PROJECT REPORT - DATA MINING

By prakash v Mahadole

Table of Contents

Problem 1: Clustering

Problem Statement:

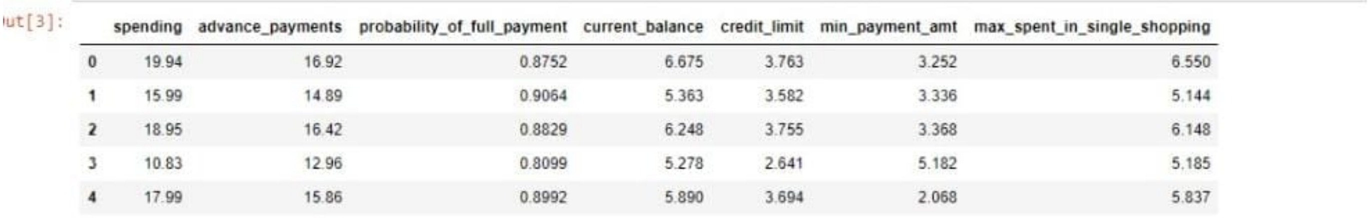
A leading bank wants to develop a customer segmentation to give promotional offers to its customers. They collected a sample that summarizes the activities of users during the past few months. You are given the task to identify the segments based on credit card usage.

Data Dictionary for Market Segmentation:

- spending: Amount spent by the customer per month (in 1000s)
- advance_payments: Amount paid by the customer in advance by cash (in 100s)
- probability_of_full_payment: Probability of payment done in full by the customer to the bank
- current_balance: Balance amount left in the account to make purchases (in 1000s)
- credit_limit: Limit of the amount in credit card (10000s)
- min_payment_amt : minimum paid by the customer while making payments for purchases made monthly (in 100s)
- max_spent_in_single_shopping: Maximum amount spent in one purchase (in 1000s) ### Q 1.1 Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis) import all the necessary libraries

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

Checking the head of the data

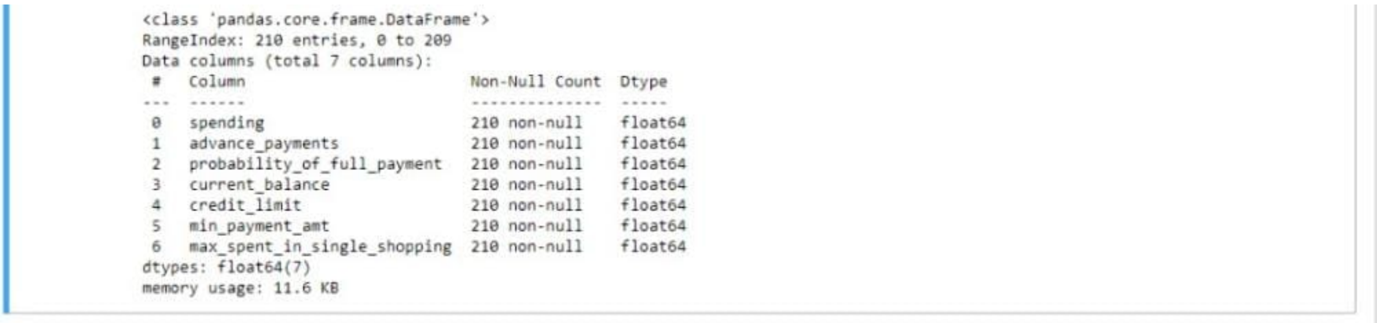


ut[3]:

	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

In this Data set we have 7 variables and 210 data’s as a whole. All the data is the credit card spending of the customers in recent months.

checking data info

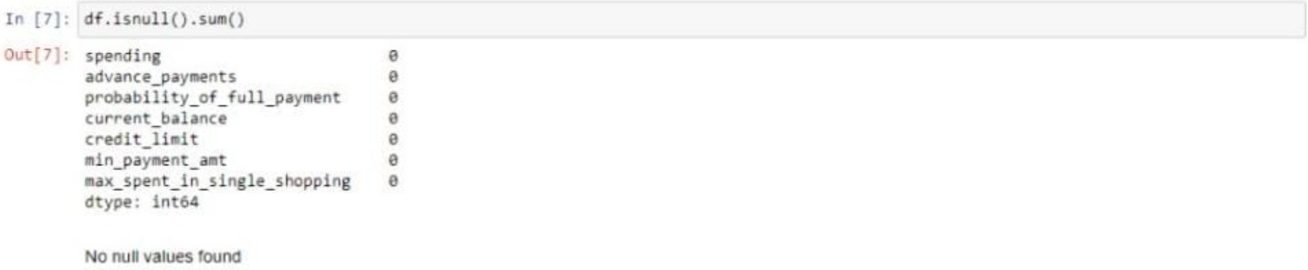


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   spending                      210 non-null    float64
 1   advance_payments              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
```

Observation:

- 7 variables and 210 records.
- No missing record based on intial analysis.
- All the variables numeric type.

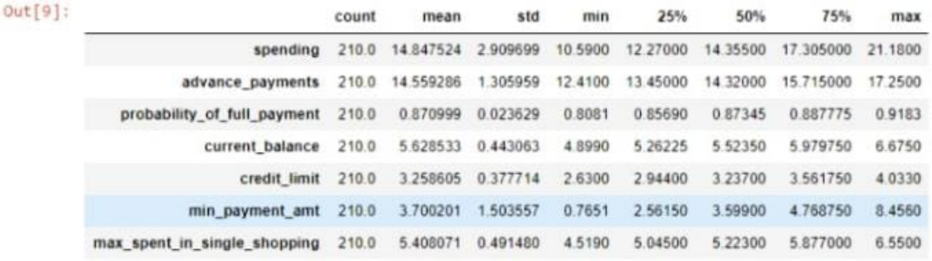
Check for null values



```
In [7]: df.isnull().sum()

Out[7]: 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

        No null values found
```

No null values as well as no duplicates in the data set.

Univariate analysis



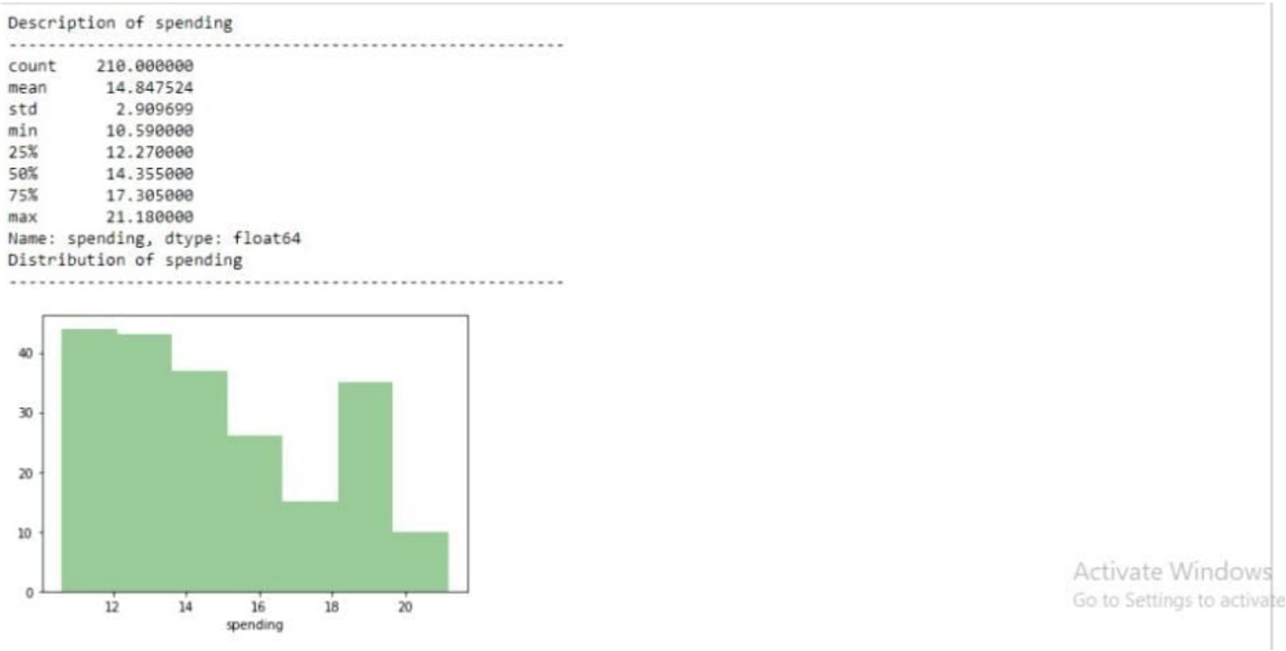
Out[9]:

	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

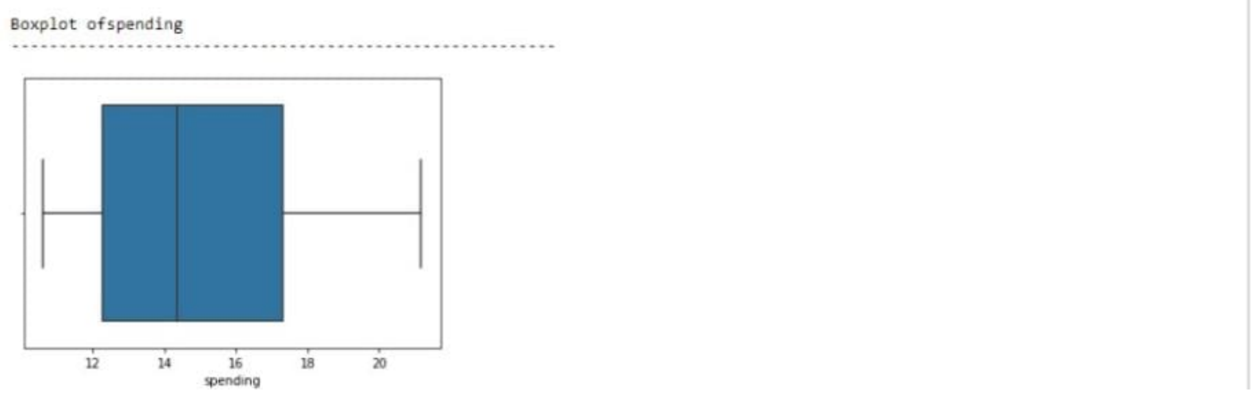
Observation

- Based on summary descriptive, the data looks good.
- We see for most of the variable, mean/medium are nearly equal
- Std Deviation is high for spending variable

description and distribution and boxplot of each variable:



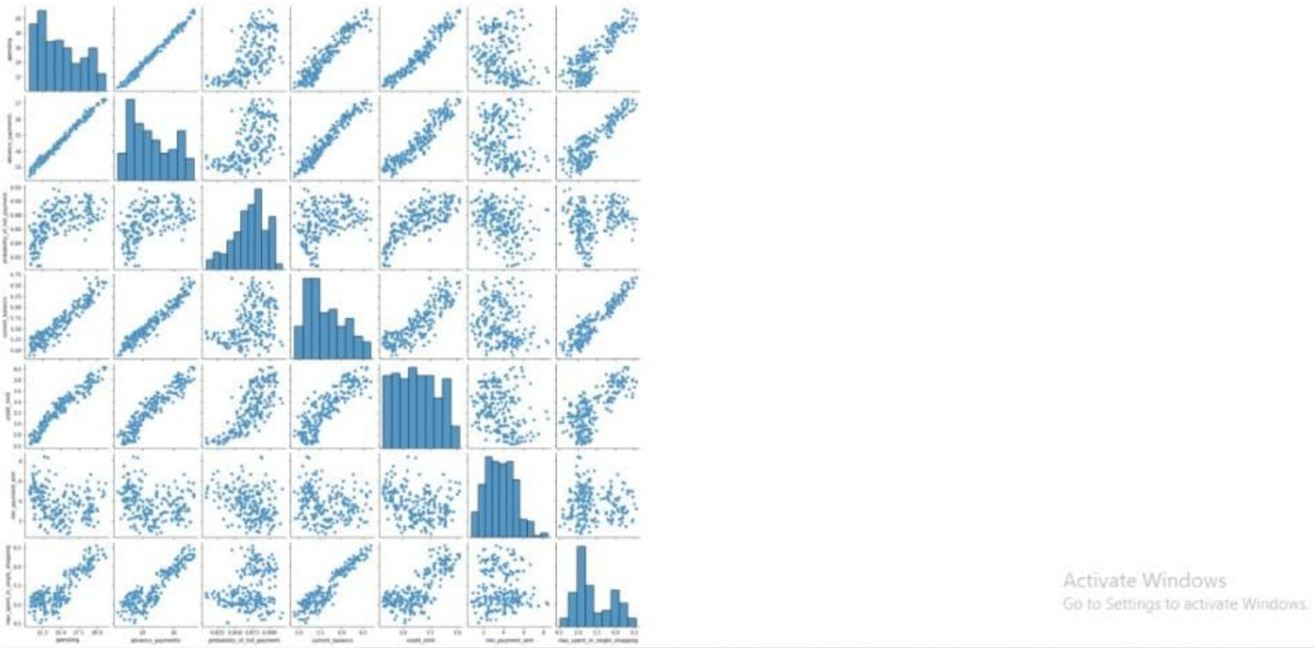
```
Description of spending
------------------------------------------------------------
count    210.000000
mean      14.847524
std        2.909699
min       10.590000
25%       12.270000
50%       14.355000
75%       17.305000
max       21.180000
Name: spending, dtype: float64
Distribution of spending
------------------------------------------------------------
```



```
Boxplot ofspending
------------------------------------------------------------
```

The output displays, total 7*3 = 21 distinct charts/columns. Hence I have put the screenshot of only one variable i.e. spending.

Pairplot



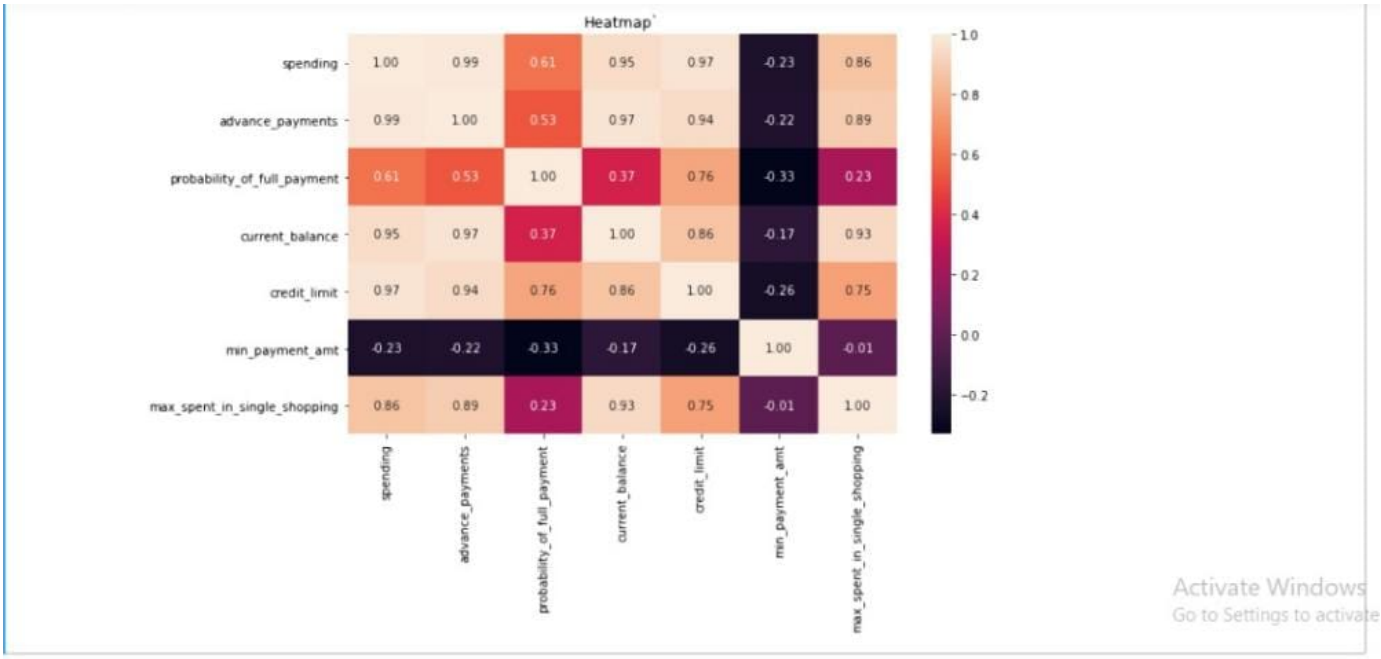
- Pair plot shows the relationship between the variables in the form of scatterplot and the distribution of the variable in the form of histogram.
- From the graph, we can see that there is strong positive linear relationship between most of the variables like advance_payment and spending, current_balance and advance_payment . From the histogram we can see that almost all of the variables are right skewed but only probability_of_ful_payment is left skewed.

Observation:

- Strong positive correlation between

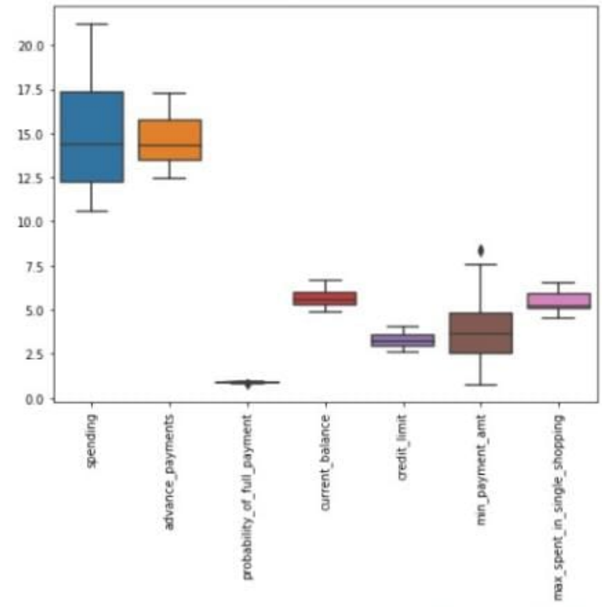
```
    - spending & advance_payments,
    - advance_payments & current_balance,
    - credit_limit & spending
    - spending & current_balance
    - credit_limit & advance_payments
    - max_spent_in_single_shopping  current_balance
```

Correlation plot



- From the correlation plot, we can see that various attributes of the car are highly correlated to each other. Correlation values near to 1 or -1 are highly positively correlated and highly negatively correlated respectively. Correlation values near to 0 are not correlated to each other.
- From the correlation plot we can see that min_payment_amt is highly negatively correlated to most of the variables.
- Then values near to 0 have no correlation and can be found in min_payment_amt and max_spent_in_single_shopping.
- We can find high positive correlation between advance_payments and spending, advance_payments and current_balance, credit_limit and spending from this we can say that people make advance payments before their spending's and spend within their credit limit

Plotting boxplot and checking outliers



observation:

seems there are some outliers.

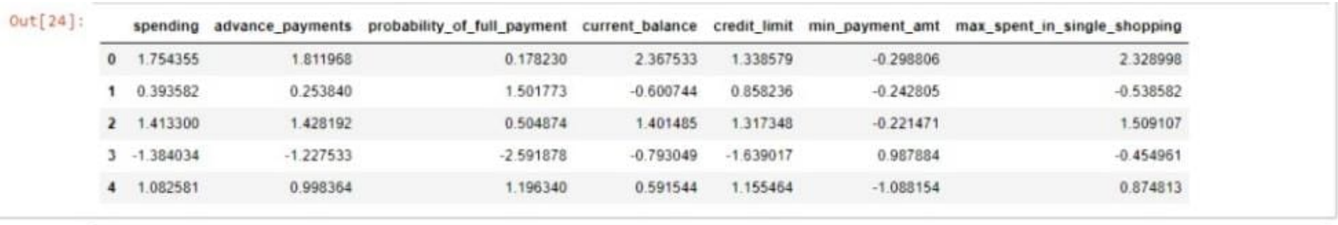
Replacing the outliers with median values

we still see one as per the boxplot, it is okay, as it is no extrme and on lower band.

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

- Yes, scaling is necessary for the data, because the variables 'advance_payments' and 'spending' have their values in higher (in ten’s) compared to others (one’s) so they dominate more.
- Scaling will have all the values in the relative same range.
- we will be using zscore to standarised the data to relative same scale -3 to +3.
- Some values are in thousands, some in hundreds, so the values are imbalanced. We can't process further without scaling.
- More over Clustering algorithms such as K-means need feature scaling. Since, clustering techniques use Euclidean Distance to form the clusters.

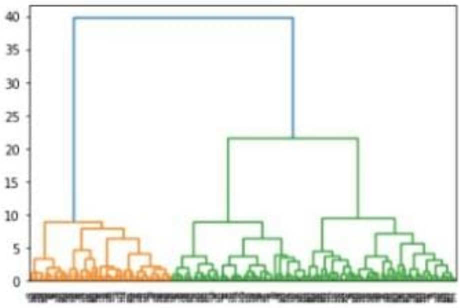
using z-score technique we will get the following scaled output


Out[24]:

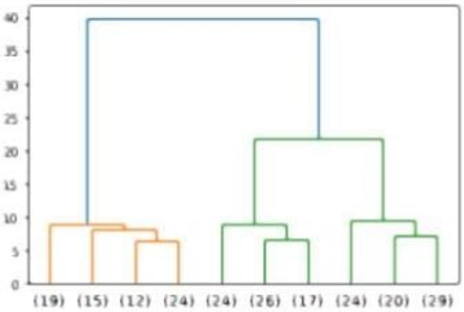
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754355	1.811968	0.178230	2.367533	1.338579	-0.298806	2.328998
1	0.393582	0.253840	1.501773	-0.600744	0.858236	-0.242805	-0.538582
2	1.413300	1.428192	0.504874	1.401485	1.317348	-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

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

Creating the Dendrogram



Cutting Dendrogram with suitable clusters



Clustering techniques

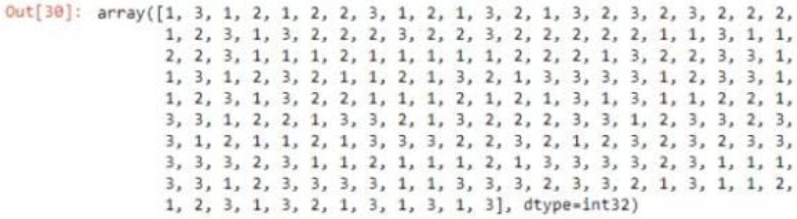
The method used for clustering is fcluster. Here we are using 3 criteria’s for forming flat clusters

- The linkage formed using the scaled data.
- The cut off value
- The criterion has 2 methods
  - Distance
  - maxcluster

Here we have used both the methods, we can choose any one of them, both gives almost the same results.

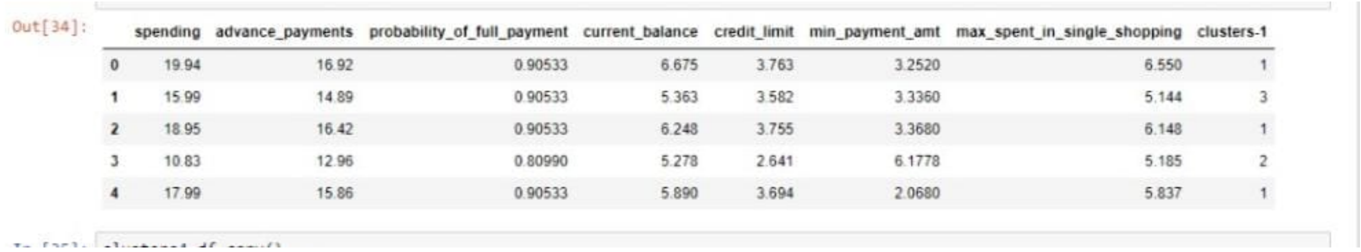
- The linkage used is ward link method.

we get the following output using wardlink



```
Out[30]: array([1, 3, 1, 2, 1, 2, 2, 3, 1, 2, 1, 3, 2, 1, 3, 2, 3, 2, 3, 2, 2, 2,
                1, 2, 3, 1, 3, 2, 2, 2, 3, 2, 2, 3, 2, 2, 2, 2, 2, 1, 1, 3, 1, 1,
                2, 2, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 3, 2, 2, 3, 3, 1,
                1, 3, 1, 2, 3, 2, 1, 1, 2, 1, 3, 2, 1, 3, 3, 3, 3, 1, 2, 3, 3, 1,
                1, 2, 3, 1, 3, 2, 2, 1, 1, 1, 2, 1, 2, 1, 3, 1, 3, 1, 1, 2, 2, 1,
                3, 3, 1, 2, 2, 1, 3, 3, 2, 1, 3, 2, 2, 2, 3, 3, 1, 2, 3, 3, 2, 3,
                3, 1, 2, 1, 1, 2, 1, 3, 3, 3, 2, 2, 3, 2, 1, 2, 3, 2, 3, 2, 3, 3,
                3, 3, 3, 2, 3, 1, 1, 2, 1, 1, 1, 2, 1, 3, 3, 3, 3, 2, 3, 1, 1, 1,
                3, 3, 1, 2, 3, 3, 3, 3, 1, 1, 3, 3, 3, 2, 3, 3, 2, 1, 3, 1, 1, 2,
                1, 2, 3, 1, 3, 2, 1, 3, 1, 3, 1, 3], dtype=int32)
```

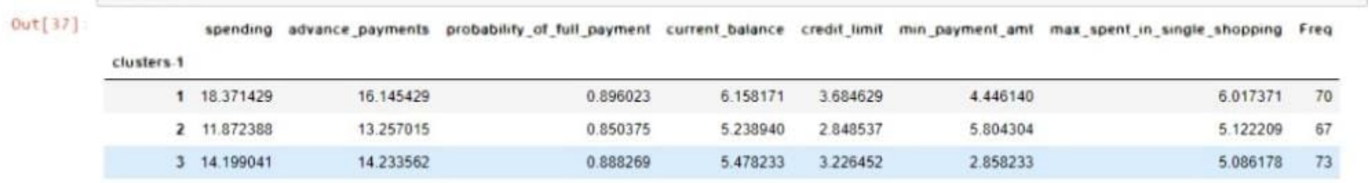
After clustering the data, we have appended the cluster to the original data frame



Out[34]:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	clusters-1
0	19.94	16.92	0.90533	6.675	3.763	3.2520	6.550	1
1	15.99	14.89	0.90533	5.363	3.582	3.3360	5.144	3
2	18.95	16.42	0.90533	6.248	3.755	3.3680	6.148	1
3	10.83	12.96	0.80990	5.278	2.641	6.1778	5.185	2
4	17.99	15.86	0.90533	5.890	3.694	2.0680	5.837	1

Grouping of entire data after clustering, to know the overall cluster formation



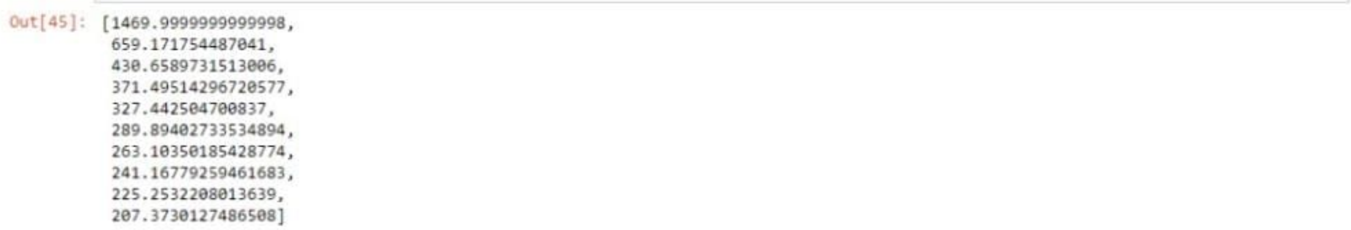
Out[37]

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
clusters-1								
1	18.371429	16.145429	0.896023	6.158171	3.684629	4.446140	6.017371	70
2	11.872388	13.257015	0.850375	5.238940	2.848537	5.804304	5.122209	67
3	14.199041	14.233562	0.888269	5.478233	3.226452	2.858233	5.086178	73

- The clusters are segregated based on spending variable.
- We for cluster grouping based on the dendrogram 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 colud have been more variables value captured - tenure, BALANCE_FREQUENCY, balance, purchase, installment 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).

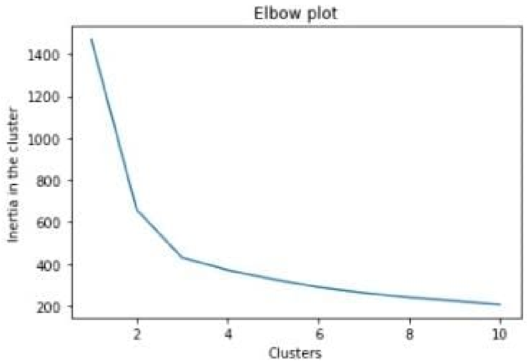
Question 1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Explain the results properly. Interpret and write inferences on the finalized clusters.

We have calculated the inertia value for the data from values ranging 1 to 11



```
Out[45]: [1469.9999999999998,
          659.171754487041,
          430.6589731513006,
          371.49514296720577,
          327.442504700837,
          289.89402733534894,
          263.10350185428774,
          241.16779259461683,
          225.2532208013639,
          207.3730127486508]
```

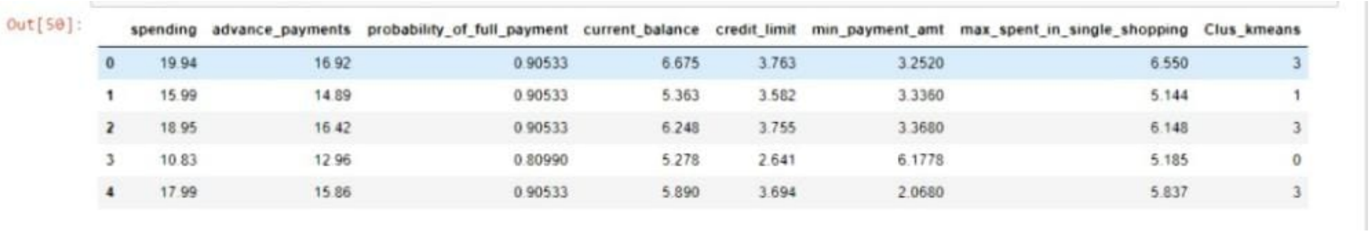
Elbow plot for the data is:



Observation:

From the above plot (Elbow curve) we can see that there is a significant drop from 2 to 3 and after 3 the inertia start decreasing in a linear fashion so we can choose (n-cluster) 3 as optimum number of clusters.

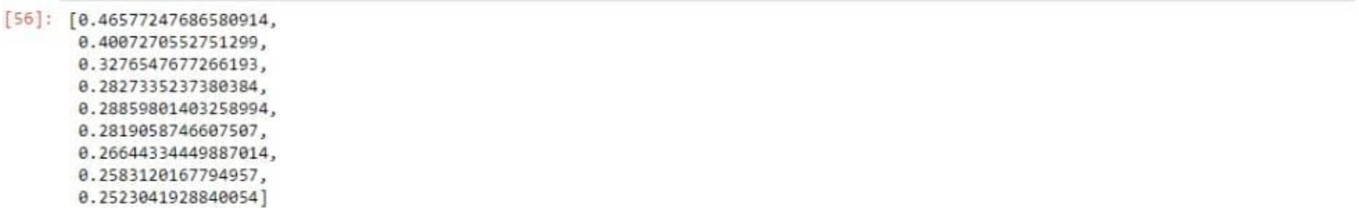
After clustering the data becomes



Out[50]:

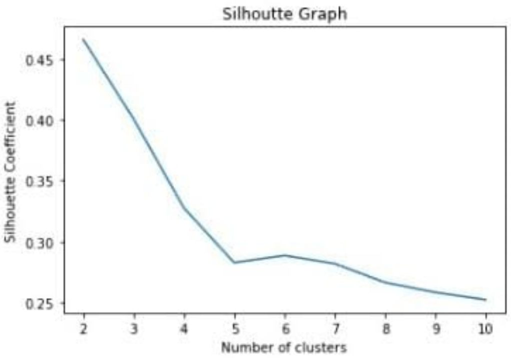
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Clus_kmeans
0	19.94	16.92	0.90533	6.675	3.763	3.2520	6.550	3
1	15.99	14.89	0.90533	5.363	3.582	3.3360	5.144	1
2	18.95	16.42	0.90533	6.248	3.755	3.3680	6.148	3
3	10.83	12.96	0.80990	5.278	2.641	6.1778	5.185	0
4	17.99	15.86	0.90533	5.890	3.694	2.0680	5.837	3

Calculating Silhouette_score for the data,



```
[56]: [0.46577247686580914,
       0.4007270552751299,
       0.3276547677266193,
       0.2827335237380384,
       0.28859801403258994,
       0.2819058746607507,
       0.26644334449887014,
       0.2583120167794957,
       0.2523041928840054]
```

silhoutte graph



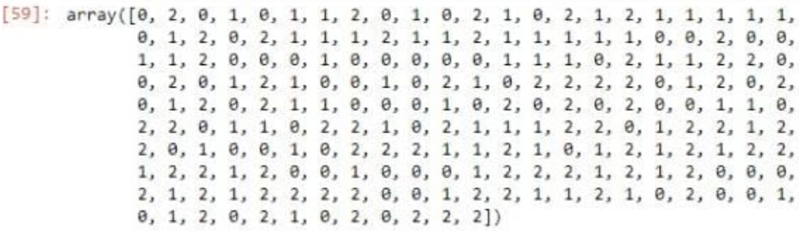
Observation:

- From the above silhouette graph we can see that n_cluster = 3 or 4 has the highest value compared to others.

As we can say that :

- If the silhouette score is close to +1 then we can say the clusters are well separated from each other
- If the silhouette score is close to 0, then we can say the clusters are not separated from each other.
- So from the above statement we can say that the best cluster to choose is 3 to be optimum

After fitting the kmeans,we get



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

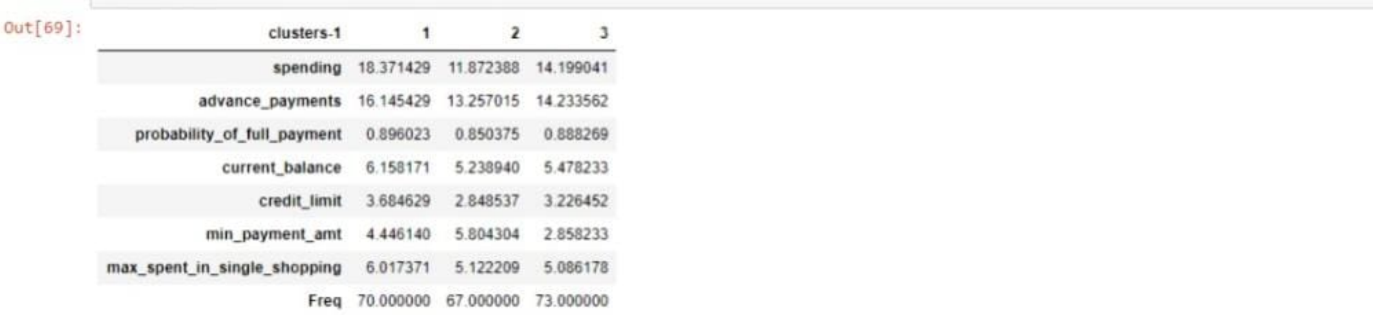
Cluster 1 contains customers who spends average, so they majority of middle class people fall under this cluster. 1) The middle class people mostly will be married, so they'll be in need of money, giving low interest personal loans / for vehicles can draw most people. As their next plan will be buying a car / house / paying fees. So they see it as a good source comparatively to other loans offers.

2) Giving offers to most used e-commerce sites during festival times drives more customers to spend on household items.

Cluster 2 contains customers who spends more, so in order to maintain them or to make them spend more we have to segregate them based on gender, 1) For female customers, they tend to spend more on cosmetics, health and beauty products. Go giving offers on specific brands, which is bit costlier will drive them to spend more to maintain their status.

2) This suits for both the genders, to update their gadgets, as it has become a status mark to use certain products, so giving offers on such products during the product release will drive more customers by giving promotions.

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



Out[69]:

clusters-1	1	2	3
spending	18.371429	11.872388	14.199041
advance_payments	16.145429	13.257015	14.233562
probability_of_full_payment	0.896023	0.850375	0.888269
current_balance	6.158171	5.238940	5.478233
credit_limit	3.684629	2.848537	3.226452
min_payment_amt	4.446140	5.804304	2.858233
max_spent_in_single_shopping	6.017371	5.122209	5.086178
Freq	70.000000	67.000000	73.000000

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 there credit limit and
- Increase spending habits
- Give loan against the credit card, as they are customers with good repayment record.
- Tie up with luxary 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 comparatively good credit score. So we can increase credit limit or can lower down interest rate.
- Promote premium cards/loyality cars to increase transcations.
- Increase spending habits by trying with premium ecommerce sites, travel portal, travel airlines/hotel, as this will encourge them to spend more

Group 2 : Low Spending Group

- customers should be given remainders for payments. Offers can be provided on early payments to improve their payment rate.
- Increase there spending habits by tieing up with grocery stores, utlities (electircity, phone, gas, others)

Problem 2: CART-RF-ANN

Problem Statement:

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.

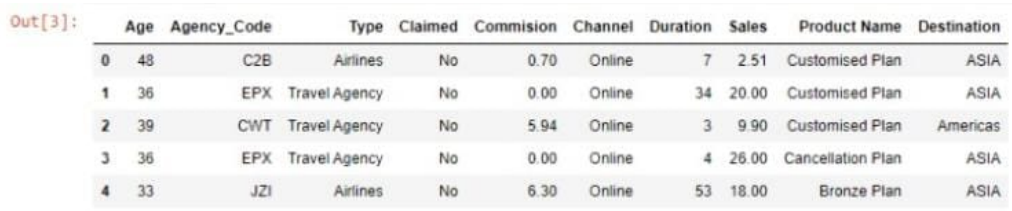
Attribute Information:

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)
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)

Import all the necesary libraries

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

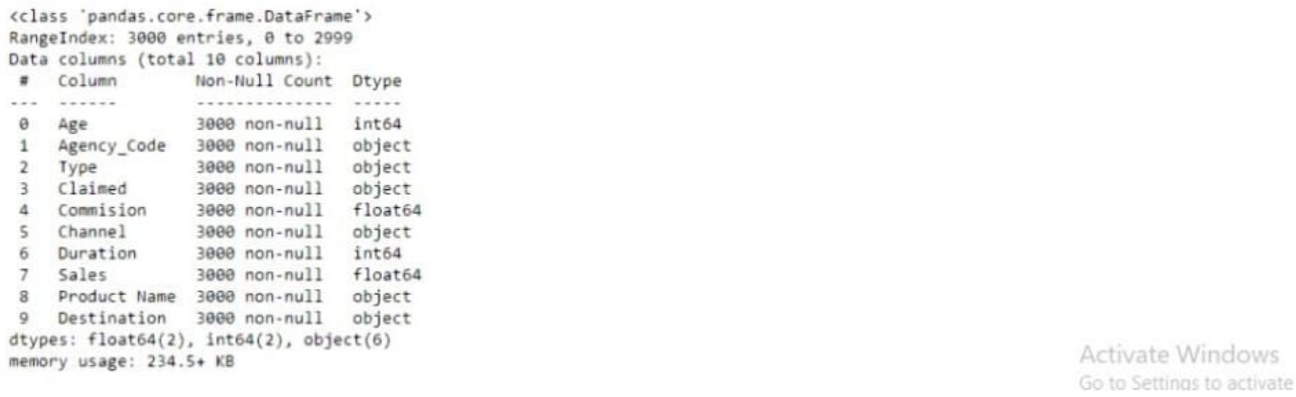
load the data in data frame and check for head



Out[3]:

	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

Checking data info



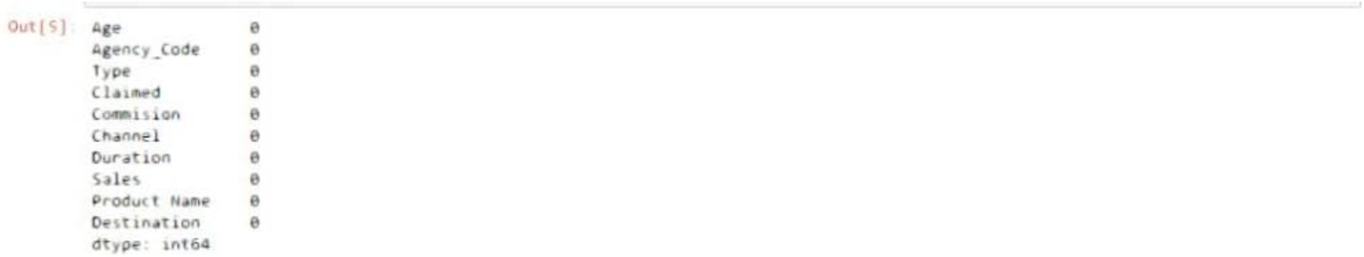
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Age           3000 non-null   int64  
 1   Agency_Code   3000 non-null   object 
 2   Type          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
```

Observatiom

- 10 variables
- Age, Commision, Duration, Sales are numeric variable
- rest are categorial variables
- 3000 records, no missing one
- 9 independant variable and one target variable - Clamied

EDA

checking for missing values

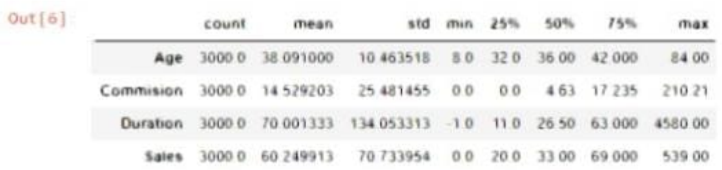


```
Out[5]: Age             0
        Agency_Code     0
        Type            0
        Claimed         0
        Commision       0
        Channel         0
        Duration        0
        Sales           0
        Product Name    0
        Destination     0
        dtype: int64
```

Observation

No missing values found.

Descriptive Statistics Summary



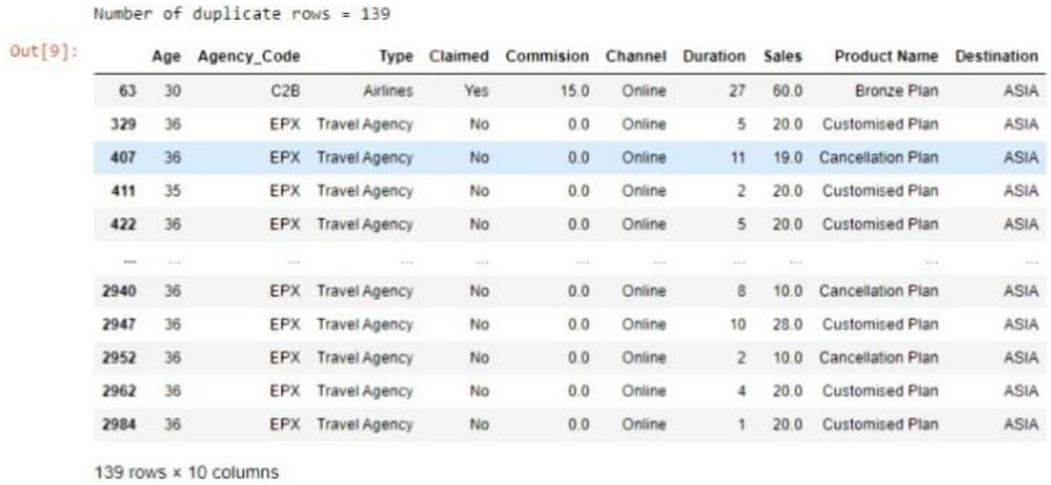
Out[6]

	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

Observation

- duration has negative value, it is not possible. Wrong entry.
- Commision & Sales- mean and median varies signficantly

Duplicate Data Detection



```
Number of duplicate rows = 139
```

Out[9]:

	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

observation

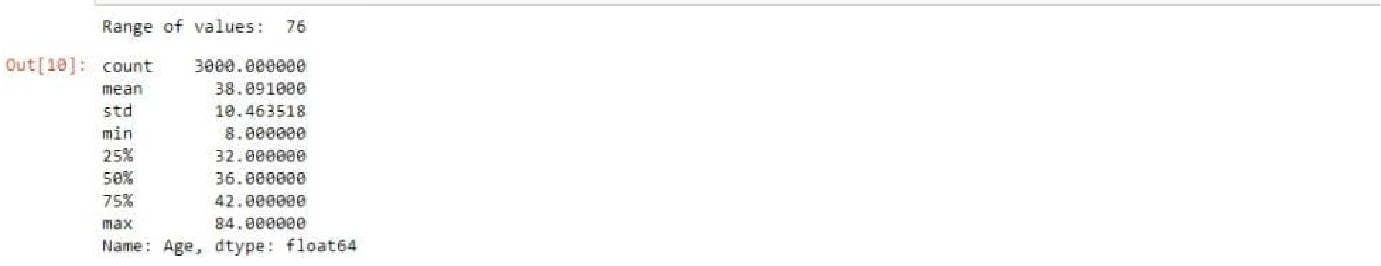
Not removing duplicates

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

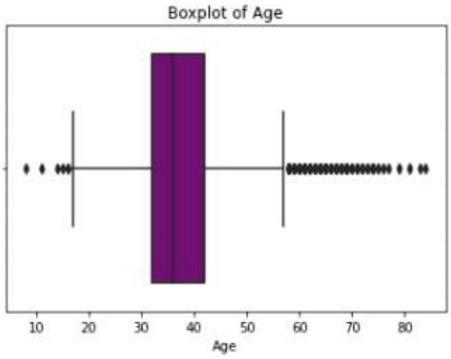
Univariate analysis

We will be doing univariate analysis of Numairc variables first, then move to catagorical variables.

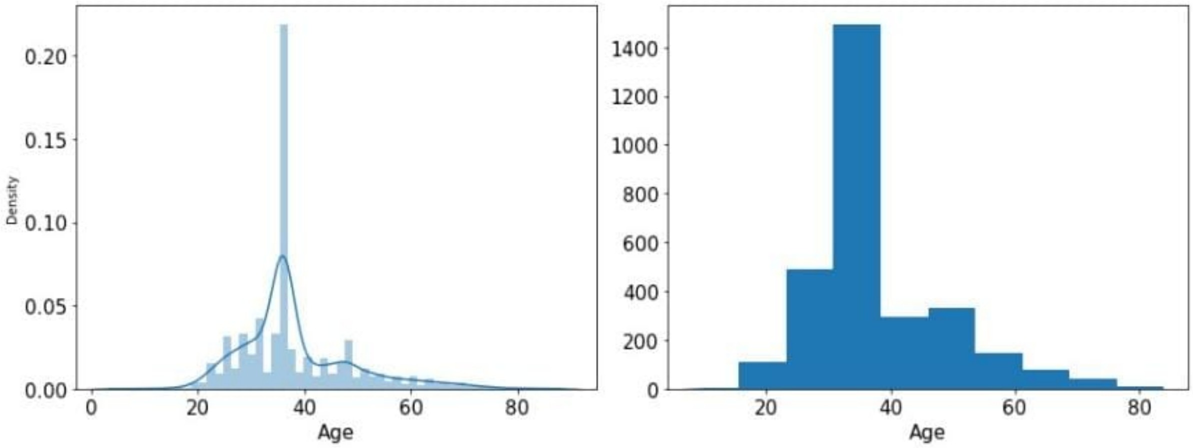
For Age Variable



```
         Range of values:  76

Out[10]: 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
```



Distplot And Histogram for Age variable



Do the same for commision, Duration and for Sales variable.

Observation:

There are outliers in all the variables, but the sales and commision can be a geneui business value. Random Forest and CART can handle the outliers. Hence, Outliers are not treated for now, we will keep the data as it is.

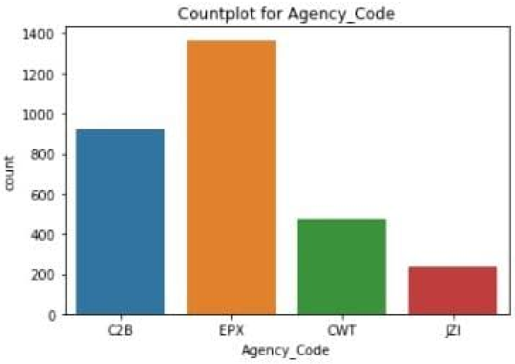
I will treat the outliers for the ANN model to compare the same after the all the steps just for comparsion.

Categorical Variable

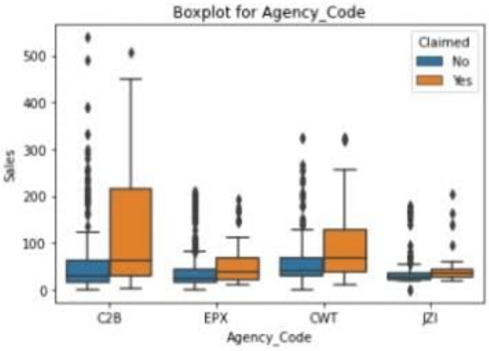
we will be plotting countplot and boxplot for all the categorical variables

Agency_Code

Countplot for Agency_Code Variable

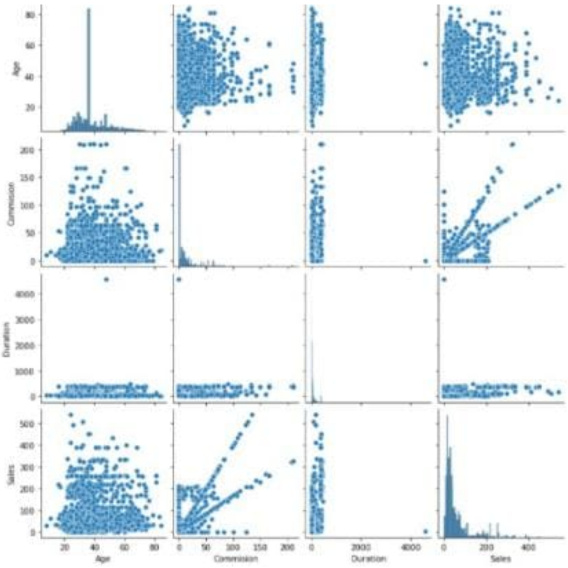


Boxplot for Agency_Code variable

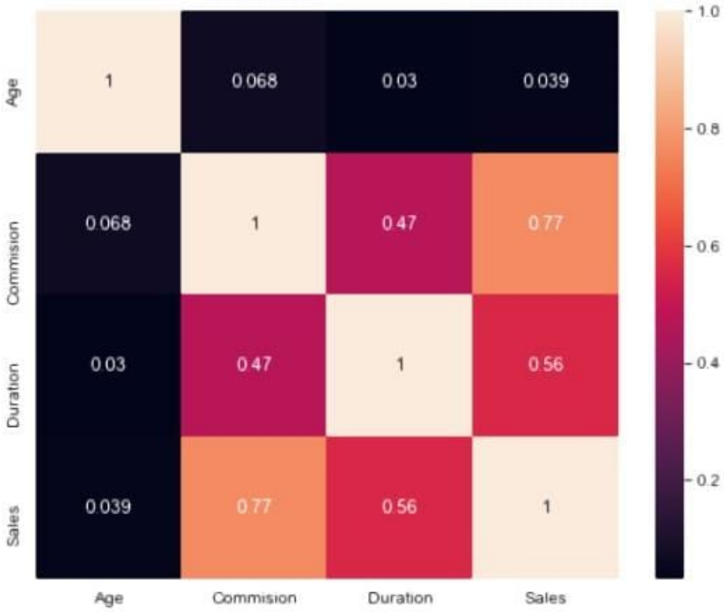


Do the Same for other categorical variables-

Checking pairwise distribution of the continuous variables



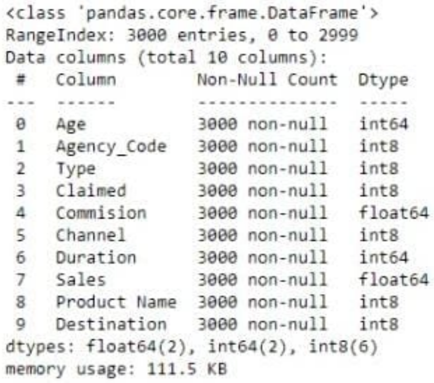
Correlation plot



From the heat map we can see that,

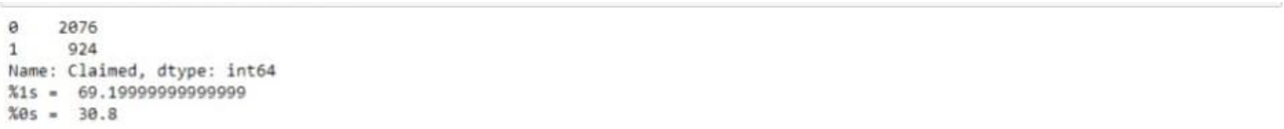
- There is no much linear correlation between Age and other variables
- There is only few positive correlation with Commission and sales, Duration and Sales

Converting all the categorical variables into numaric and checking info



```
<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   int8
 2   Type          3000 non-null   int8
 3   Claimed       3000 non-null   int8
 4   Commision     3000 non-null   float64
 5   Channel       3000 non-null   int8
 6   Duration      3000 non-null   int64
 7   Sales         3000 non-null   float64
 8   Product Name  3000 non-null   int8
 9   Destination   3000 non-null   int8
dtypes: float64(2), int64(2), int8(6)
memory usage: 111.5 KB
```

Proportion of 1s and 0s



```
0    2076
1     924
Name: Claimed, dtype: int64
%1s =  69.19999999999999
%0s =  30.8
```

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

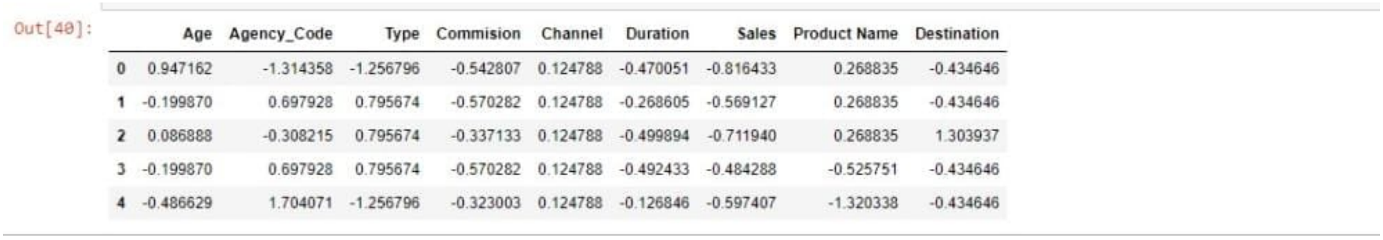
Extracting the target column into separate vectors for training set and test set .

we will drop the Dependent variable = ‘Claimed’ and separate them using train_test_split. So extracting the target variable.

Then we will scale the data for ANN, but not necessary for Random Forest and CART model because both of these are trees and are not distance based algorithm, but ANN is distance based, so scaled the data for ANN.

we have already converted the object data type into numerical data type as the algorithms will only accept numerical data.

Scaled data set head



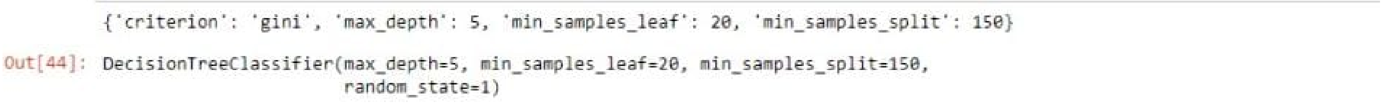
Out[40]:

	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

Building a Decision Tree Classifier

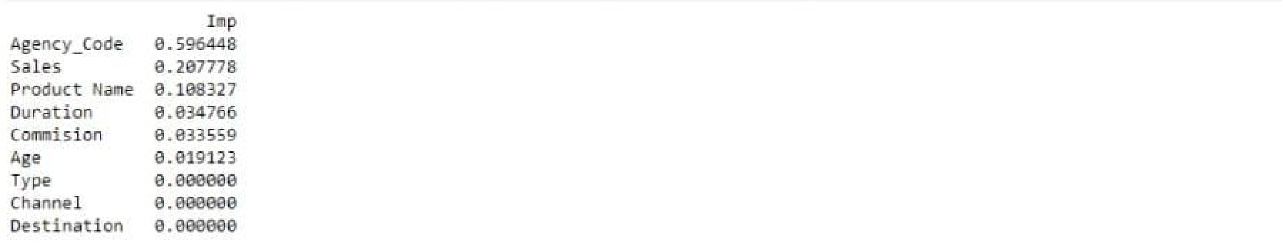
- The criterion gini is what based on which the variables are chosen for the split.
- Max_depth is the maximum levels of trees can extend for best results
- Min_sample_leaf is the value of the last child note to which it extends.
- Min_sample_split is the value that the leaf node should contain during the split.

In this we have the following parameters



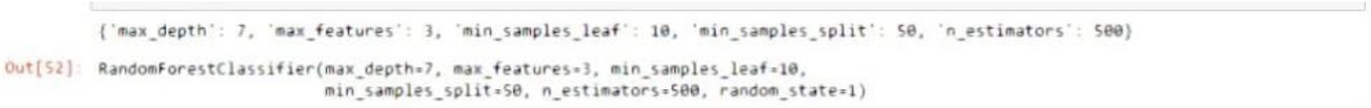
```
        {'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 20, 'min_samples_split': 150}

Out[44]: DecisionTreeClassifier(max_depth=5, min_samples_leaf=20, min_samples_split=150,
                                random_state=1)
```

Variable importance



```
                  Imp
Agency_Code   0.596448
Sales         0.207778
Product Name  0.108327
Duration      0.034766
Commision     0.033559
Age           0.019123
Type          0.000000
Channel       0.000000
Destination   0.000000
```

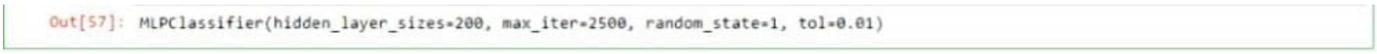
Random tree Classifier



```
        {'max_depth': 7, 'max_features': 3, 'min_samples_leaf': 10, 'min_samples_split': 50, 'n_estimators': 500}

Out[52]: RandomForestClassifier(max_depth=7, max_features=3, min_samples_leaf=10,
                                min_samples_split=50, n_estimators=500, random_state=1)
```

MLPClassifier

In this we have the following parameters

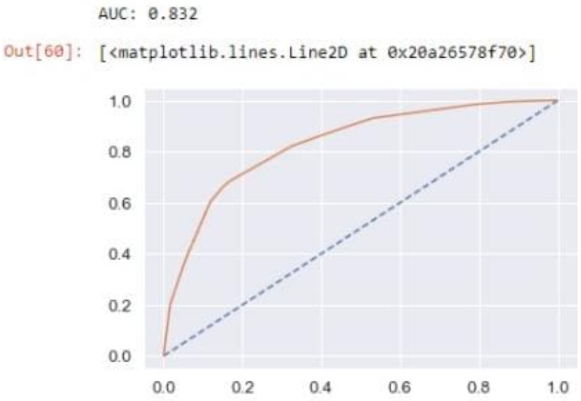


```
Out[57]: MLPClassifier(hidden_layer_sizes=200, max_iter=2500, random_state=1, tol=0.01)
```

All the above parameters are chosen at random based on the parameter grid search method. Based on these parameters we will build the models by fitting the data into the algorithm.

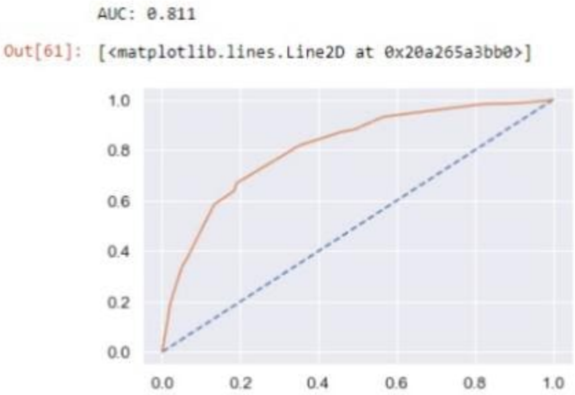
Question 2.3 Performance Metrics: Comment and Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score, classification reports for each model.

CART - AUC and ROC for the training data



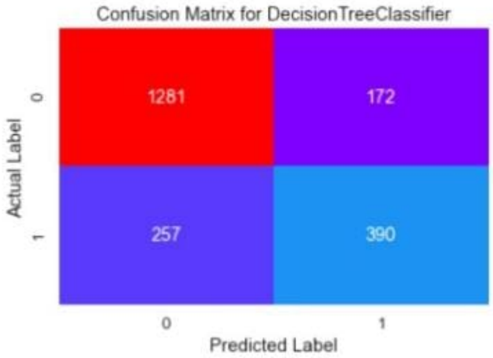
```
AUC: 0.832
Out[60]: [<matplotlib.lines.Line2D at 0x20a26578f70>]
```

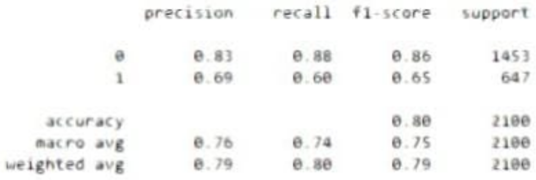
CART - AUC and ROC for the test data



```
AUC: 0.811
Out[61]: [<matplotlib.lines.Line2D at 0x20a265a3bb0>]
```

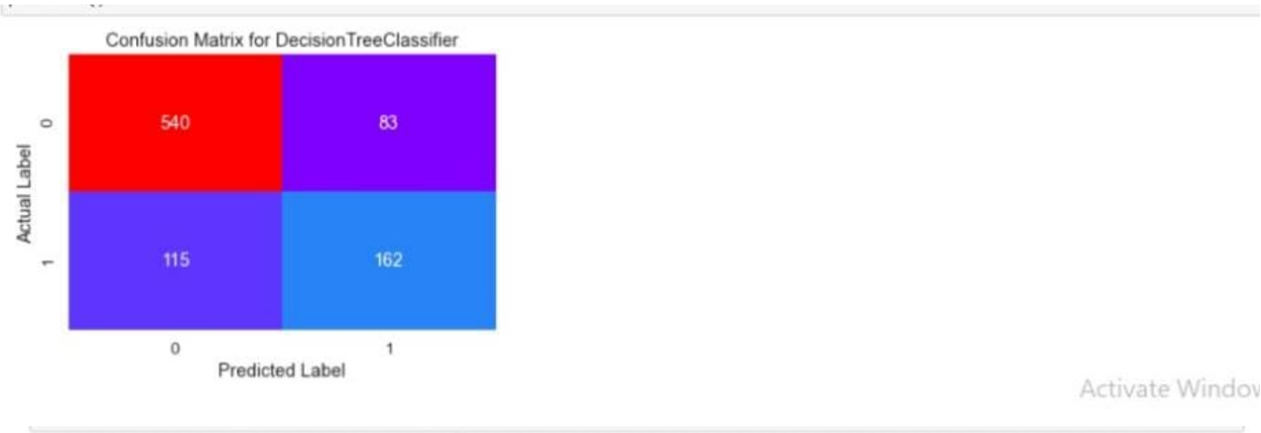
Confusion matrix for DecisionTreeClassifier for training data

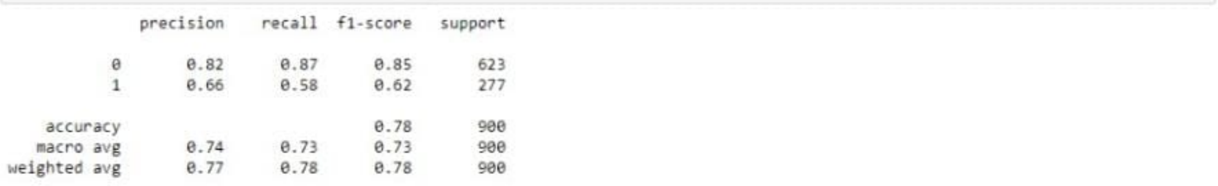




```
              precision    recall  f1-score   support

           0       0.83      0.88      0.86      1453
           1       0.69      0.60      0.65       647

    accuracy                           0.80      2100
   macro avg       0.76      0.74      0.75      2100
weighted avg       0.79      0.80      0.79      2100
```

Confusion matrix for DecisionTreeClassifier for test data





```
              precision    recall  f1-score   support

           0       0.82      0.87      0.85       623
           1       0.66      0.58      0.62       277

    accuracy                           0.78       900
   macro avg       0.74      0.73      0.73       900
weighted avg       0.77      0.78      0.78       900
```

Cart Conclusion

Train Data:

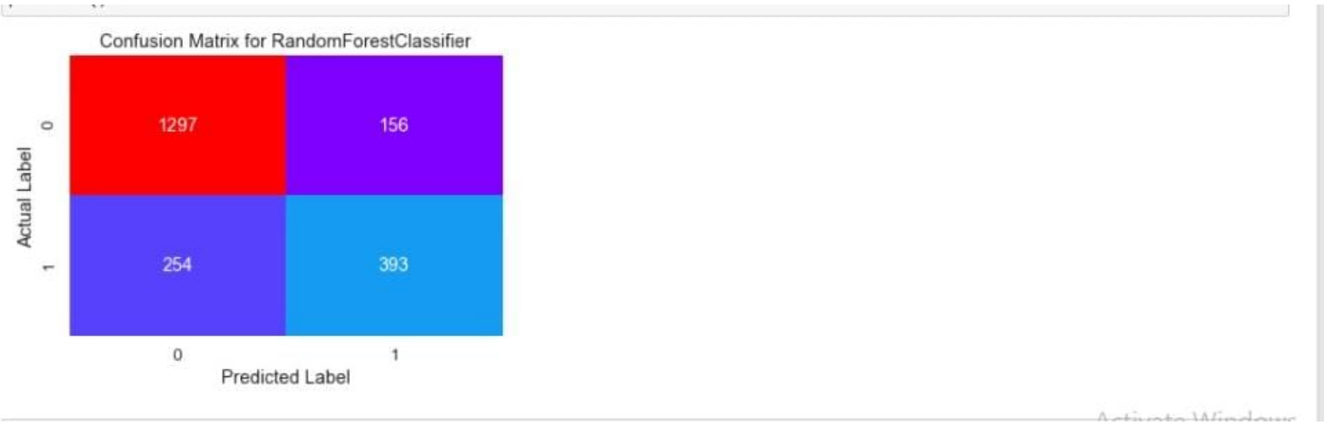
- AUC: 83%
- Accuracy: 79%
- Precision: 60%
- f1-Score: 65%

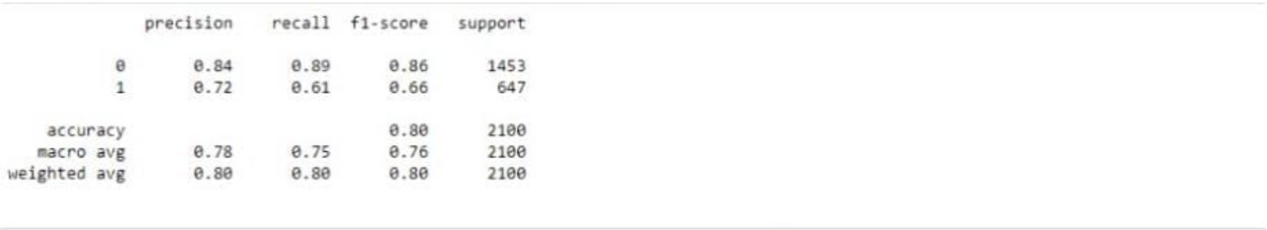
Test Data:

- AUC: 81%
- Accuracy: 78%
- Precision: 66%
- f1-Score: 62%

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

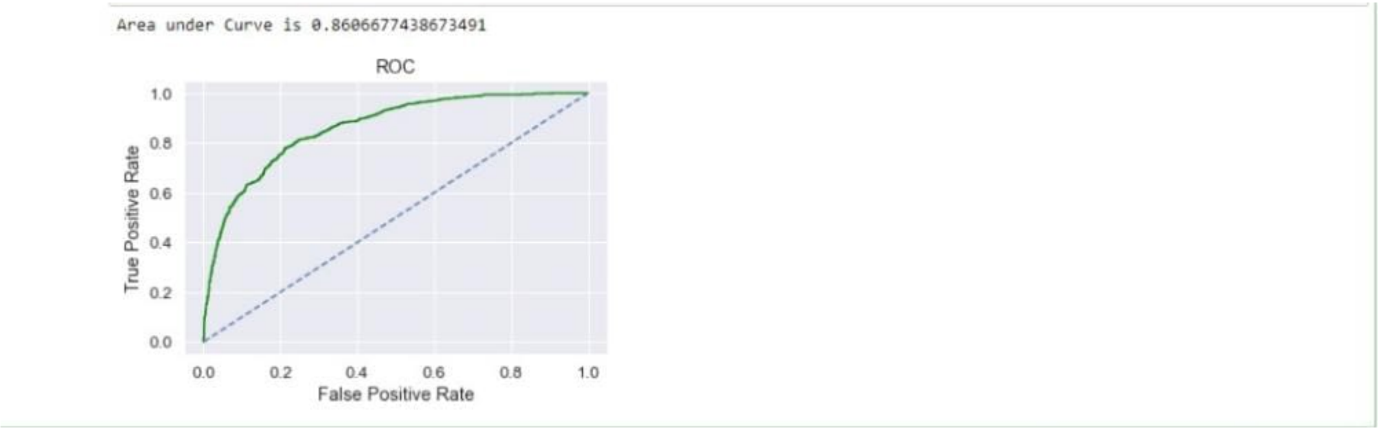
Confusion matrix for random forest for training data





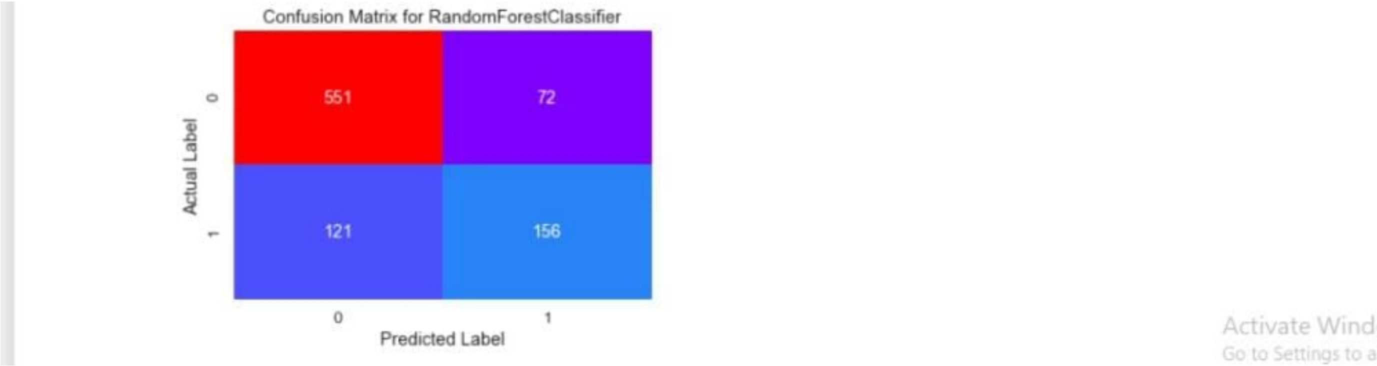
```
              precision    recall  f1-score   support

           0       0.84      0.89      0.86      1453
           1       0.72      0.61      0.66       647

    accuracy                           0.80      2100
   macro avg       0.78      0.75      0.76      2100
weighted avg       0.80      0.80      0.80      2100
```

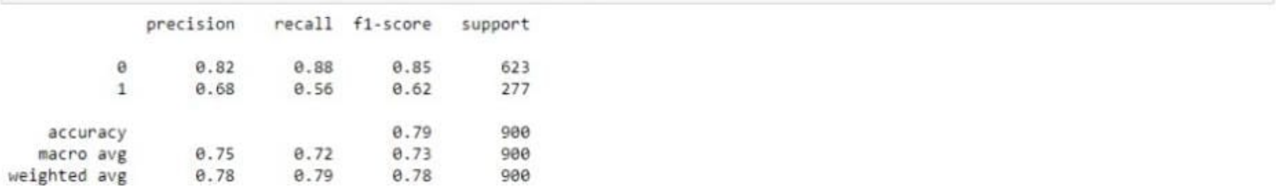
RF - AUC and ROC for the training data



```
Area under Curve is 0.8606677438673491
```

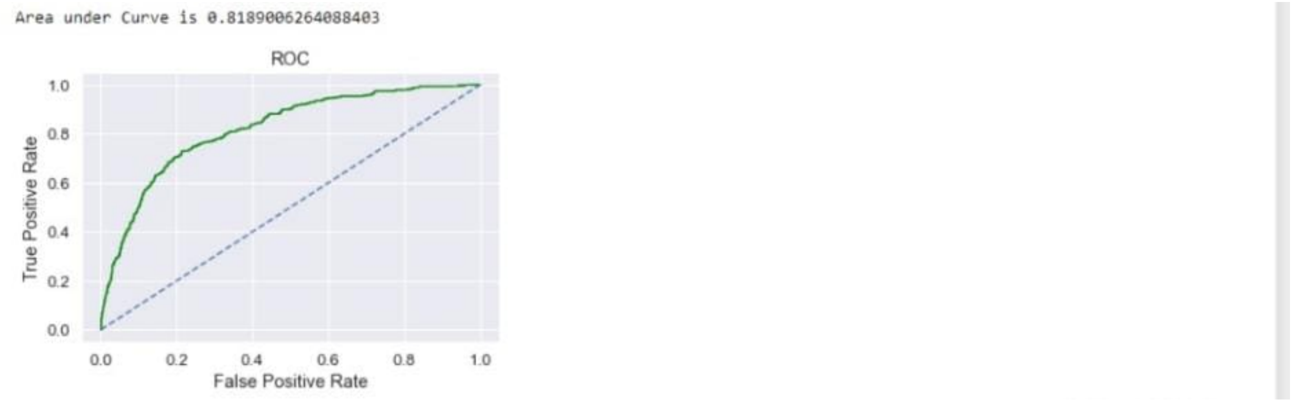
Confusion matrix for random forest for test data





```
              precision    recall  f1-score   support

           0       0.82      0.88      0.85       623
           1       0.68      0.56      0.62       277

    accuracy                           0.79       900
   macro avg       0.75      0.72      0.73       900
weighted avg       0.78      0.79      0.78       900
```

RF - AUC and ROC for the test data



```
Area under Curve is 0.8189006264088403
```

Random Forest Conclusion

Train Data:

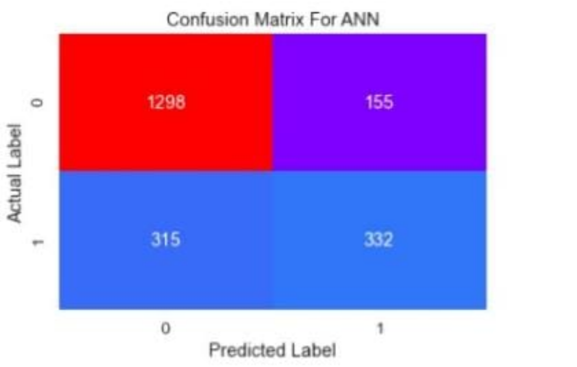
- AUC: 86%
- Accuracy: 80%
- Precision: 72%
- f1-Score: 66%

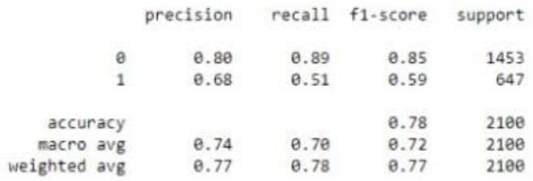
Test Data:

- AUC: 82%
- Accuracy: 78%
- Precision: 68%
- f1-Score: 62

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

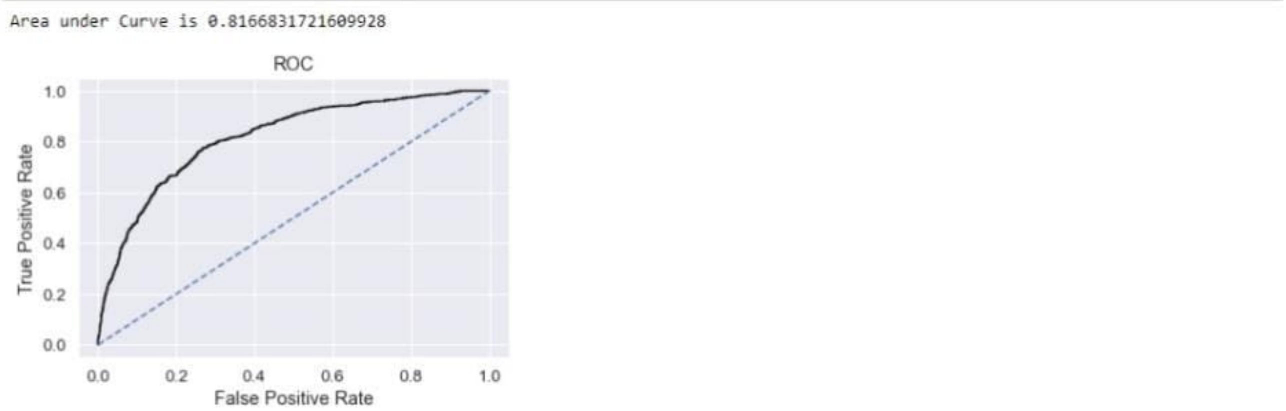
Neural Network Confusion Matrix for training data





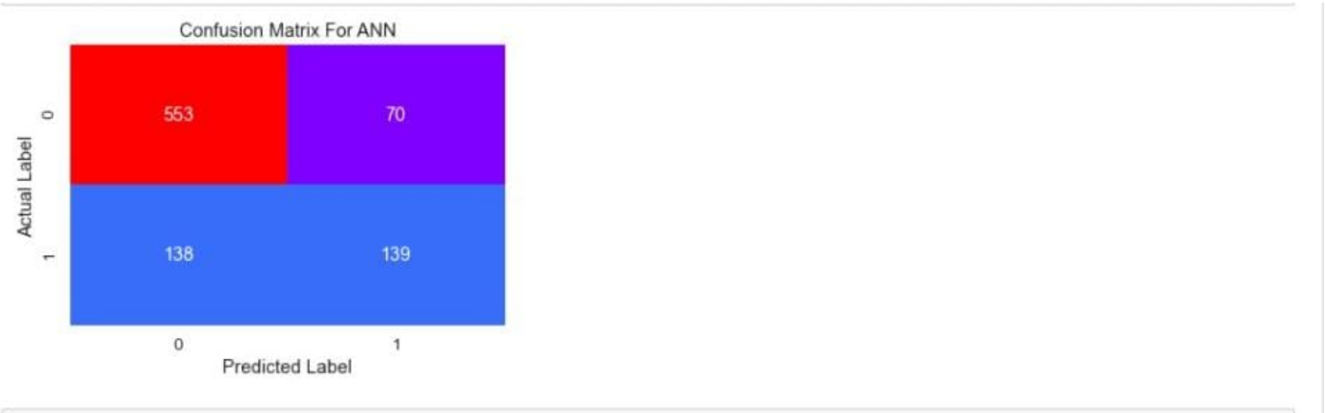
```
              precision    recall  f1-score   support

           0       0.80      0.89      0.85      1453
           1       0.68      0.51      0.59       647

    accuracy                           0.78      2100
   macro avg       0.74      0.70      0.72      2100
weighted avg       0.77      0.78      0.77      2100
```

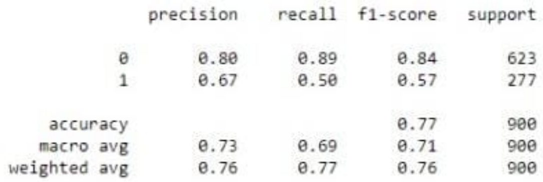
ANN - AUC and ROC for the training data



```
Area under Curve is 0.8166831721609928
```

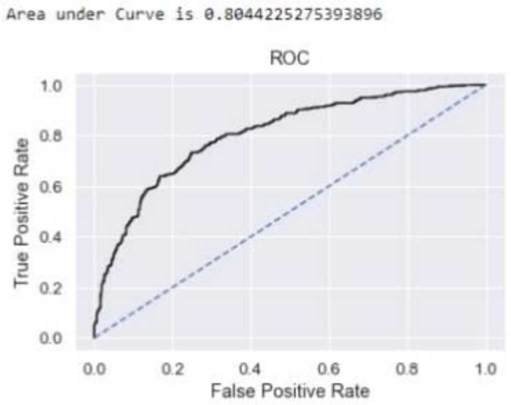
Neural Network Confusion Matrix for test data





```
              precision    recall  f1-score   support

           0       0.80      0.89      0.84       623
           1       0.67      0.50      0.57       277

    accuracy                           0.77       900
   macro avg       0.73      0.69      0.71       900
weighted avg       0.76      0.77      0.76       900
```

ANN - AUC and ROC for the test data



```
Area under Curve is 0.8044225275393896
```

Neural Network Conclusion

Train Data:

- AUC: 82%
- Accuracy: 78%
- Precision: 68%
- f1-Score: 59

Test Data:

- AUC: 80%
- Accuracy: 77%
- Precision: 67%
- f1-Score: 57%

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

From the above ROC curve and score for testing and training data we can say that:

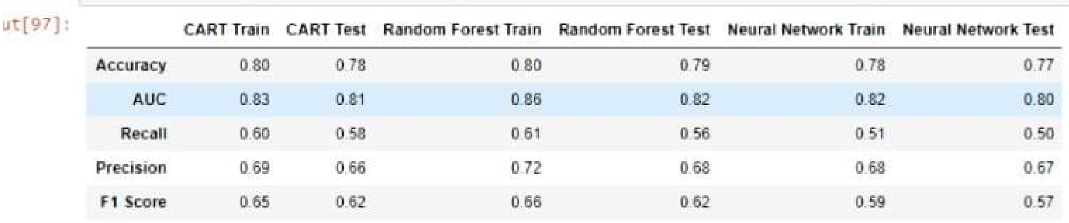
- RF perform better than ANN and decision tree on test set as the AUC is greater for Random Forest
- ANN has least accuracy score for traing set.
- The accuracy for ANN is 77 % where as for RFCL is 79 % and DTC is 78 % , So we can say that RFCL has more accuracy followed by DTC then ANN

We can work around n_estimators for random forest and try grid_search or increase layers in ANN, to make better predictions.

F1 score and Accuracy is better for Random Forest compared to other two so both are important compared to predictions.

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

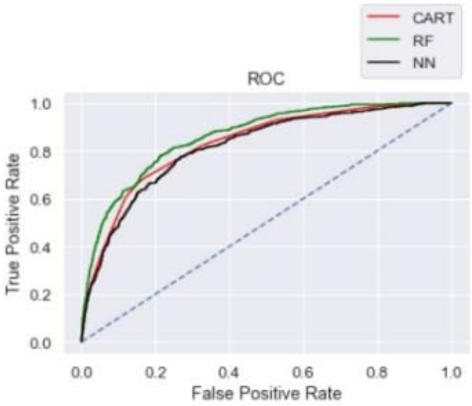
Comparison of the performance metrics from the 3 models



```
ut[97]:
```

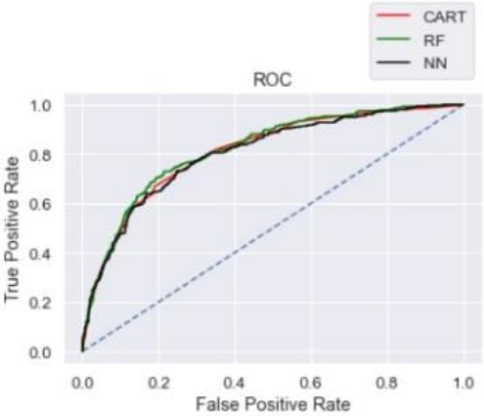
	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.80	0.78	0.80	0.79	0.78	0.77
AUC	0.83	0.81	0.86	0.82	0.82	0.80
Recall	0.60	0.58	0.61	0.56	0.51	0.50
Precision	0.69	0.66	0.72	0.68	0.68	0.67
F1 Score	0.65	0.62	0.66	0.62	0.59	0.57

ROC Curve for the 3 models on the Training data



ROC Curve for the 3 models on the Test data

```
Out[99]: <matplotlib.legend.Legend at 0x20a2937e9d0>
```



CONCLUSION :

I am selecting the RF model, as it has better accuracy, precsion, recall, f1 score better than other two CART & Neural Network.

Comparing all the models test data , we can say that Random Forest performs better than MLP Classifiers and Decision Tree Classifier from the overall ROC_AUC curve graph

The higher the AUC, The better the performance of the model,So the area under the curve is more for Random Forest it performs better.

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

- Using online the customers are benefitted, leading to an increase in conversions, which subsequently raised profits.

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

- Almost all the offline business has an associate claimed with it, so we need to limit offline business to a maximum. And increase the sales in online and make more conditions in claims applied. Because associates know how to get the claim approved, so there is a possibility they may add some cooked up documents.

- More sales happen via Agency than Airlines and the trend shows the claims are processed more at Airline.

- In silver plan and gold plan the sales is high as well as the claim is also high, so in order to reduce the number of claims, we need to increase the premium amount and change in plans so that the claims will be less.

- In Destination “Asia” the claim is high so, the insurance company has to apply some more conditions on the clients who travel to Asia in order to avoid more claims. We can include conditions like if accident or death occurs by travelling in particular mode of vehicle or during own travel plan other than the vehicle assigned by travel agency, we will reject the claim because most of the accidents in ASIA occurs in road.

- The travel agency C2B has claimed more compared to others, so we have to set those slabs or conditions for insurance claims.
- The JZI agency’s sales are low, we have to give suggestions to increase their claim sales, like they need to give marketing campaigns or to have tie up with travel agencies.
- Based on the model we are getting almost 80% accuracy, so we need customer books airline tickets or plans, and more number of data’s.

In [ ]: