[Year]

# Project - Data Mining

## Clustering & CART-RF-ANN

S NITIN KUMAR

[COMPANY NAME] | [Company address]

# 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.

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

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

```
df.isnull().sum()
```

```
spending                      0
advance_payments              0
probability_of_full_payment   0
current_balance               0
credit_limit                  0
min_payment_amt               0
max_spent_in_single_shopping  0
dtype: int64
```

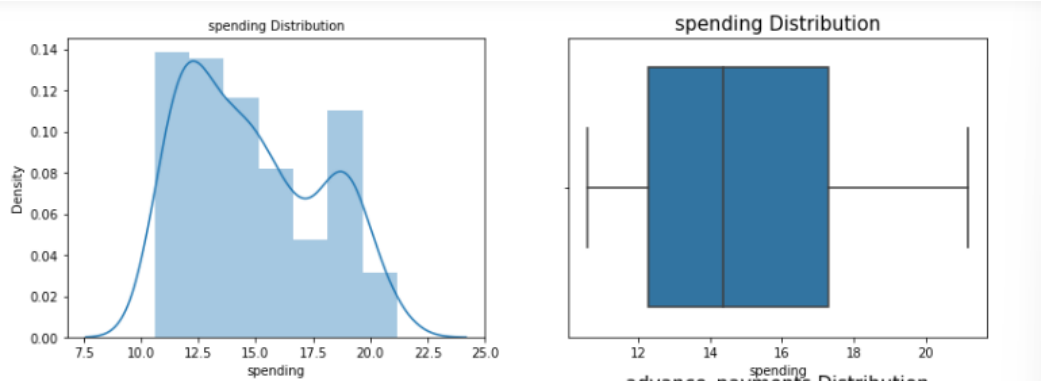
```
df.duplicated().sum()
```

```
0
```

Data set contain 7 different attributes of credict data. There are 210 entrie. All data types are of type numerical and there are no missing values and duplicated values in the data set data looks good.

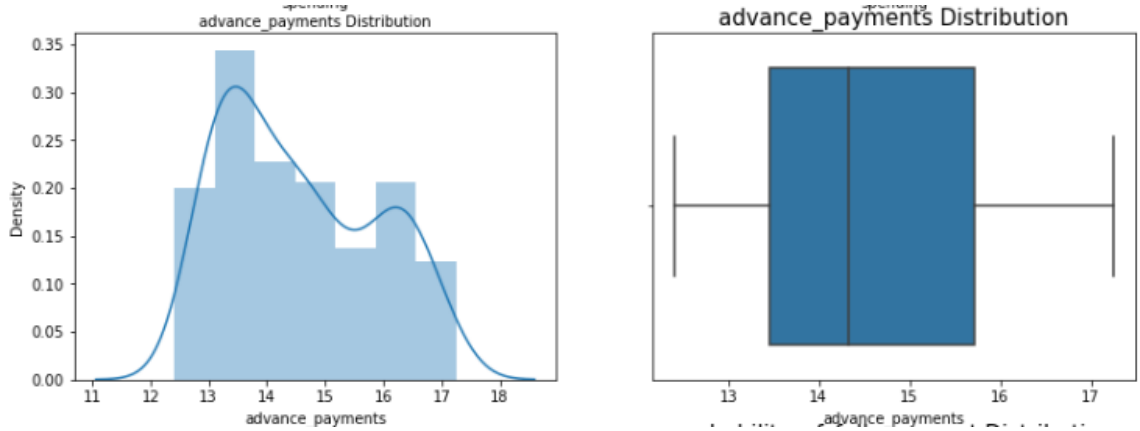
## Univariant analysis

**spending**



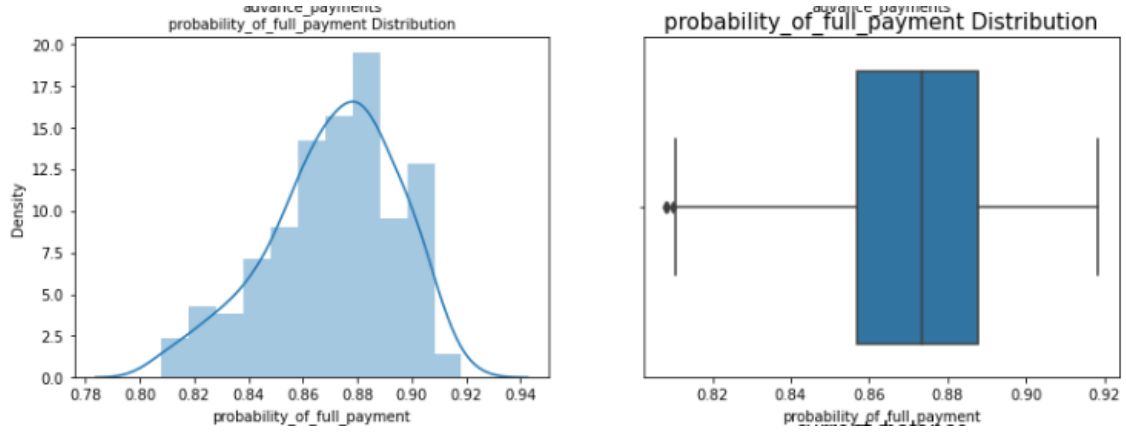

From the boxplot we can see that there are no outliers in spending ad the average spending is in between 12 to 17

Advance payment distribution



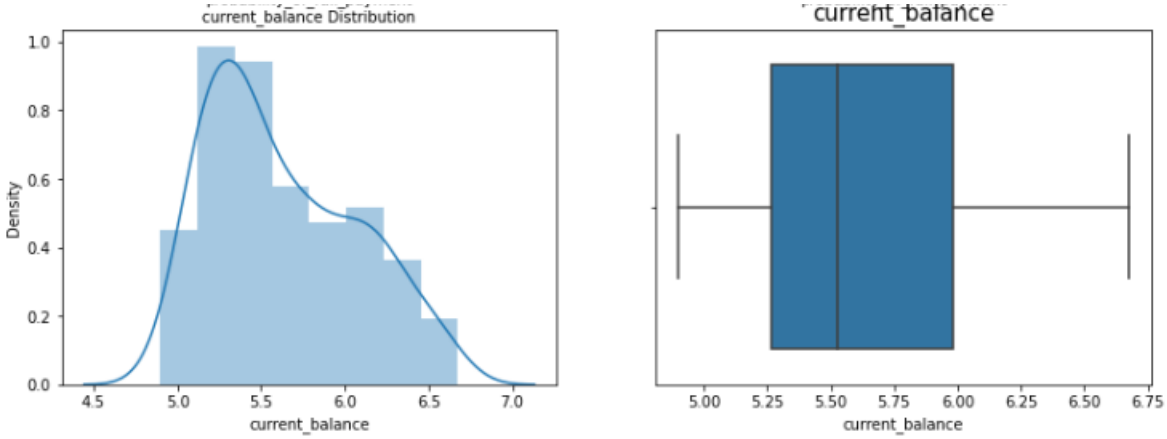

From the boxplot of Advance payment distribution, we can see that there are no outliers in Advance payment distribution and the average Advance payment distribution is in between 13.5 to 15.7

**Probability of full payment distribution**



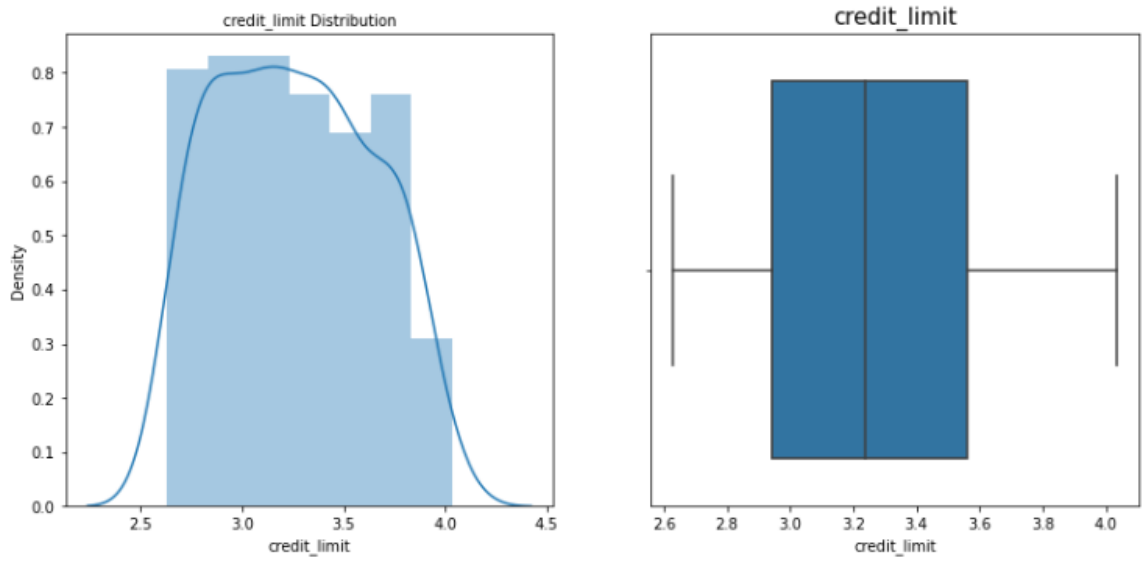

From the boxplot of Probability of full payment distribution, we can see that there are negligible outliers in Probability of full payment distribution which may not effect to the model and the average Probability of full payment distribution is in between 0.85 to 0.89

**Current balance distribution**



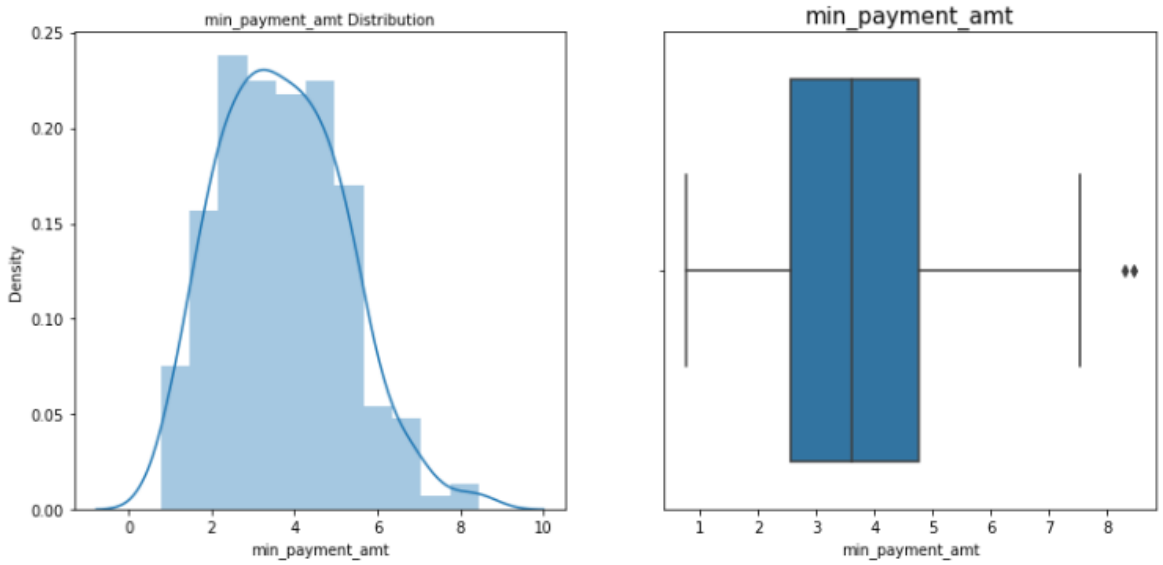

From the boxplot of Current balance distribution, we can see that there are no outliers in Current balance distribution and the average Current balance distribution is in between 5.25 to 6.00

**Credit limit distribution**



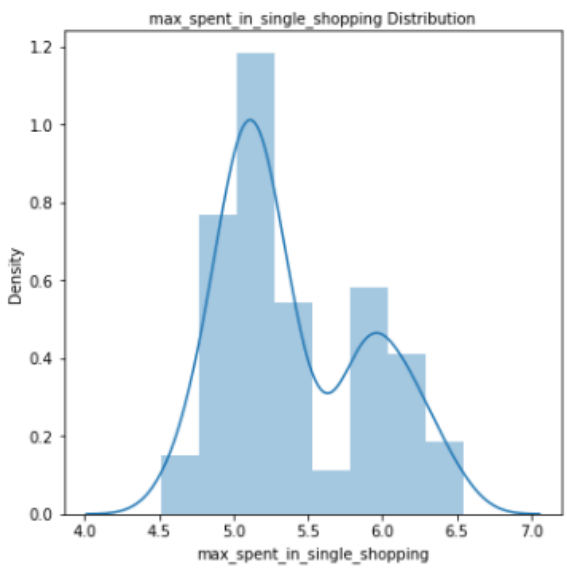

From the boxplot of Credit limit distribution, we can see that there are no outliers in Credit limit distribution and the average Credit limit distribution is in between 2.9 to 3.6

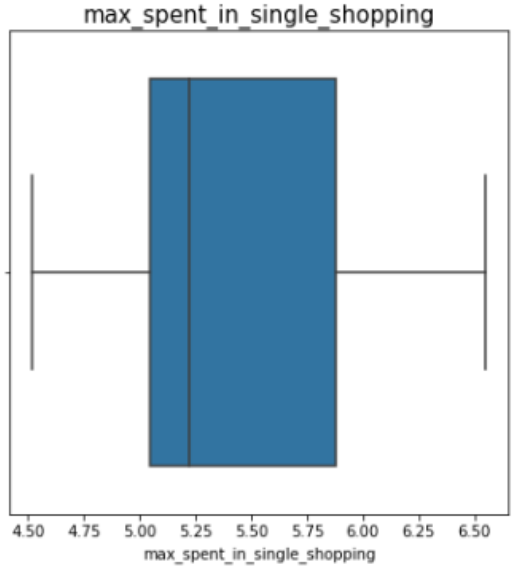
**Minimum payment amount distribution**




From the boxplot of Minimum payment amount distribution, we can see that there are few outliers in Minimum payment amount distribution and the Minimum payment amount distribution is in between 2.5 to 4.8

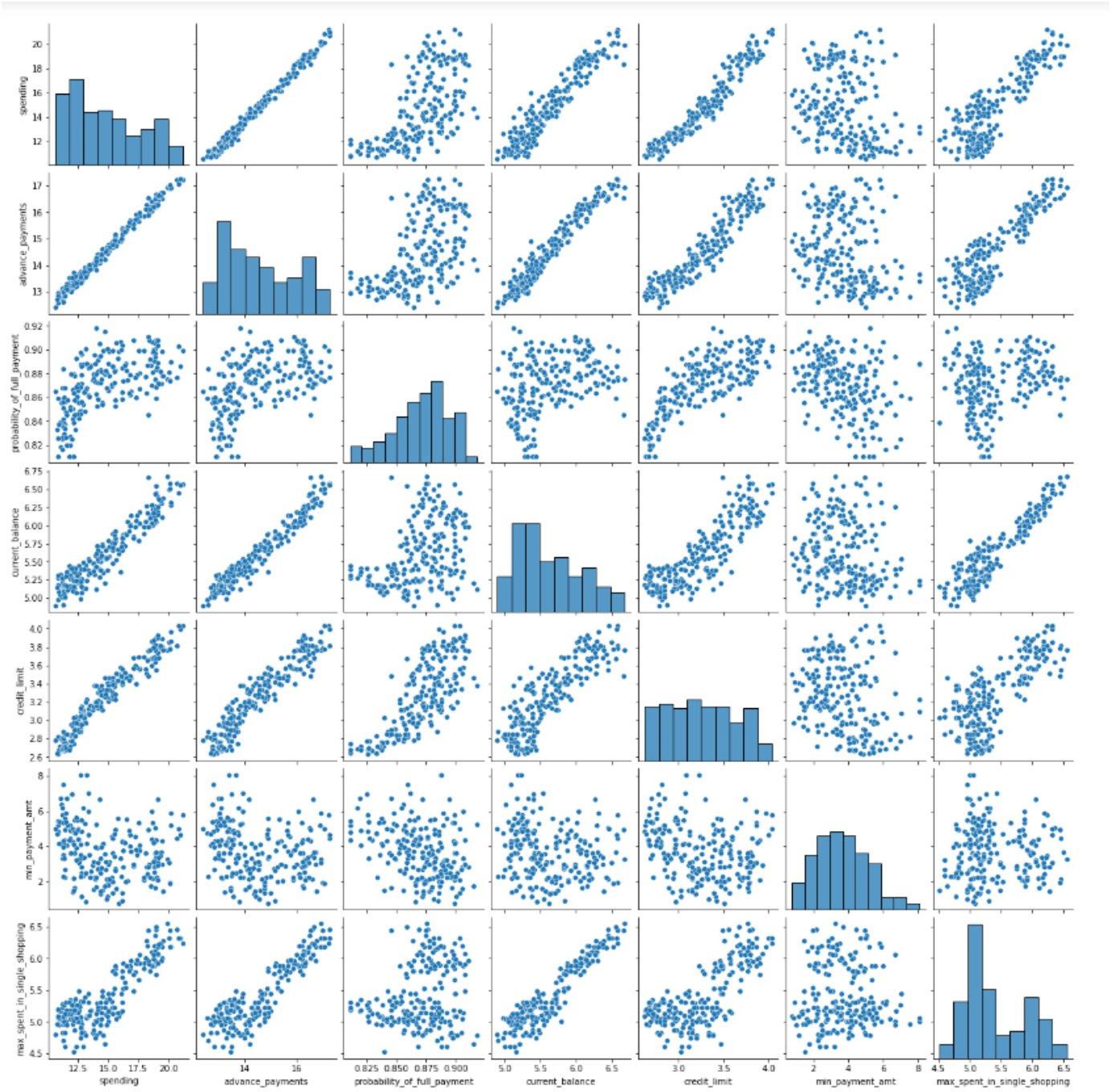
**Maximum spent in single shoping distrubation**




From the boxplot of Maximum spent in single shoping distrubation , we can see that there are no outliers in Maximum spent in single shoping distrubation and the average Maximum spent in single shoping distrubation is in between 5.1 to 5.85

**Pair plot**




clearly from the above bar plot & pair plot we can see that all the attributes are not scaled and pre scaling will be required before performing clustering.

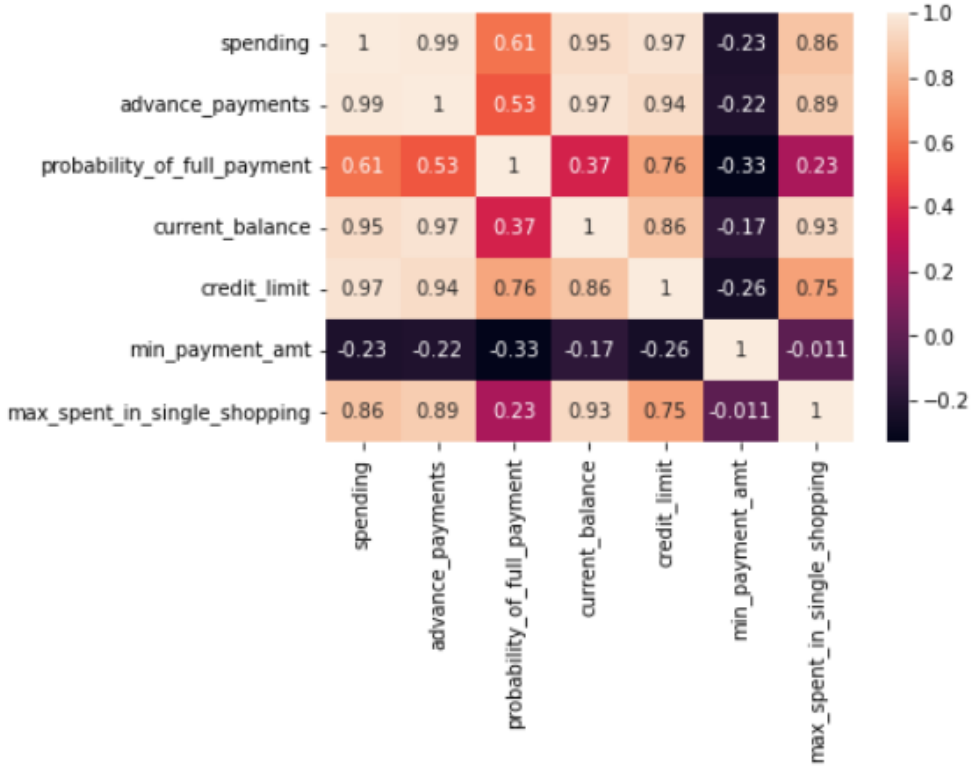
-Strong positive correlation between

- max_spent_in_single_shopping & current balance
- spending & advance payments,
- advance payments & current balance,
- credit limit & spending
- spending & current balance
- credit limit & advance payments

**Heat map corelation values**

```
df_corr=df.corr()
df_corr
```

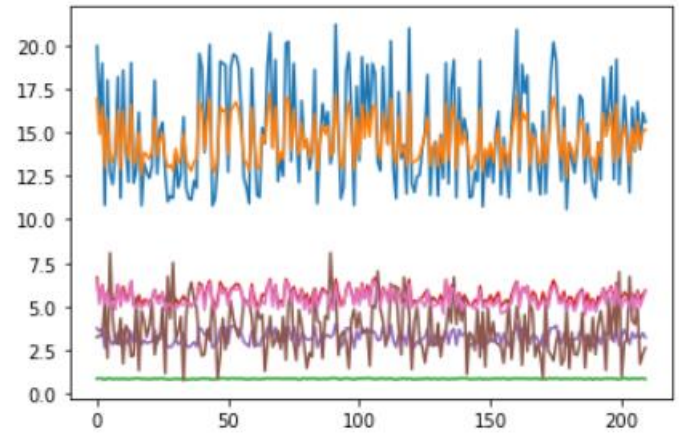
| | spending | advance_payments | probability_of_full_payment | current_balance | credit_limit | min_payment_amt | max_spent_in_single_shopping |
|---|---|---|---|---|---|---|---|
| **spending** | 1.000000 | 0.994341 | 0.608288 | 0.949985 | 0.970771 | -0.229572 | 0.863693 |
| **advance_payments** | 0.994341 | 1.000000 | 0.529244 | 0.972422 | 0.944829 | -0.217340 | 0.890784 |
| **ıability_of_full_payment** | 0.608288 | 0.529244 | 1.000000 | 0.367915 | 0.761635 | -0.331471 | 0.226825 |
| **current_balance** | 0.949985 | 0.972422 | 0.367915 | 1.000000 | 0.860415 | -0.171562 | 0.932806 |
| **credit_limit** | 0.970771 | 0.944829 | 0.761635 | 0.860415 | 1.000000 | -0.258037 | 0.749131 |
| **min_payment_amt** | -0.229572 | -0.217340 | -0.331471 | -0.171562 | -0.258037 | 1.000000 | -0.011079 |
| **ent_in_single_shopping** | 0.863693 | 0.890784 | 0.226825 | 0.932806 | 0.749131 | -0.011079 | 1.000000 |



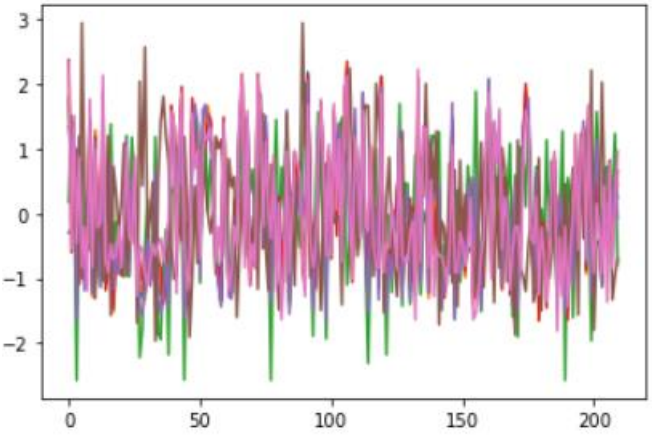

## 1.2 Do you think scaling is necessary for clustering in this case? Justify

Scaling needs to be done as the values of the variables are different. spending, advance payments are in different values and this may get more weightage. Also have shown below the plot of the data prior and after scaling. Scaling will have all the values in the relative same range.

Before scaling

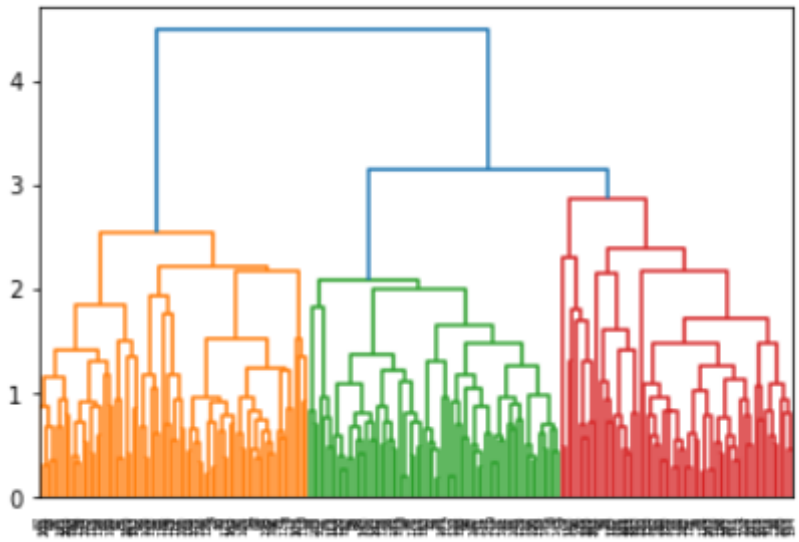



After scaling

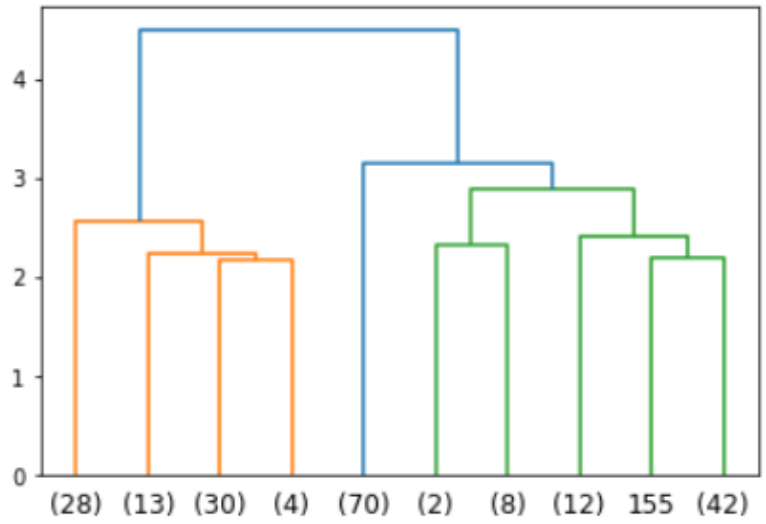



1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them

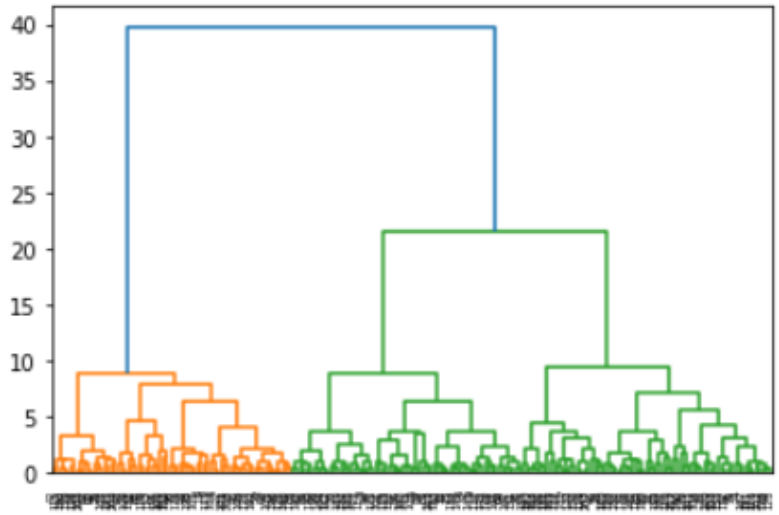
Creating dendrogram using linkage method



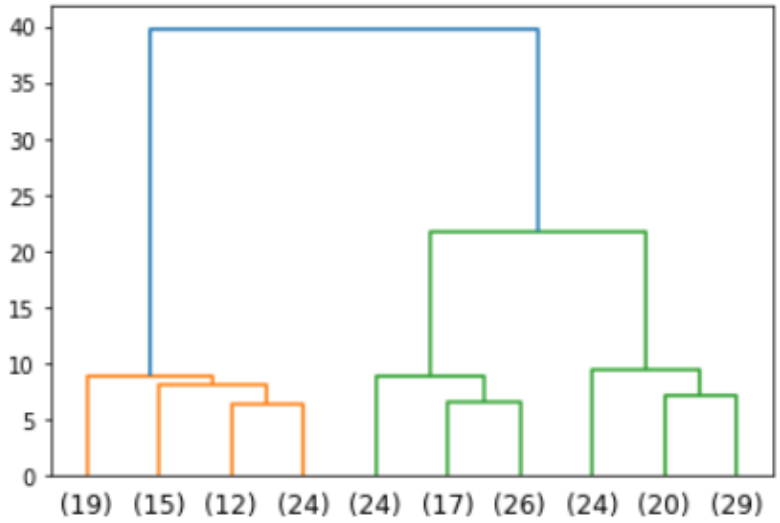

Selecting last 10 levels in dendron




Creating dendrogram using ward-linkage method



Selecting last 10 levels from ward-linkage




No of 3 clusters from the dendrogram in linkage method

```
1    75
2    70
3    65
Name: clusters-3, dtype: int64
```

No of 3 clusters from the dendrogram in linkage method

```
1    70
2    67
3    73
Name: clusters-3, dtype: int64
```

Both the 2 method are almost similar means , minor variation, which we know it occurs.

We for cluster grouping based on the dendrogram, 3 or 4 looks good. Did the further analysis, and based on the dataset had gone for 3 group cluster solution based on the hierarchical clustering

Also in real time, there could have been more variables value captured - tenure, BALANCE_FREQUENCY, balance, purchase, instalment of purchase, others.

And three group cluster solution gives a pattern based on high/medium/low spending with max_spent_in_single_shopping (high value item) and probability_of_full_payment(payment made).

The 3 identical cluster groups that are formed and shown in the below figure

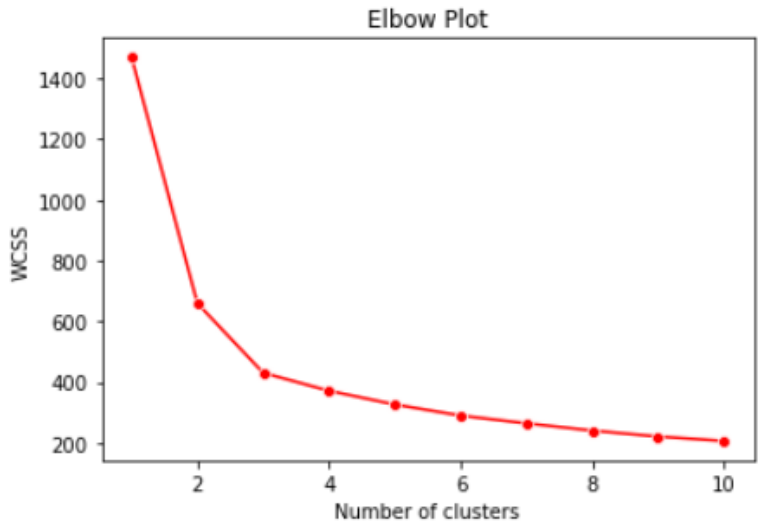
| clusters-3 | 1 | 2 | 3 |
|---|---|---|---|
| spending | 18.371429 | 11.872388 | 14.199041 |
| advance_payments | 16.145429 | 13.257015 | 14.233562 |
| probability_of_full_payment | 0.884400 | 0.848155 | 0.879190 |
| current_balance | 6.158171 | 5.238940 | 5.478233 |
| credit_limit | 3.684629 | 2.848537 | 3.226452 |
| min_payment_amt | 3.639157 | 4.940302 | 2.612181 |
| max_spent_in_single_shopping | 6.017371 | 5.122209 | 5.086178 |
| Freq | 70.000000 | 67.000000 | 73.000000 |

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.

When we Apply k-means clusters on scaled data for the range from 1 to 10 the result obtained in shown below

```
[1469.9999999999995,
 659.1474009548498,
 430.29848175122294,
 371.0356644664014,
 325.97412847298756,
 289.45524862464816,
 263.859944426353,
 239.9444663501791,
 220.5935394610811,
 205.76334196787008]
```

The number of best clusters that we wanted to choose can be decided from the elbow plot




From the above figure we can decide to go with 3 clusters

```
silhouette_score(df_scaled,labels_3)
```

```
0.4008059221522216
```

From SC Score, the number of optimal clusters is 3

Silhouette score is used to evaluate the quality of clusters created using clustering algorithms such as K-Means in terms of how well samples are clustered with other samples that are similar to each other and from the above score we can see that the clusters are formed good

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

Final K-means clusters for 3 groups is show in figure

| K_clusters | 0 | 1 | 2 |
|---|---|---|---|
| spending | 18.495373 | 11.856944 | 14.437887 |
| advance_payments | 16.203433 | 13.247778 | 14.337746 |
| probability_of_full_payment | 0.884210 | 0.848330 | 0.881597 |
| current_balance | 6.175687 | 5.231750 | 5.514577 |
| credit_limit | 3.697537 | 2.849542 | 3.259225 |
| min_payment_amt | 3.632373 | 4.733892 | 2.707341 |
| max_spent_in_single_shopping | 6.041701 | 5.101722 | 5.120803 |
| sil_width | 0.468077 | 0.399556 | 0.338593 |
| Freq | 67.000000 | 72.000000 | 71.000000 |

Cluster Group Profiles

- Group 1 : High Spending
- Group 3 : Medium Spending

- Group 2 : Low Spending

### Promotional strategies for each cluster

#### Group 1: High Spending Group

- Giving any reward points might increase their purchases.

- maximum max_spent_in_single_shopping is high for this group, so can be offered discount/offer on next transactions upon full payment

- Increase their credit limit and

- Increase spending habits

- Give loan against the credit card, as they are customers with good repayment record.

- Tie up with luxury brands, which will drive more one_time_maximun spending

#### Group 3: Medium Spending Group¶

- They are potential target customers who are paying bills and doing purchases and maintaining comp aratively good credit score. So we can increase credit limit or can lower down interest rate.

- Promote premium cards/loyalty cars to increase transactions.

- Increase spending habits by trying with premium ecommerce sites, travel portal, travel airlines/hotel , as this will encourage them to spend more

#### Group 2: Low Spending Group¶

- customers should be given remainders for payments. Offers can be provided on early payments to i mprove their payment rate.

- Increase their spending habits by tying up with grocery stores, utilities (electricity, phone, gas, others )

# Problem 2: CART-RF-ANN

An Insurance firm providing tour insurance is facing higher claim frequency. The management decides t o collect data from the past few years. You are assigned the task to make a model which predicts the cl aim status and provide recommendations to management. Use CART, RF & ANN and compare the mo dels' 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)

7. Destination of the tour (Destination)
8. Amount of sales of tour insurance policies (Sales)
9. The commission received for tour insurance firm (Commission)
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).

| | 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 |

From the data we can observe that

- 10 variables

- Age, Commission, Duration, Sales are numeric variable

- rest are categorial variables

- 3000 records, no missing one

- 9 independent variable and one target variable – Claimed

Checking for the missing values

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

There are no missing values in the data set

Checking for duplicates

```
df_in.duplicated().sum()
```

```
139
```

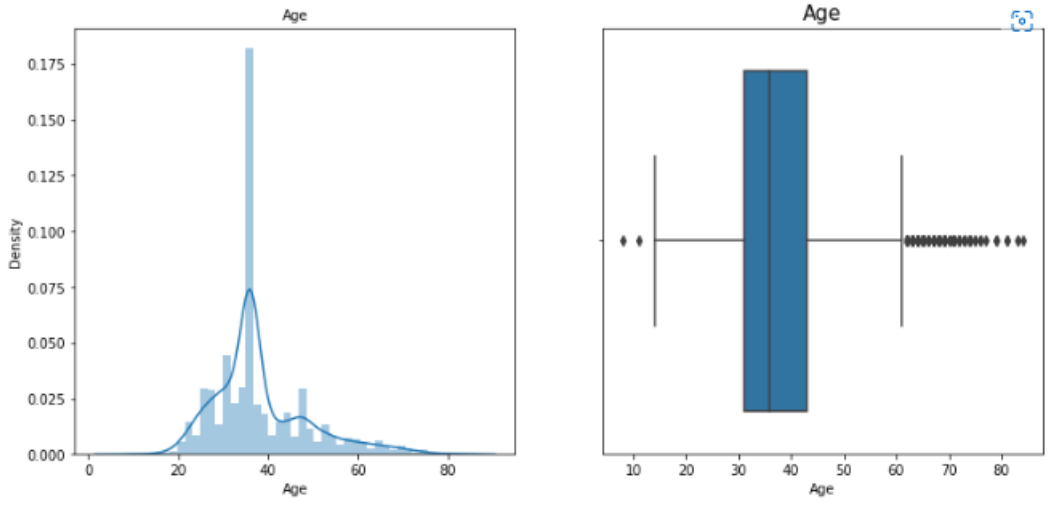
| | Age | Agency_Code | Type | Claimed | Commision | Channel | Duration | Sales | Product Name | Destination |
|---|---|---|---|---|---|---|---|---|---|---|
| **63** | 30 | C2B | Airlines | Yes | 15.0 | Online | 27 | 60.0 | Bronze Plan | ASIA |
| **329** | 36 | EPX | Travel Agency | No | 0.0 | Online | 5 | 20.0 | Customised Plan | ASIA |
| **407** | 36 | EPX | Travel Agency | No | 0.0 | Online | 11 | 19.0 | Cancellation Plan | ASIA |
| **411** | 35 | EPX | Travel Agency | No | 0.0 | Online | 2 | 20.0 | Customised Plan | ASIA |
| **422** | 36 | EPX | Travel Agency | No | 0.0 | Online | 5 | 20.0 | Customised Plan | ASIA |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **2940** | 36 | EPX | Travel Agency | No | 0.0 | Online | 8 | 10.0 | Cancellation Plan | ASIA |
| **2947** | 36 | EPX | Travel Agency | No | 0.0 | Online | 10 | 28.0 | Customised Plan | ASIA |
| **2952** | 36 | EPX | Travel Agency | No | 0.0 | Online | 2 | 10.0 | Cancellation Plan | ASIA |
| **2962** | 36 | EPX | Travel Agency | No | 0.0 | Online | 4 | 20.0 | Customised Plan | ASIA |
| **2984** | 36 | EPX | Travel Agency | No | 0.0 | Online | 1 | 20.0 | Customised Plan | ASIA |

139 rows × 10 columns

Though it shows there are 139 records, but it can be of different customers, there is no customer ID o r any unique identifier, so I am not dropping them off.

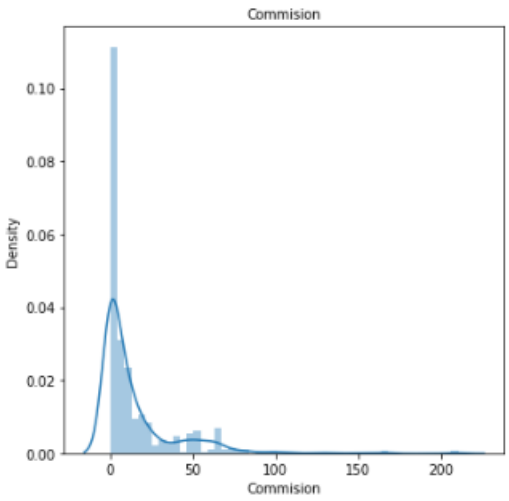
## Univariant analysis

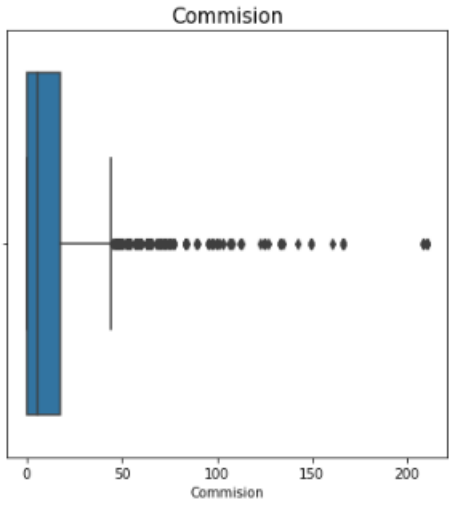
### Age




Minimum Age: 8
Maximum Age: 84
Mean value: 38.091
Median value: 36.0
Standard deviation: 10.463518245377944

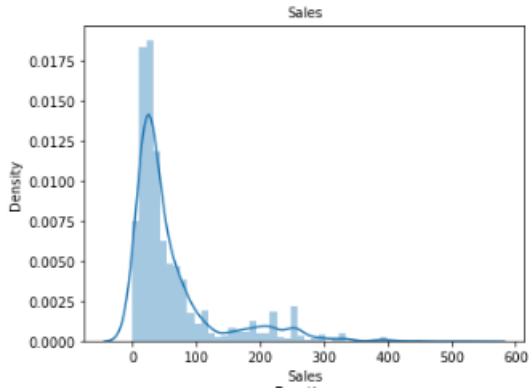
## Commision:

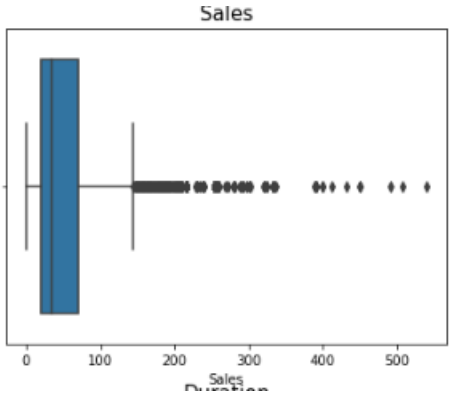



Minimum Commission: 0.0
Maximum Commission: 210.21
Mean value: 14.529203333333266
Median value: 4.63
Standard deviation: 25.48145450662553
Null values: False

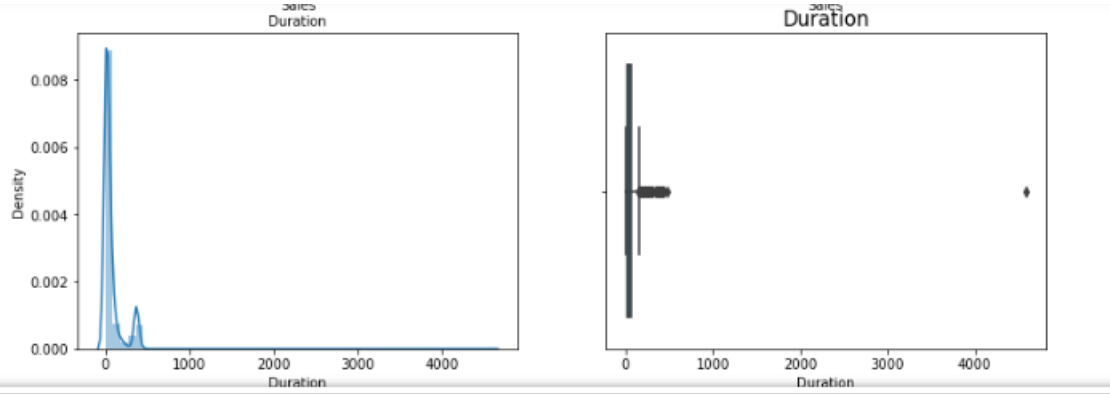
## Sales



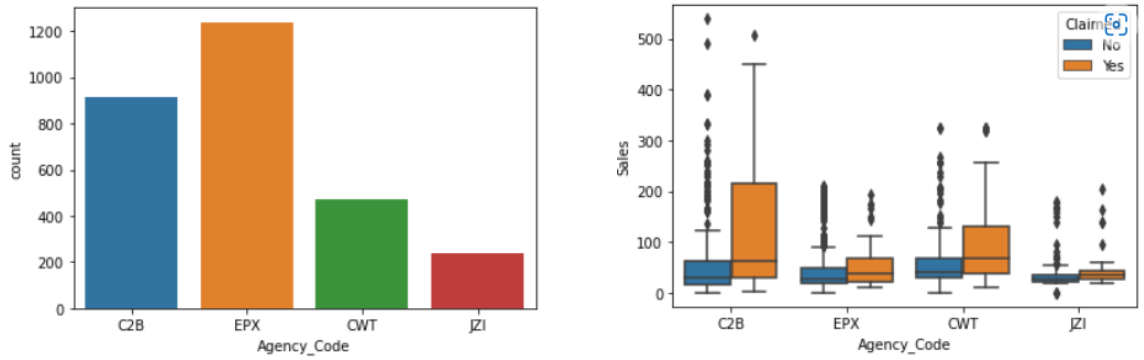

Minimum Sales: 0.0
Maximum Sales: 539.0
Mean value: 60.24991333333344
Median value: 33.0
Standard deviation: 70.73395353143047
Null values: False

## Duration

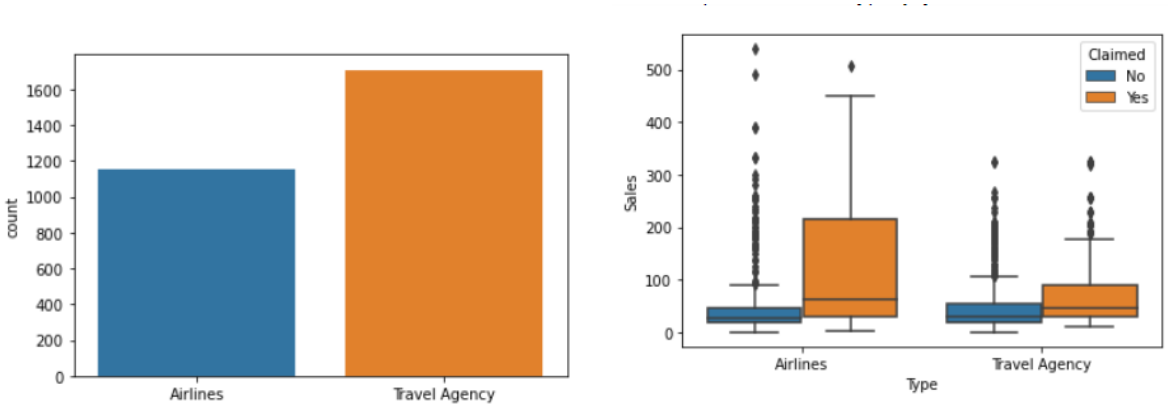



Minimum Duration: -1
Maximum Duration: 4580
Mean value: 70.00133333333333
Median value: 26.5
Standard deviation: 134.05331313253495
Null values: False

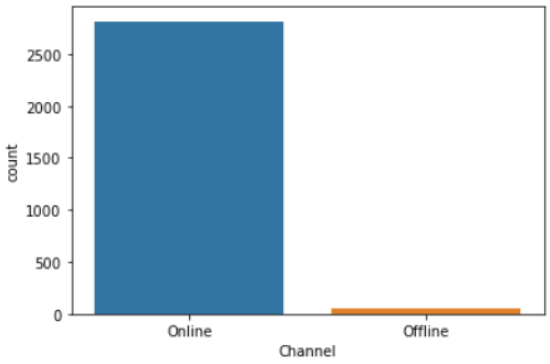
## Agency codes:

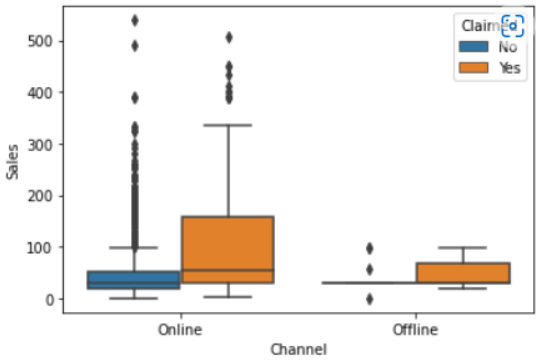



From the above figure we can say that the epx is popular or holds high number of proportions people have claimed more in C2B agency and the lowest in JZI

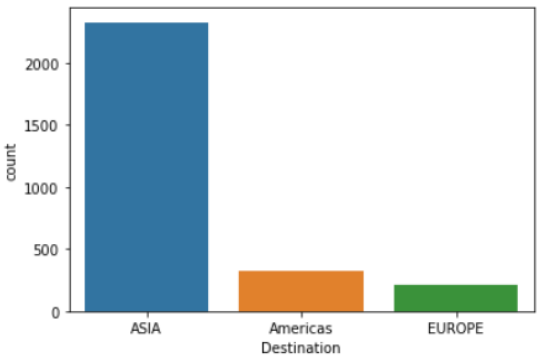


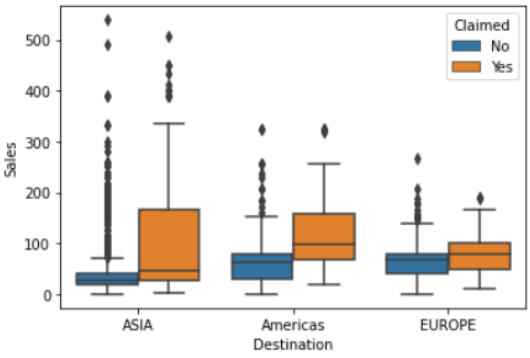

There are two types of agency they are airlines and travel agency people have claimed more in airlines and lower in travel agency



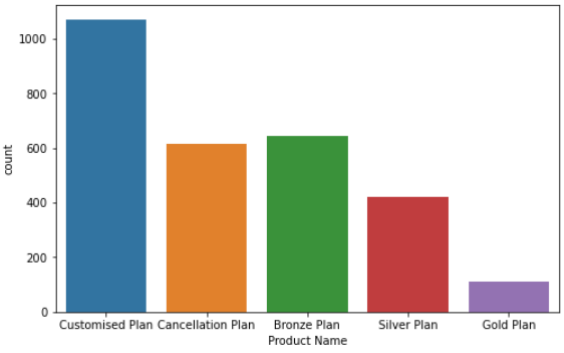



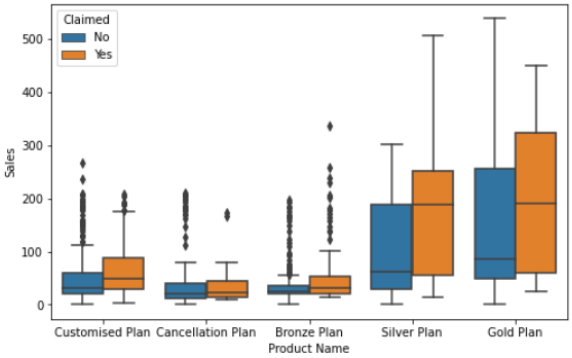

There are two types of channels which people can proceed with online and offline and people have preferred more through online processes




From the above figure There are 3 destination to go with  ASIA AMERICA AND EUROPE and people have travelled more to ASIA and less to EUROPE when compared




From the above figure there are 4 product planes and people have gone foe gold plan the more

## Checking for Correlations and plotting the heat map

| | Age | Commision | Duration | Sales |
|---|---|---|---|---|
| **Age** | 1.000000 | 0.064759 | 0.027457 | 0.036187 |
| **Commision** | 0.064759 | 1.000000 | 0.462114 | 0.762181 |
| **Duration** | 0.027457 | 0.462114 | 1.000000 | 0.549889 |
| **Sales** | 0.036187 | 0.762181 | 0.549889 | 1.000000 |


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

Before doing the data split variables types were checked. As there are many columns with the type as object, these variables were converted into categorical type. Then target variable was captured into separate vector for training and test data set. Then the dataset was split into train and test in the ratio of 70:30.

### Building a Decision Tree Classifier

Decision tree classifier model built using below hyper parameters.

```
'criterion': ['gini'],
'max_depth': [4.85, 4.90,4.95, 5.0,5.05,5.10,5.15],
'min_samples_leaf': [40, 41, 42, 43, 44],
'min_samples_split': [150, 175, 200, 210, 220, 230, 240, 250, 260, 270],
```

Agency Code is the most important variable for predicting claiming insurance.

| | Imp |
|---|---|
| Agency_Code | 0.608787 |
| Sales | 0.206681 |
| Product Name | 0.085441 |
| Duration | 0.059490 |
| Age | 0.028028 |
| Commision | 0.011574 |
| Type | 0.000000 |
| Channel | 0.000000 |
| Destination | 0.000000 |

Prediction of train and test result are stored into ytrain_predict_dtcl and ytest_predict_dtcl respectively.

Head of the predicted class

| | 0 | 1 |
|---|---|---|
| **0** | 0.343066 | 0.656934 |
| **1** | 0.343066 | 0.656934 |
| **2** | 0.852941 | 0.147059 |
| **3** | 0.343066 | 0.656934 |
| **4** | 0.208955 | 0.791045 |

## Building a Random Forest Classifier

Random Forest Classifier classifier model built using below hyper parameters

```
'max_depth': [4,5,6],
'max_features': [2,3,4,5],
'min_samples_leaf': [8,9,11,15],
'min_samples_split': [46,50,55],
'n_estimators': [290,350,400]
```

Agency Code is the most important variable for predicting claiming insurance.

| | Imp |
|---|---|
| Agency_Code | 0.328744 |
| Product Name | 0.247950 |
| Sales | 0.176545 |
| Commision | 0.097763 |
| Duration | 0.063332 |
| Type | 0.042416 |
| Age | 0.036395 |
| Destination | 0.006219 |
| Channel | 0.000635 |

Prediction of train and test result are stored into ytrain_predict_rfdt and ytest_predict_rfdt respectively.

Head of the predicted class

| | 0 | 1 |
|---|---|---|
| 0 | 0.320022 | 0.679978 |
| 1 | 0.409195 | 0.590805 |
| 2 | 0.909398 | 0.090602 |
| 3 | 0.307967 | 0.692033 |
| 4 | 0.192523 | 0.807477 |

## Building a Neural Network Classifier

Neural network classifier model built using below hyper parameters

Scaling is necessary before building a neural network classifier. • After scaling independent train and test data the resulting output is stored on to Scaled_X_train and Scaled_X_test. • Neural-network classifier is built using below hyper-parameters:

```
'hidden_layer_sizes': [50,100,200], # 50, 200
'max_iter': [2500,3000,4000], #5000,2500
'solver': ['adam'], #sgd
'tol': [0.01],
```

Head of the predicted class

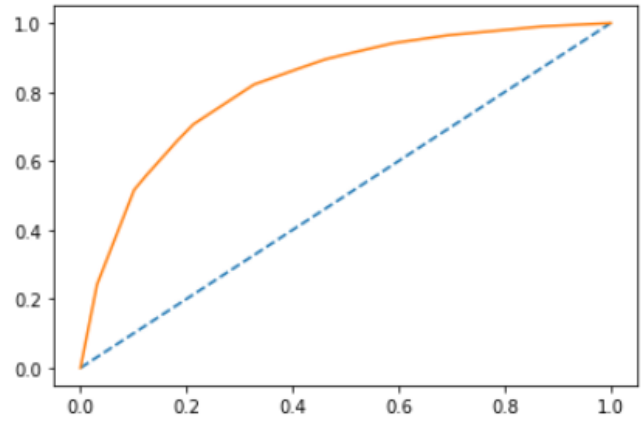
| | 0 | 1 |
|---|---|---|
| 0 | 0.454200 | 0.545800 |
| 1 | 0.352656 | 0.647344 |
| 2 | 0.925234 | 0.074766 |
| 3 | 0.118354 | 0.881646 |
| 4 | 0.341589 | 0.658411 |

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.

CART - AUC and ROC for the training data

```
AUC: 0.820
```

```
[<matplotlib.lines.Line2D at 0x2974fe7ba30>]
```

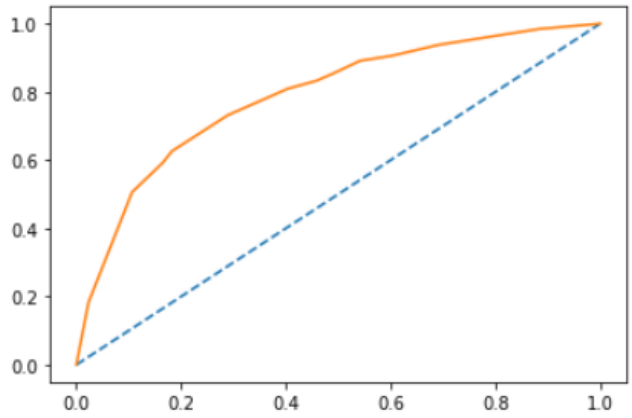



Auc score for training data 0.820

CART - AUC and ROC for the testing data

```
AUC: 0.788
```

```
[<matplotlib.lines.Line2D at 0x2974fecdfa0>]
```




Auc score for testing data 0.788

Classification matrix for testing data:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.79 | 0.90 | 0.84 | 1345 |
| 1 | 0.71 | 0.52 | 0.60 | 657 |
| accuracy | | | 0.77 | 2002 |
| macro avg | 0.75 | 0.71 | 0.72 | 2002 |
| weighted avg | 0.77 | 0.77 | 0.76 | 2002 |

Recall score is 0.52

F1-score is 0.60

Accuracy is 0.77

Confusion matrix

```
array([[1209,  136],
       [ 318,  339]],
```

Classification matrix for training data:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.79 | 0.91 | 0.85 | 602 |
| 1 | 0.68 | 0.43 | 0.53 | 257 |
| accuracy | | | 0.77 | 859 |
| macro avg | 0.73 | 0.67 | 0.69 | 859 |
| weighted avg | 0.76 | 0.77 | 0.75 | 859 |

Recall score is 0.43

F1-score is 0.53

Accuracy is 0.77

Confusion matrix

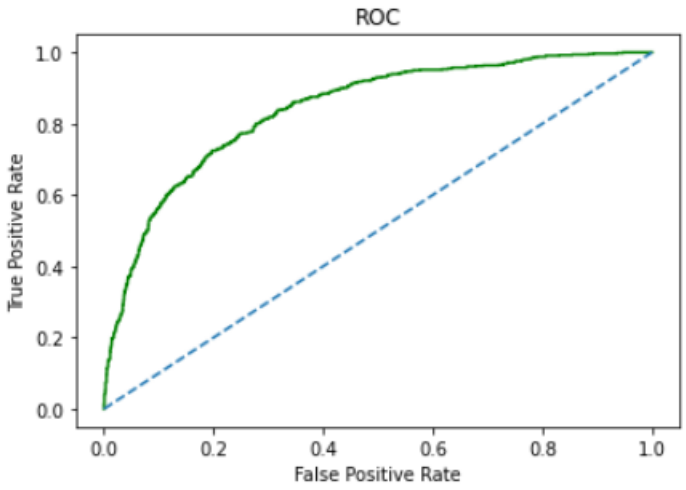
```
array([[550,  52],
       [147, 110]],
```

Training and Test set results are almost similar, and with the overall measures high, the model is a good model.

## Model performance measures for a Random Forest Classifier

AUC and ROC for the training data

```
Area under Curve is 0.8403325921022107
```




Classification matrix for training data:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.83 | 0.87 | 0.85 | 1345 |
| 1 | 0.70 | 0.63 | 0.66 | 657 |
| accuracy | | | 0.79 | 2002 |
| macro avg | 0.76 | 0.75 | 0.75 | 2002 |
| weighted avg | 0.78 | 0.79 | 0.79 | 2002 |

Recall score is 0.63

F1-score is 0.66

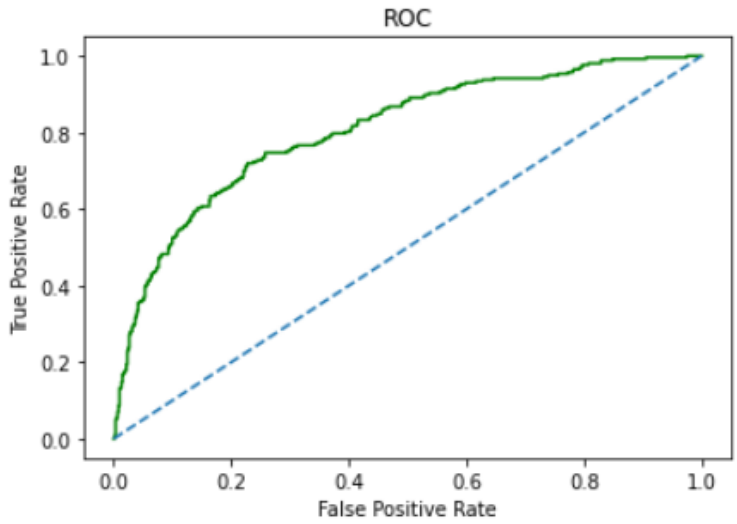
Accuracy is 0.79

confusion matrix for testing data

```
array([[550,  52],
       [147, 110]],
```

**AUC and ROC for the testing data**

```
Area under Curve is 0.8063717569192187
```




**Classification matrix for testing data**

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.82 | 0.88 | 0.85 | 602 |
| 1 | 0.67 | 0.55 | 0.60 | 257 |
| accuracy | | | 0.78 | 859 |
| macro avg | 0.74 | 0.72 | 0.73 | 859 |
| weighted avg | 0.77 | 0.78 | 0.78 | 859 |

Recall score is 0.55

F1-score is 0.60

Accuracy is 0.78

Confusion matrix for testing data:

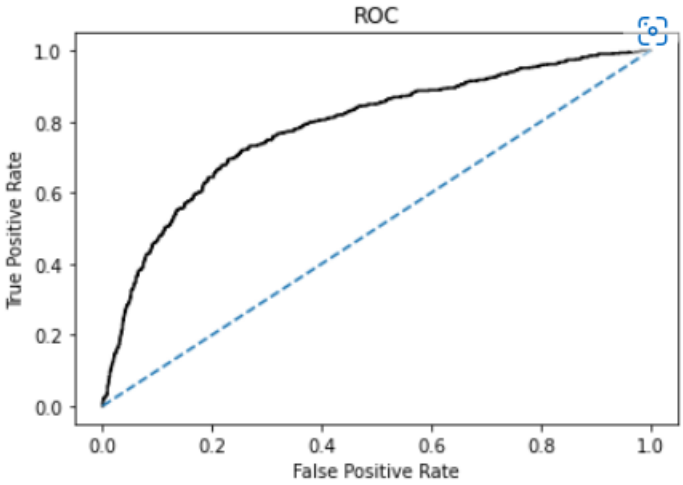
```
array([[531,  71],
       [116, 141]],
```

Training and Test set results are almost similar, and with the overall measures high, the model is a good model.

## Model performance measures for a neural network classifier

## ROC-AUC curve for testing data

```
Area under Curve is 0.7834688484889636
```




## Classification matrix for testing data

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.76 | 0.92 | 0.84 | 1345 |
| 1 | 0.73 | 0.41 | 0.52 | 657 |
| accuracy | | | 0.76 | 2002 |
| macro avg | 0.74 | 0.67 | 0.68 | 2002 |
| weighted avg | 0.75 | 0.76 | 0.73 | 2002 |

Recall score is 0.41

F1-score is 0.52

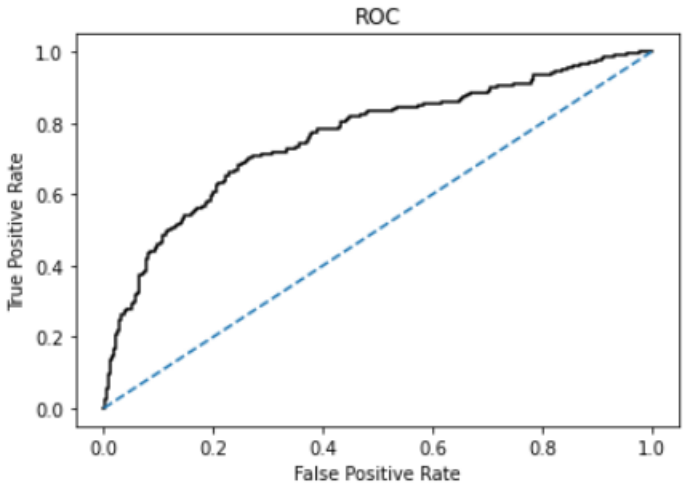
Accuracy is 0.76

## Confusion matrix for testing data

```
array([[1243,  102],
       [ 388,  269]], dtype=int64)
```

## ROC-AUC curve for training data

```
Area under Curve is 0.7613467430226095
```




Classification matric for training data

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.78 | 0.93 | 0.85 | 602 |
| 1 | 0.69 | 0.37 | 0.48 | 257 |
| accuracy | | | 0.76 | 859 |
| macro avg | 0.73 | 0.65 | 0.67 | 859 |
| weighted avg | 0.75 | 0.76 | 0.74 | 859 |

Recall score is 0.37

F1-score is 0.48

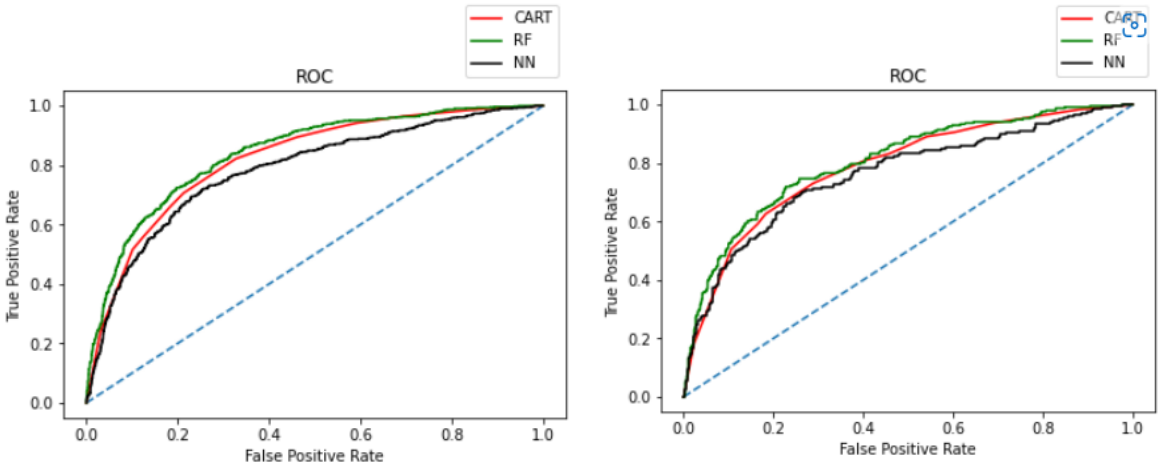
Accuracy is 0.76

Confusion matrix for training data

```
array([[559,  43],
       [161,  96]],
```

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

**ROC Curve for the 3 models on the Training data and testing data**




• Out of the 3 models, Random Forest has slightly better performance than the Cart and Neural network 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 person has claimed or not. If claimed is yes, then those customers have more chances of getting tour insurance.

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

This is understood by looking at the insurance data by drawing relations between different variables such as day of the incident, time, age group, and associating it with other external information such as location, behaviour patterns, weather information, airline/vehicle types, etc.

• Streamlining online experiences benefitted customers, leading to an increase in conversions, which subsequently raised profits.

• As per the data 90% of insurance is done by online channel.

• Other interesting fact, is almost all the offline business has a claimed associated, need to find why?
• Need to train the JZI agency resources to pick up sales as they are in bottom, need to run promotional marketing campaign or evaluate if we need to tie up with alternate agency

• Also based on the model we are getting 80%accuracy, so we need customer books airline tickets or plans, cross sell the insurance based on the claim data pattern.

• Other interesting fact is more sales happen via Agency than Airlines and the trend shows the claim are processed more at Airline. So, we may need to deep dive into the process to understand the workflow and why?

Key performance indicators (KPI) The KPI's of insurance claims are:

• Reduce claims cycle time

• Increase customer satisfaction

• Combat fraud

• Optimize claims recovery

• Reduce claim handling costs Insights gained from data and AI-powered analytics could expand the boundaries of insurability, extend existing products, and give rise to new risk transfer solutions in areas like a non-damage business interruption and reputational damage.