above boxplots, we can see that there are outliers present in the dataset for each numerical variable.

Distribution Plots



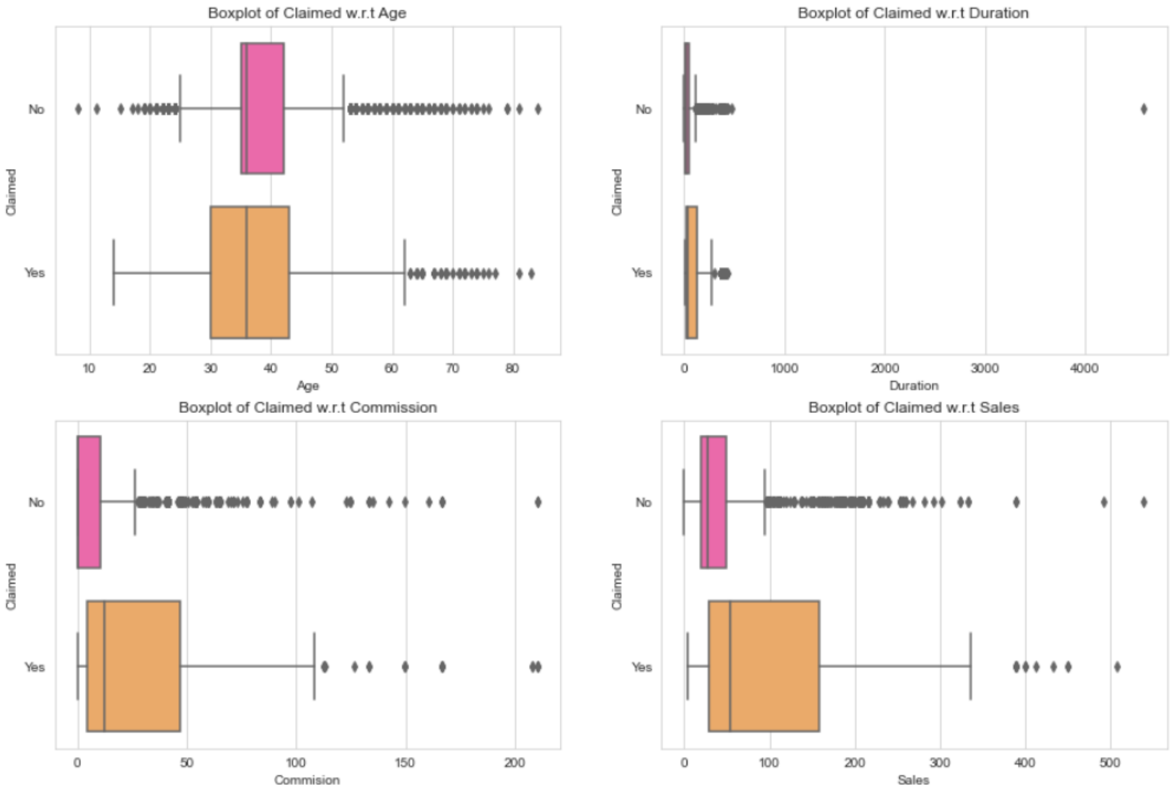
From the above distribution plots we can see that the variables are highly skewed except for Age. Let's validate the same with Skewness and Kurtosis scores.



```
Skewness of Age is 1.1
Kurtosis of Age is 1.44
Skewness of Duration is 13.79
Kurtosis of Duration is 422.71
Skewness of Commision is 3.1
Kurtosis of Commision is 13.59
Skewness of Sales is 2.34
Kurtosis of Sales is 5.97
```

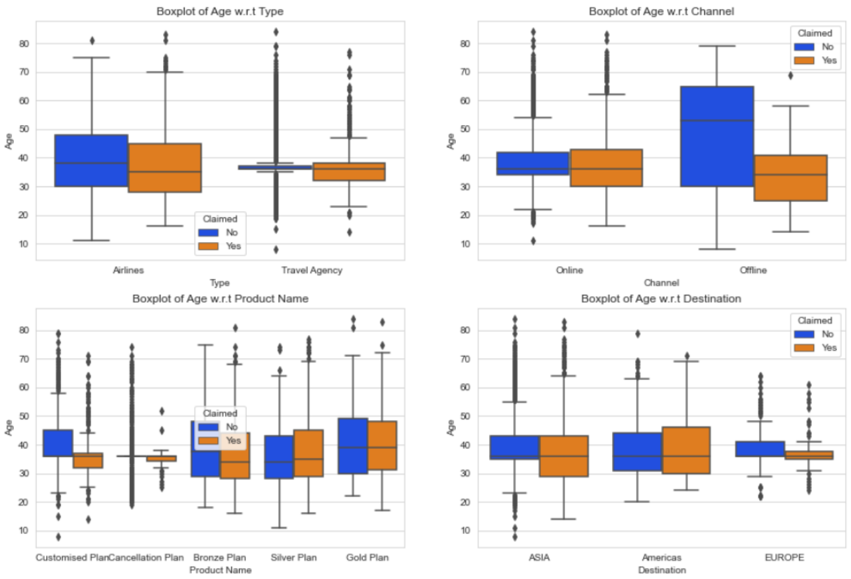
Hence, our assumption has been confirmed.

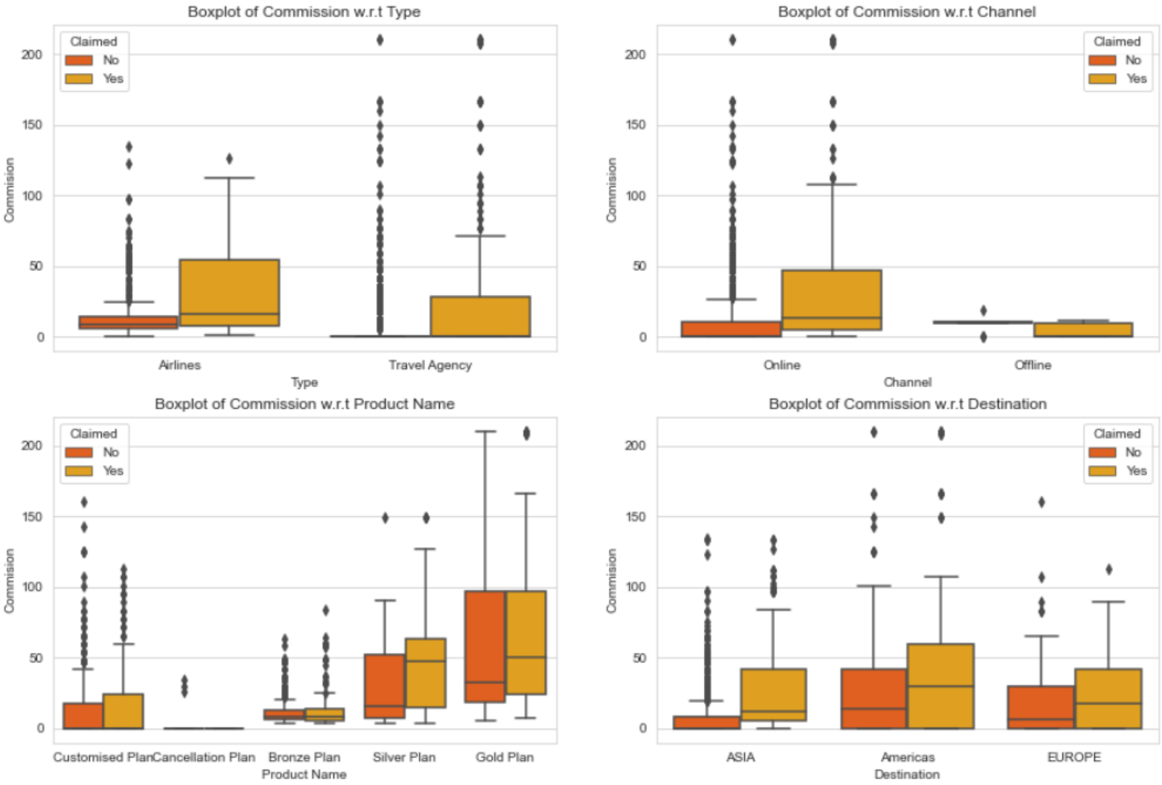
Plotting numerical variables w.r.t Claimed status



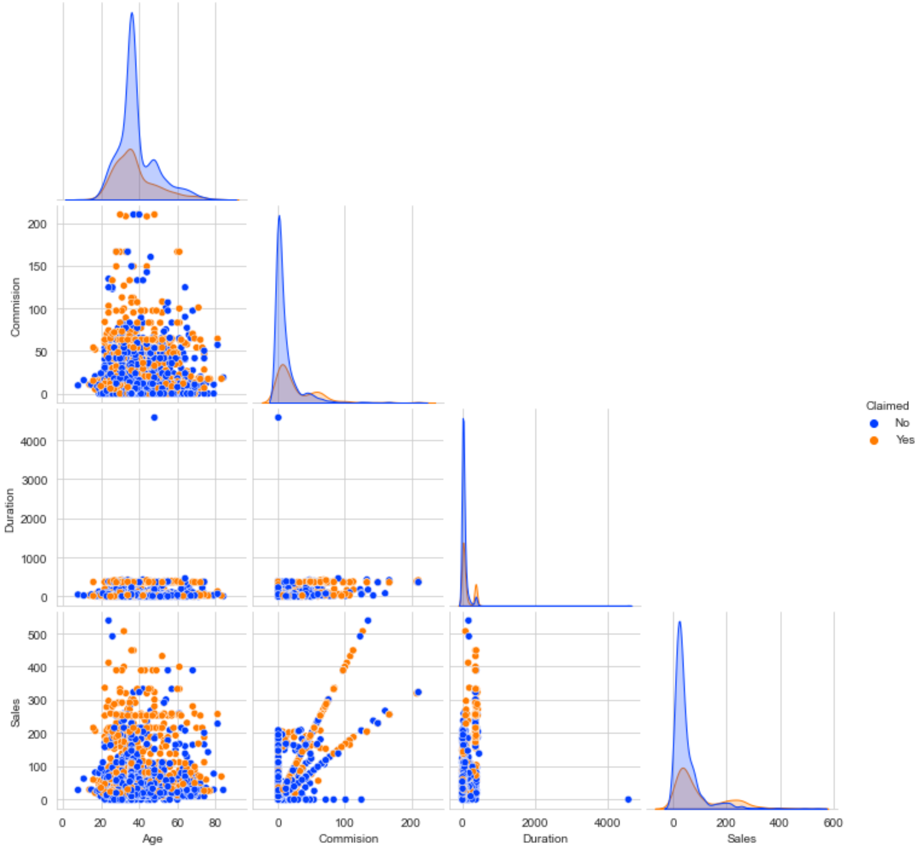
From the above boxplots below are the observations: Among the age, we see we have more customers who have claimed the insurance though we have more or less same spread for both categories. When we see commission in relation to claimed status, we see that customers who have claimed are more compared to those who have not claimed the insurance. We can hardly see any difference with respect to duration. Sales are higher for those who have claimed compared to those who have not claimed.

Bivariate Analysis



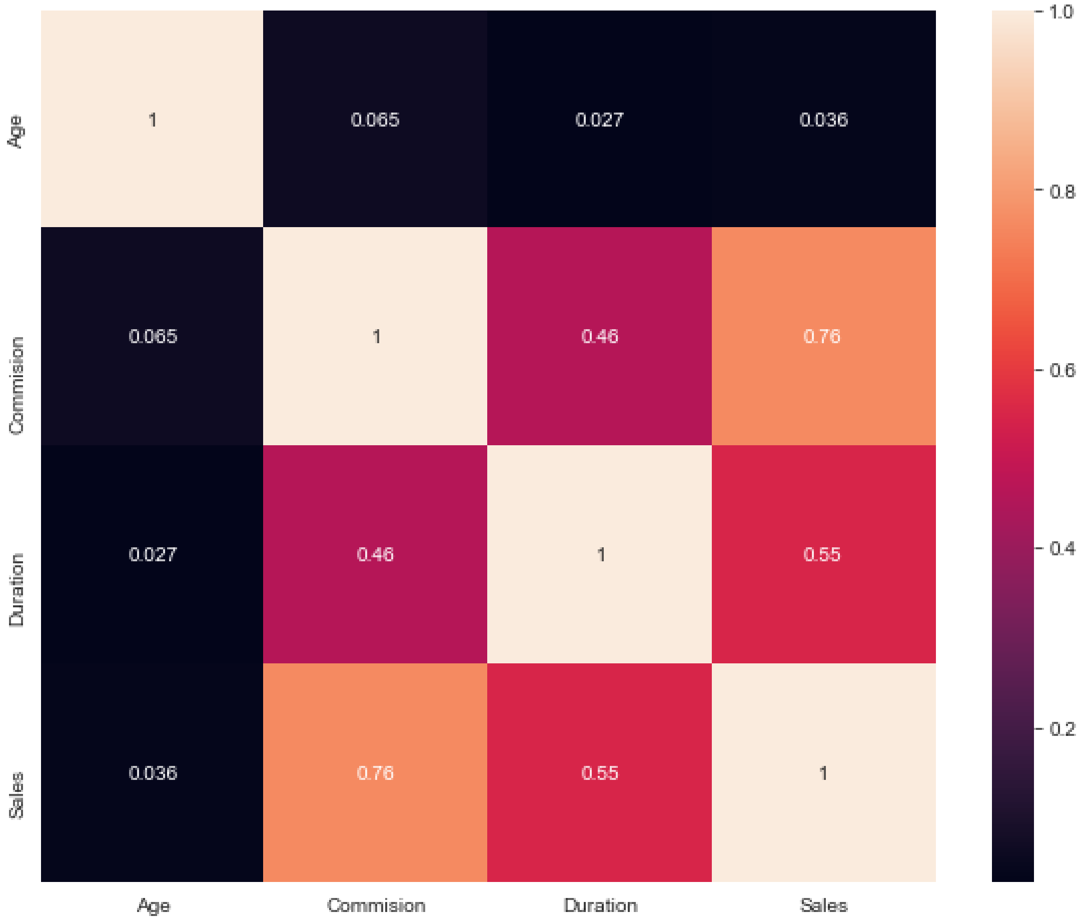

Boxplot of Commission w.r.t Type
Claimed
No
Yes
Commision
0
50
100
150
200
Airlines
Travel Agency
Type
Boxplot of Commission w.r.t Channel
Online
Offline
Channel
Boxplot of Commission w.r.t Product Name
Customised Plan
Cancellation Plan
Bronze Plan
Silver Plan
Gold Plan
Product Name
Boxplot of Commission w.r.t Destination
ASIA
Americas
EUROPE
Destination

Pairplot


Commision
Duration
Sales
Age
Claimed
No
Yes
0
20
40
60
80
100
200
2000
4000
400
600
1000
3000
150
50
300
500

From the above pairplot we can see that there is hardly any multicollinearity between the variables. Sales and Commission kind of show some positive relationship but need to check the correlation matrix to see how strong the relationship is.

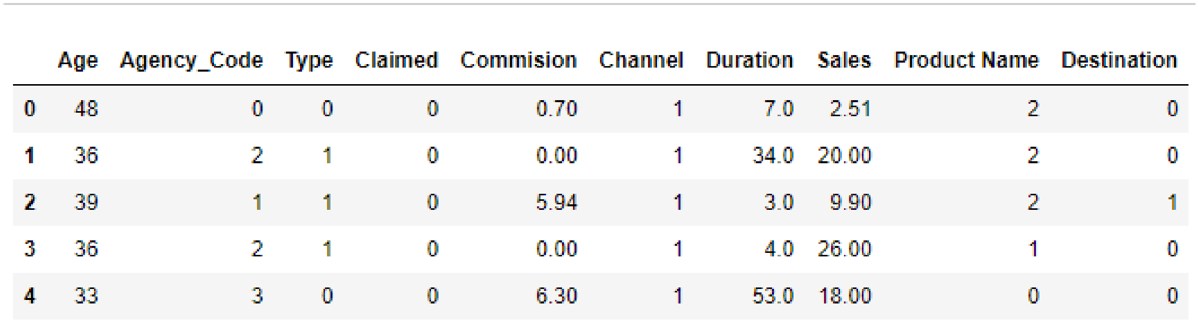
Correlation Heatmap



There is hardly any correlation between the variables. Sales and Commission have a positive correlation but they are not strong enough.

CART models only work on numerical data, hence let us encode the categorical variables into numerical codes. After encoding, we can see that the data only has numeric data types which we will use for CART model building

We can also see the head of the dataset.



	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	0	0	0	0.70	1	7.0	2.51	2	0
1	36	2	1	0	0.00	1	34.0	20.00	2	0
2	39	1	1	0	5.94	1	3.0	9.90	2	1
3	36	2	1	0	0.00	1	4.0	26.00	1	0
4	33	3	0	0	6.30	1	53.0	18.00	0	0

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

CART and Random Forest (RF) models would only work on numerical data. Hence let's encode the categorical variables and transform them to numeric data.

Building CART / Decision Tree Model

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

Classification And Regression Tree (CART) is a binary tree. It uses Gini index as the calculating criteria.

We will be using the sklearn model selection package to split the data into train and test data. We will use sklearn train_test_split package to split the data.

We will use sklearn DecisionTreeClassifier and GridSearchCV for building Decision Tree model.

Train data will be split into 70% and test data will be split into 30%.

For the given dataset, we have taken the below hyper parameters:

Criterion: Gini

Max_Depth: 10, 13, 15

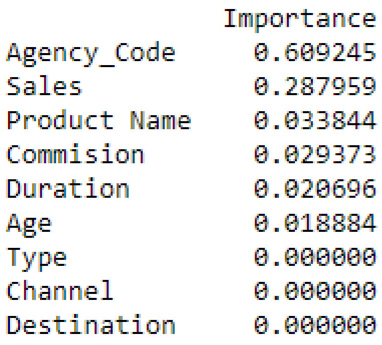
Min_Samples_Leaf: 10,50,100

Min_Samples_Split: 100,150,200

Cross Validation Iterations = 3

After applying Grid Search Cross Validation, the best parameters were as below: Criterion: Gini Max_Depth: 10 Min_Samples_Leaf: 50 Min_Samples_Split: 150

Now we have fit the best grid to the train and test. Let us see the feature importance.

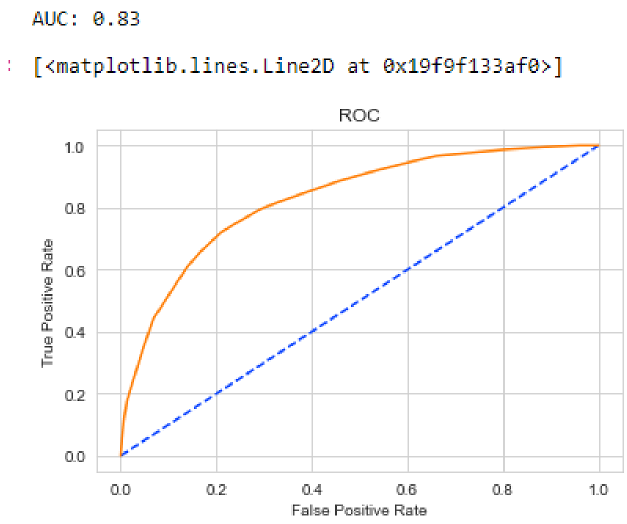


```
              Importance
Agency_Code     0.609245
Sales           0.287959
Product Name    0.033844
Commision       0.029373
Duration        0.020696
Age             0.018884
Type            0.000000
Channel         0.000000
Destination     0.000000
```

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 Evaluation – CART

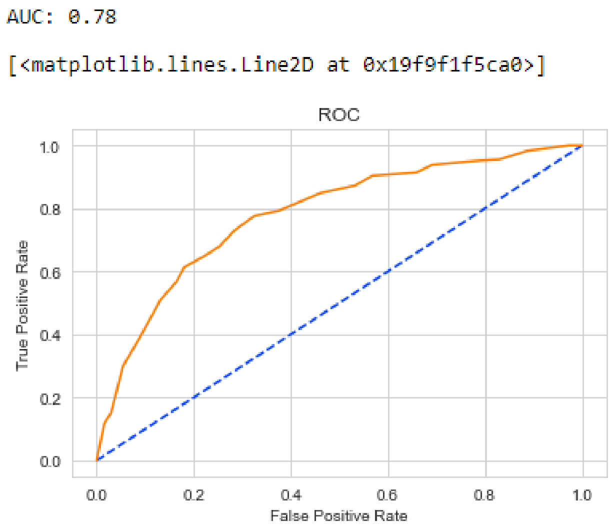
AUC and ROC for training data -CART



```
AUC: 0.83
```

```
[<matplotlib.lines.Line2D at 0x19f9f133af0>]
```

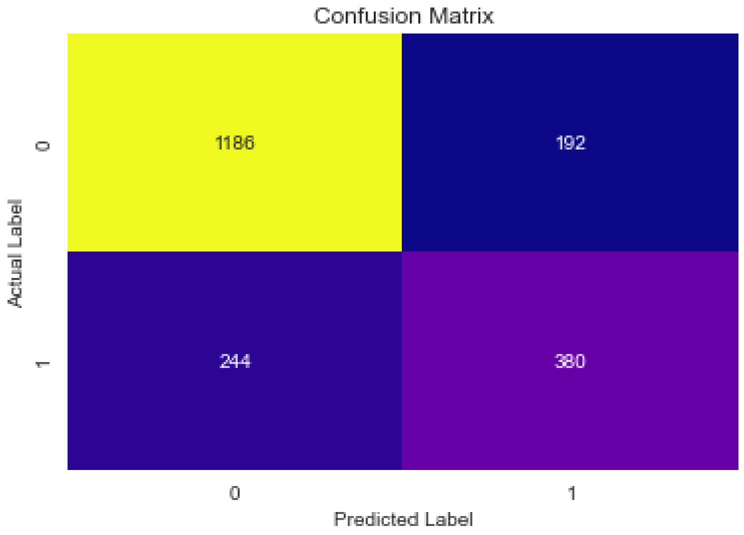
AUC and ROC for testing data – CART



```
AUC: 0.78
```

```
[<matplotlib.lines.Line2D at 0x19f9f1f5ca0>]
```

Confusion Matrix and Classification Report for training Data – CART

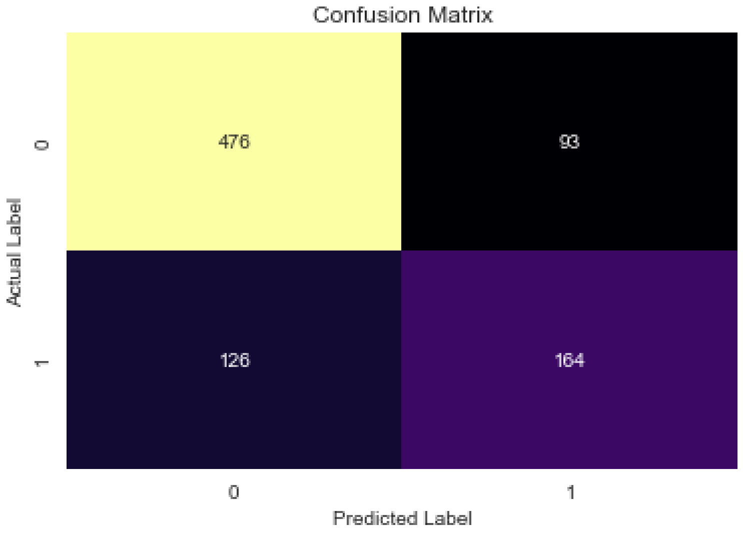


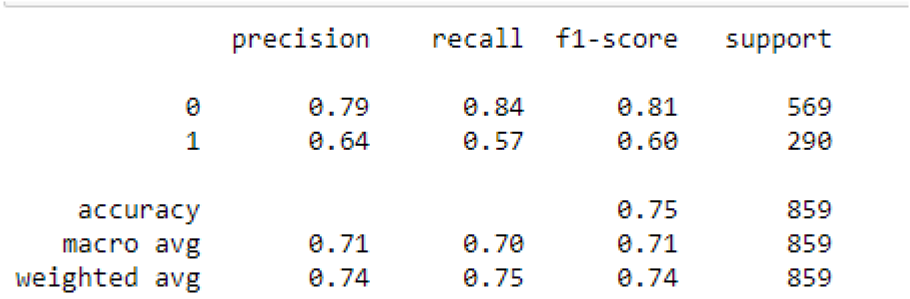
```
              precision    recall  f1-score   support

           0       0.83      0.86      0.84      1378
           1       0.66      0.61      0.64       624

    accuracy                           0.78      2002
   macro avg       0.75      0.73      0.74      2002
weighted avg       0.78      0.78      0.78      2002
```

Confusion Matrix and Classification Report for testing Data – CART





```
              precision    recall  f1-score   support

           0       0.79      0.84      0.81       569
           1       0.64      0.57      0.60       290

    accuracy                           0.75       859
   macro avg       0.71      0.70      0.71       859
weighted avg       0.74      0.75      0.74       859
```

Building Random Forest Model

Random Forests is an ensemble machine learning technique that combines several base models in order to produce one optimal predictive model. Random Forests are a collection of decision trees. In random forests, each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. Furthermore, when splitting each node during the construction of a tree, the best split is found either from all input features or a random subset of size max_features.

The purpose of these two sources of randomness is to decrease the variance of the forest estimator. Indeed, individual decision trees typically exhibit high variance and tend to overfit. The injected randomness in forests yield decision trees with somewhat decoupled prediction errors. By taking an average of those predictions, some errors can cancel out. Random forests achieve a reduced variance by combining diverse trees, sometimes at the cost of a slight increase in bias. In practice the variance reduction is often significant hence yielding an overall better model.

Random Forest Algorithm

Draw multiple random samples, with replacement, from the data. This sampling approach is called the bootstrap. Using a random subset of predictors at each stage, fit a classification (or regression) tree to each sample (and thus obtaina “forest”). Combine the predictions/classifications from the individual trees to obtain improved predictions. Use voting for classification and averaging for prediction.

Out-Of-Bag (OOB) Dataset

When we create a bootstrapped dataset, 1/3 of the original data does not end up in the bootstrapped dataset. This is called Out-Of-Bag dataset. OOB samples are used to measure how accurate our random forest is.

For the given dataset, we have taken the below hyper parameters:

Criterion: Gini

Max_Depth: 10

Max_Features: 5

Min_Samples_Leaf: 10

Min_Samples_Split: 100

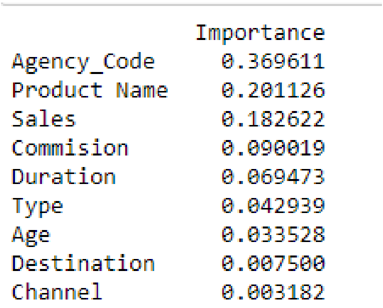
n_estimators: 201

Cross Validation Iterations = 3

After applying Grid Search Cross Validation, the best parameters were as below:

Max_Depth: 10 Max_Features: 5 Min_Samples_Leaf: 10 Min_Samples_Split: 100 n_estimators: 201 Cross Validation Iterations = 3

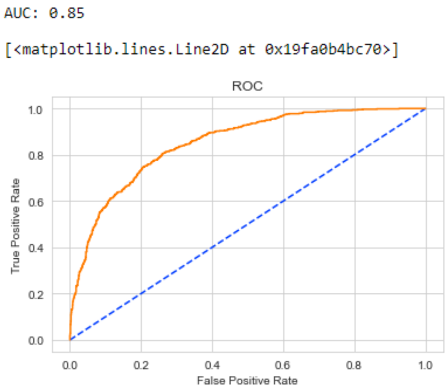
Now we have fit the best grid to the train and test. Let us see the feature importance.



```
              Importance
Agency_Code     0.369611
Product Name    0.201126
Sales           0.182622
Commision       0.090019
Duration        0.069473
Type            0.042939
Age             0.033528
Destination     0.007500
Channel         0.003182
```

Model Evaluation – Random Forest

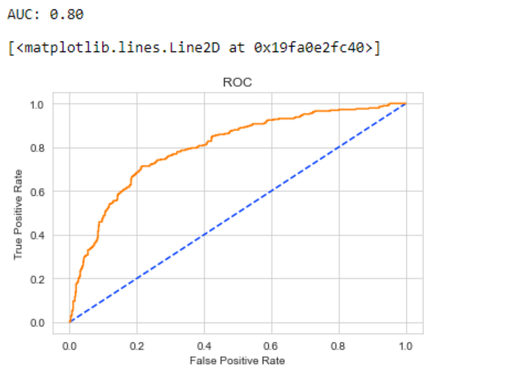
AUC and ROC for Training data –



```
AUC: 0.85

[<matplotlib.lines.Line2D at 0x19fa0b4bc70>]
```

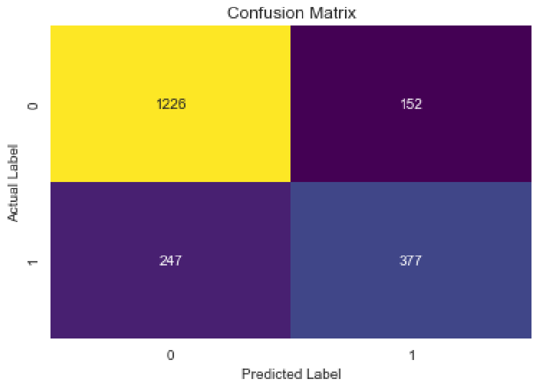
AUC and ROC for Testing data –

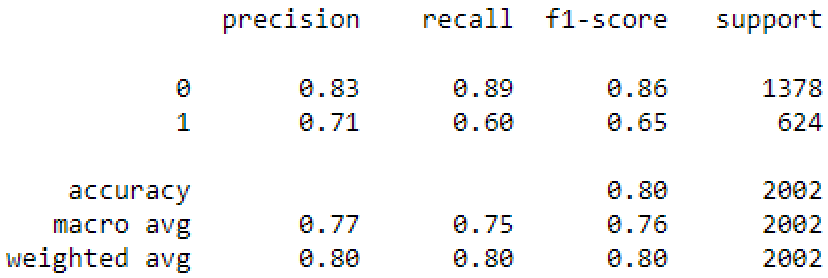


```
AUC: 0.80

[<matplotlib.lines.Line2D at 0x19fa0e2fc40>]
```

Confusion Matrix and Classification Report for Training Data – Random Forest

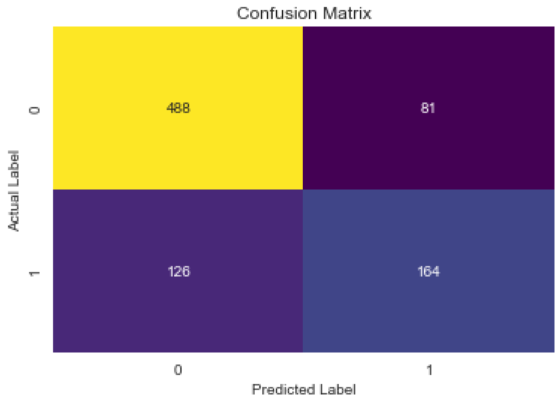


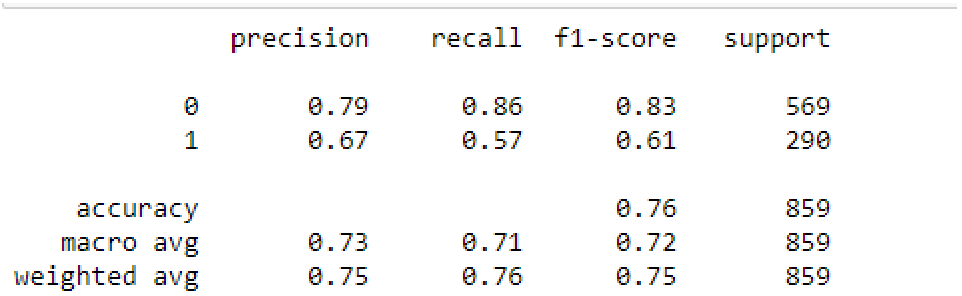

```
              precision    recall  f1-score   support

           0       0.83      0.89      0.86      1378
           1       0.71      0.60      0.65       624

    accuracy                           0.80      2002
   macro avg       0.77      0.75      0.76      2002
weighted avg       0.80      0.80      0.80      2002
```

Confusion Matrix and Classification Report for Testing data - Random Forest




```
              precision    recall  f1-score   support

           0       0.79      0.86      0.83       569
           1       0.67      0.57      0.61       290

    accuracy                           0.76       859
   macro avg       0.73      0.71      0.72       859
weighted avg       0.75      0.76      0.75       859
```

Building an Artificial Neural Network (ANN) Model

Artificial Neural Network (ANN) is machine learning algorithm that is roughly modeled around what is currently known about how the human brain functions. It models the relationship between a set of input signals and an output similar to a biological brain response to stimuli from sensory inputs. The brain uses a network of interconnected cells called neurons to provide learning capability. ANN uses a network of artificial neurons or nodes to solve challenging learning problems.

We will be using the sklearn neural network Multi-Layer Perceptron Classifier to build this model. ANNs are effective when the data is scaled. Hence, we have scaled the data using sklearn Standard Scalar package.

We have used the below parameters while building the model.

Hidden Layer Size: 500,1000,1500 Max_Iteration: 5000,10000 Solver: adam, sgd Tolerance: 0.001,0.01

After applying Grid Search Cross Validation, the best parameters were as below:

Hidden Layer Size: 1000

Max_Iteration: 5000

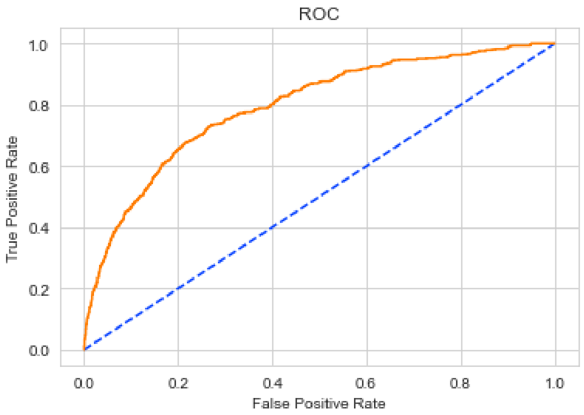
Solver: adam

Tolerance: 0.01

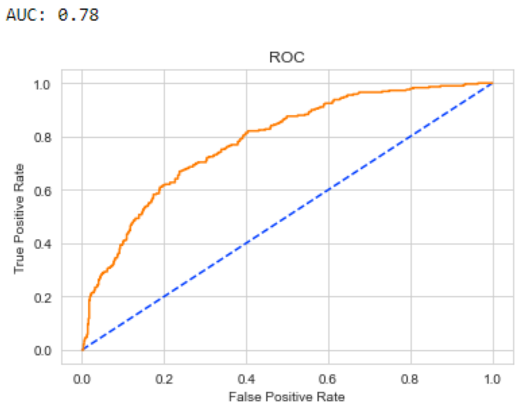
Model Evaluation – ANN

AOC and ROC for Training data – ANN

```
AUC: 0.80
```

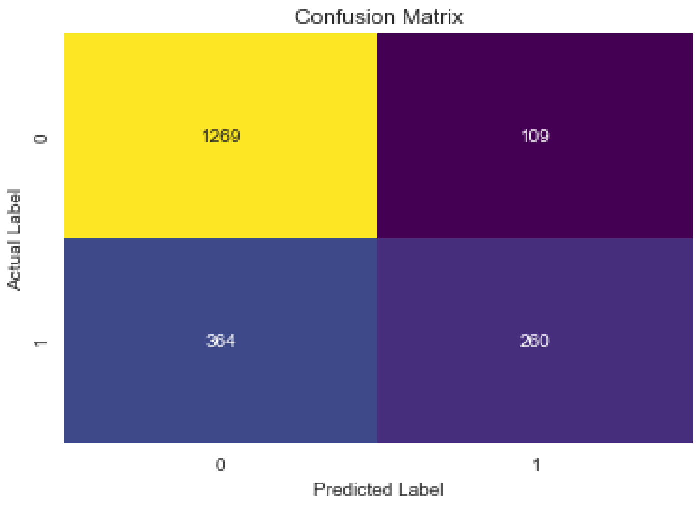


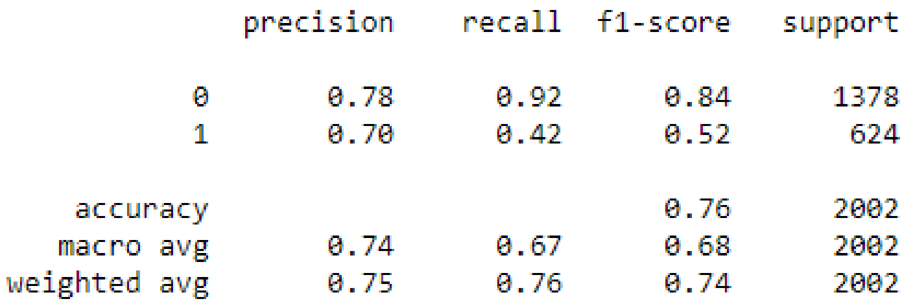
AOC and ROC for Testing data – ANN



```
AUC: 0.78
```

Confusion Matrix and Classification Report for Training data – ANN



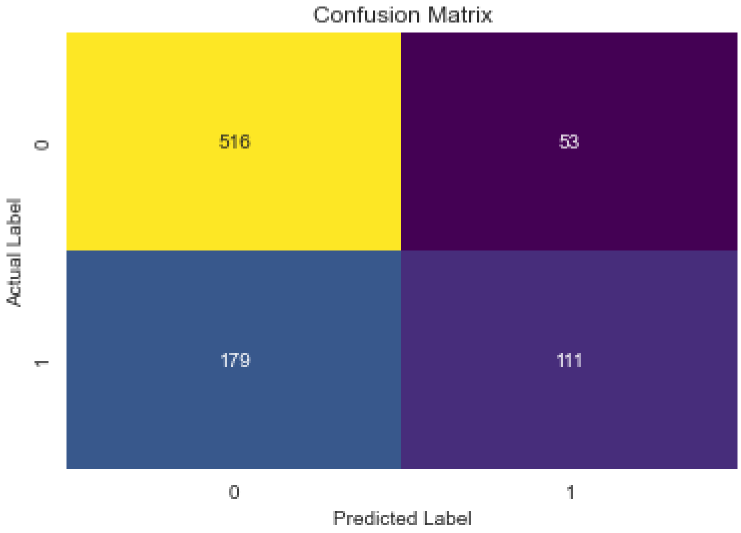


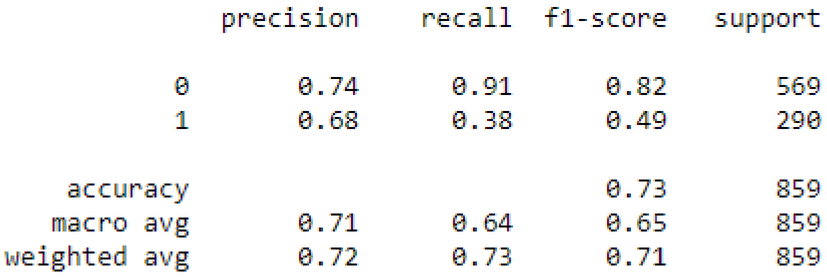
```
              precision    recall  f1-score   support

           0       0.78      0.92      0.84      1378
           1       0.70      0.42      0.52       624

    accuracy                           0.76      2002
   macro avg       0.74      0.67      0.68      2002
weighted avg       0.75      0.76      0.74      2002
```

Confusion Matrix and Classification Report for Testing data – ANN



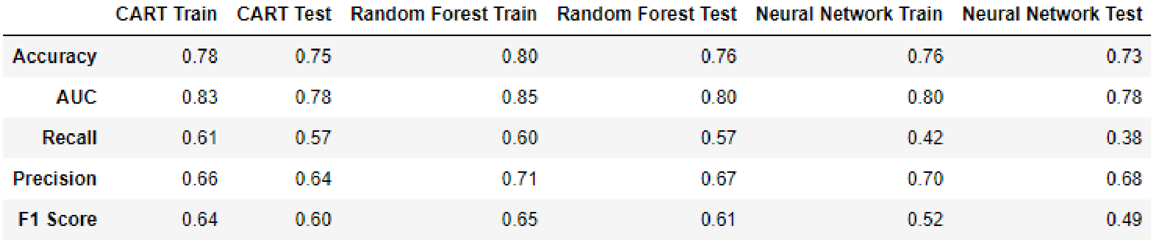

```
              precision    recall  f1-score   support

           0       0.74      0.91      0.82       569
           1       0.68      0.38      0.49       290

    accuracy                           0.73       859
   macro avg       0.71      0.64      0.65       859
weighted avg       0.72      0.73      0.71       859
```

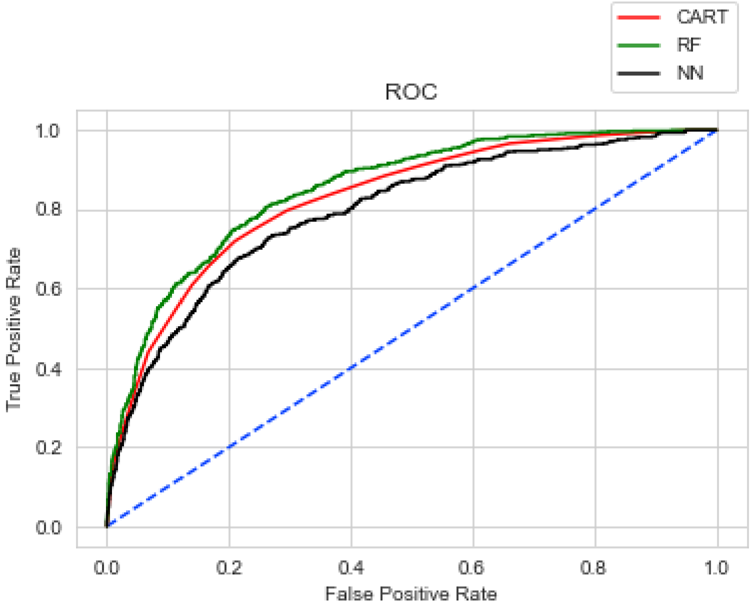
2.4 Final Model: Compare all the models and write an inference which model is best/optimized.

Below is the comparison of the 3 models with their recall, precision, accuracy, f1-score and AUC (Area Under the curve)

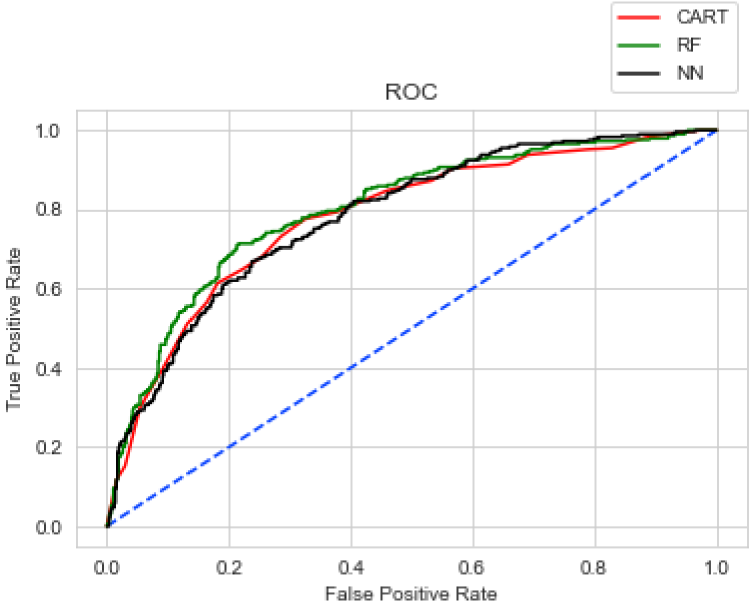


	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.78	0.75	0.80	0.76	0.76	0.73
AUC	0.83	0.78	0.85	0.80	0.80	0.78
Recall	0.61	0.57	0.60	0.57	0.42	0.38
Precision	0.66	0.64	0.71	0.67	0.70	0.68
F1 Score	0.64	0.60	0.65	0.61	0.52	0.49

ROC Curve for the 3 models on the Training data



ROC Curve for the 3 models on the Testing data



Out of the 3 models, Random Forest has slightly better performance than the CART and ANN model.

Overall, all the 3 models are reasonably stable enough to be used for making any future predictions. From CART and Random Forest Model, the variable Agency_Code is found to be the most useful feature amongst all other features for predicting if a customer has claimed the insurance or not.

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

Based on the above 3 models, below are the recommendations.

1. Out of 2861 observations after removing duplicates, the proportion of claimed status “No” is 1947 whereas “Yes” is 914 which clearly shows that No is 68.05% of the total data and Yes is 31.94% of the total data. Hence there is a clear imbalance in the proportion of class labels.

2. Business needs to collect more data in order to balance the proportions and thereby build an effective model.

3. The business should focus on Agency_Codes which is the main attribute on which these models are built. They should look at forming a better relationship with the agencies to collaborate better. The agency “C2B” has a very high claim rate.

4. Plans such as “cancellation plan” and “customization plan” have a lower claim rate. So, the business can focus on these plans and promote them further.

5. Would recommend the business to promote plans that have a lower amount of sale and lower duration as the claim rate is also low.

------------Thanks------------