greatlearning
Power Ahead

4/14/2021

DATAMINING PROJECT

PREPARED BY : ASHWIN KUMAR

Ashwin Kumar – PGDSBA –NOV'20

PROBLEM 1 : CLUSTERING

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 and do exploratory data analysis. Describe the data briefly.

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

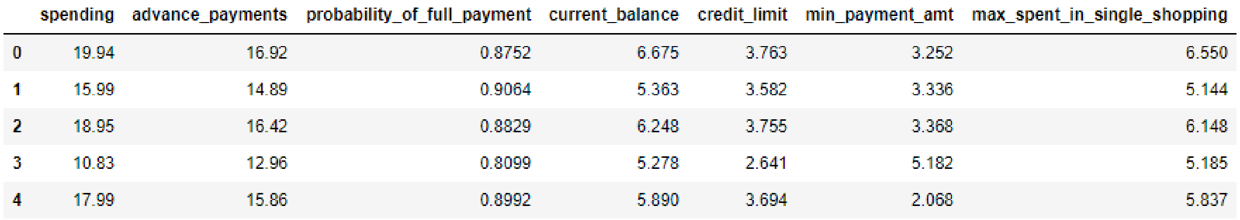
1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them

1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score.

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

1.1 Read the data and do exploratory data analysis. Describe the data briefly.

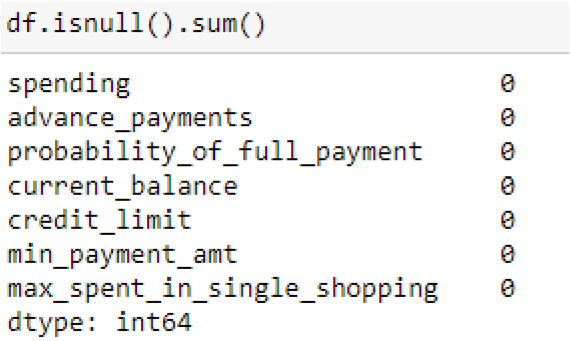
IMPORT DATASET & Loading the necessary Libraries



	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

CHECKING THE SHAPE & MISSING VALUES IN THE DATA SET

- *The Shape of the Data is (210,7)*
- *There were No Null Values in the Data*

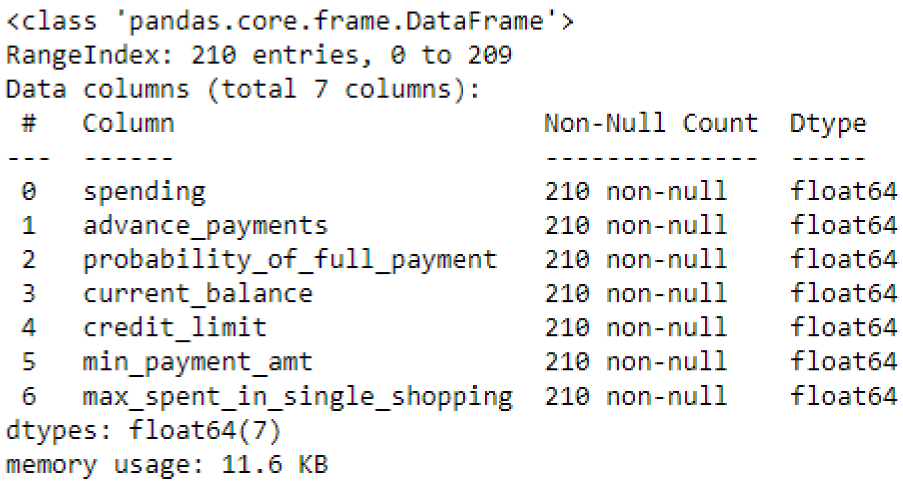


```
df.isnull().sum()
```

```
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
```

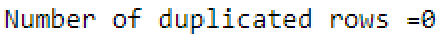
INFORMATION OF THE DATASET:

- ***From the data it was observed all the values are Float Values***



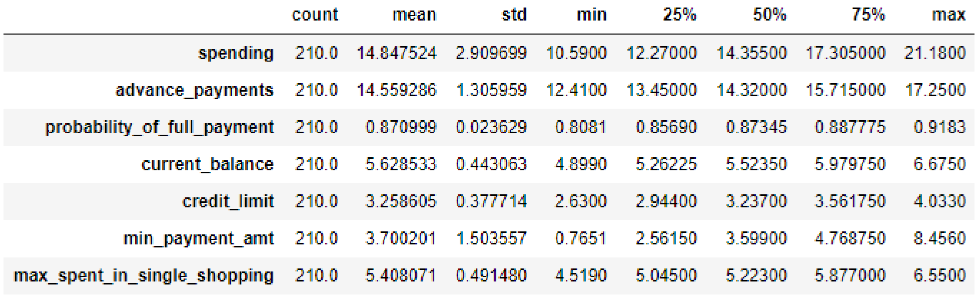
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   spending                      210 non-null    float64
 1   advance_payments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

DUPLICATES:



```
Number of duplicated rows =0
```

Summary of the dataframe:

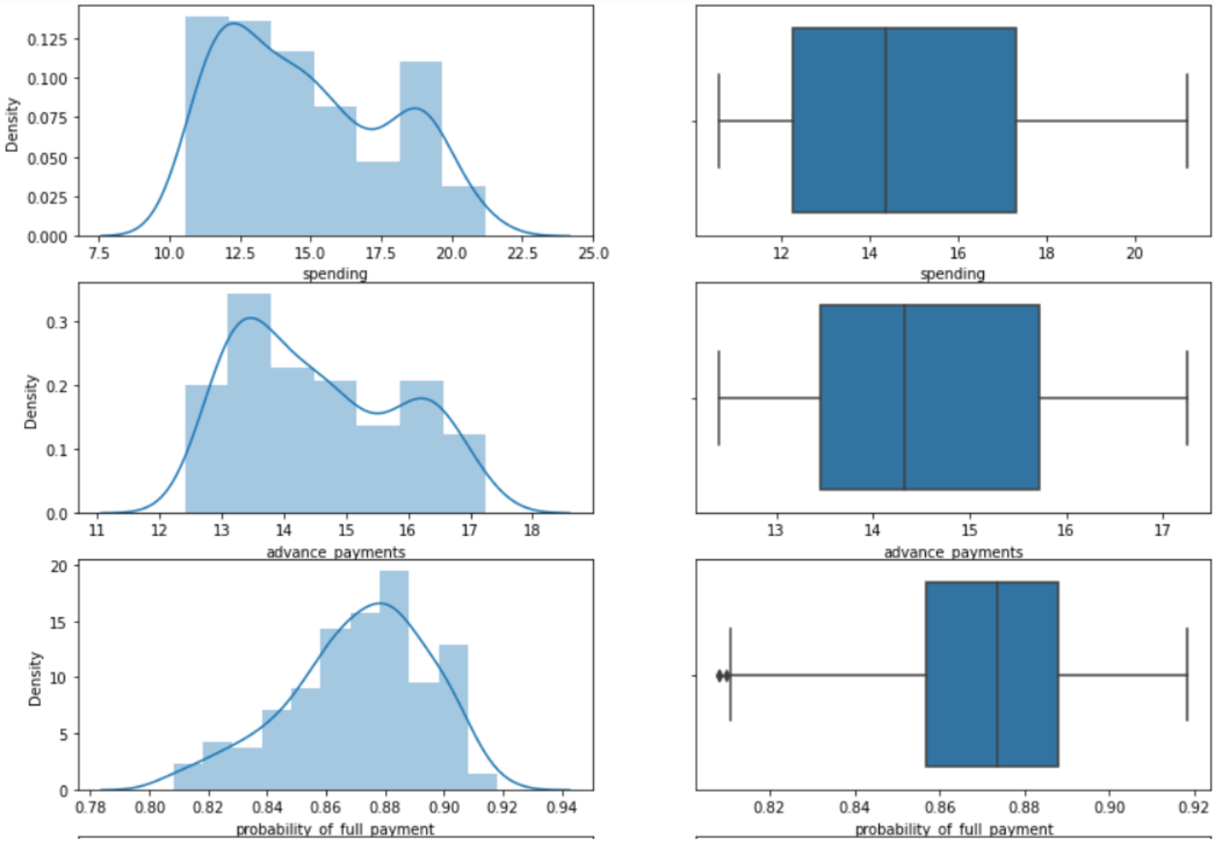


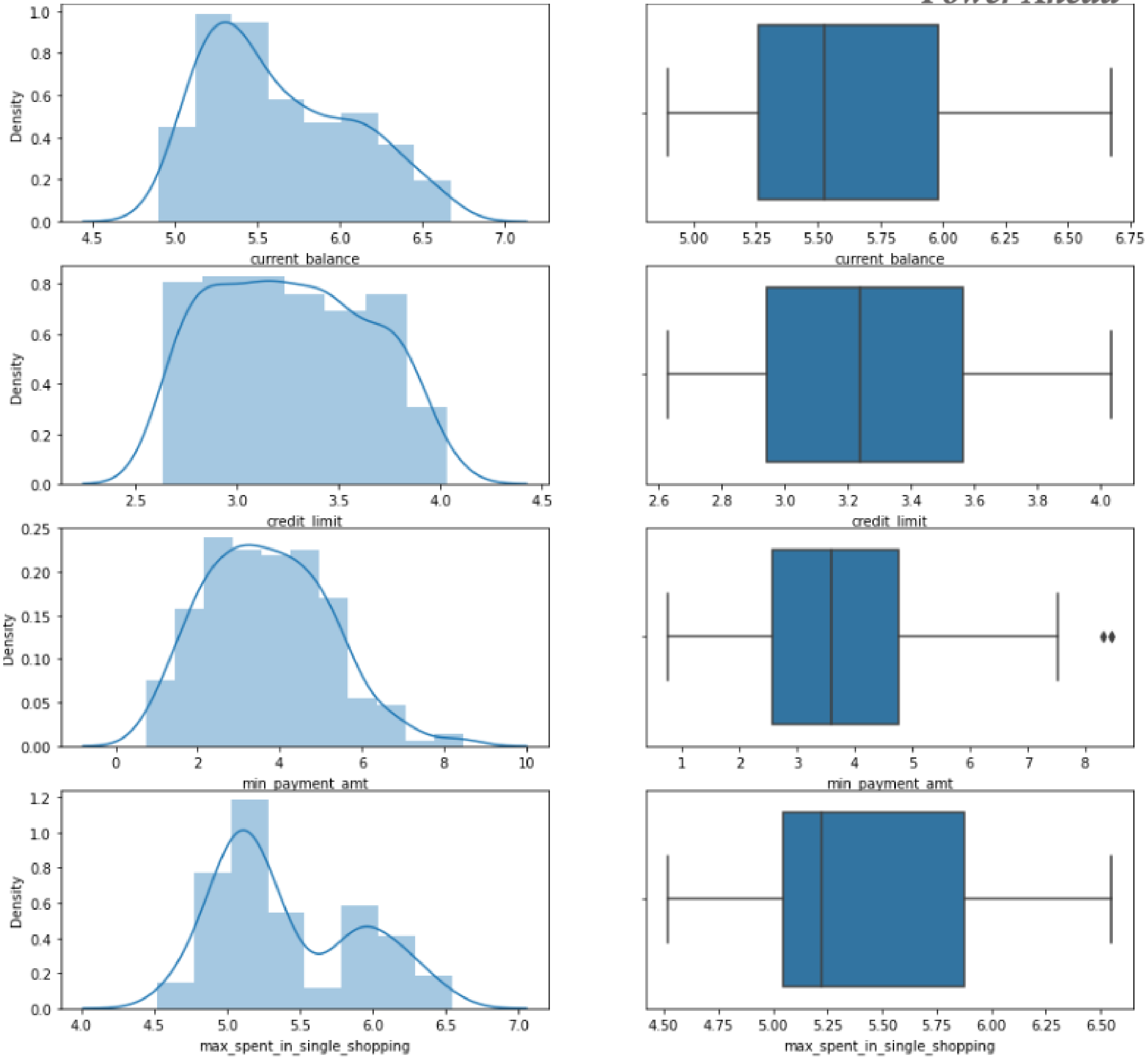
	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.847524	2.909699	10.5900	12.27000	14.35500	17.305000	21.1800
advance_payments	210.0	14.559286	1.305959	12.4100	13.45000	14.32000	15.715000	17.2500
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.85690	0.87345	0.887775	0.9183
current_balance	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 above summary,it can be seen that the Std Deviation of spending is high when compared with the other variables.Also,there are no null values present in any of the variables.

UNIVARIATE ANALYSIS :

- ***Boxplot is a indication of how the values in the data are spread out and also indicates if any outlier is present.***
- ***Displot gives the univariant set of observations .***

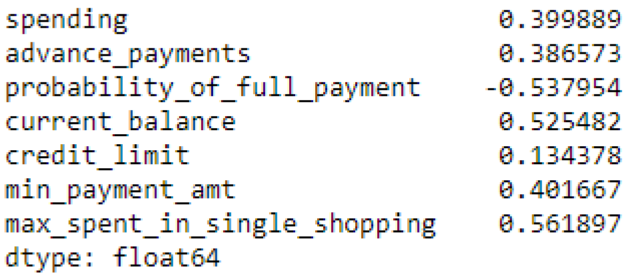

Density
0.125
0.100
0.075
0.050
0.025
0.000
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0
spending
12
14
16
18
20
spending
Density
0.3
0.2
0.1
0.0
11
12
13
14
15
16
17
18
advance_payments
13
14
15
16
17
advance_payments
Density
20
15
10
5
0
0.78
0.80
0.82
0.84
0.86
0.88
0.90
0.92
0.94
probability_of_full_payment
0.82
0.84
0.86
0.88
0.90
0.92
probability_of_full_payment



From ,the above Univariate Analysis,the data was summarised & patterns can be visualised. Outlier is present only in one variable which can be seen in min_payment_amt which clearly indicates that there are only a few customers whose minimum payment amount falls on the higher side on an average and probability_of_full_payment is in very low which indicates there are few customers whose probability to pay to the bank in full is on the lower side of average.

min_payment_amt & probability_of_full_payment variables alone have a very small Outliers,hence the requirement for Outlier Treatment is not necessary.

SKEWNESS:

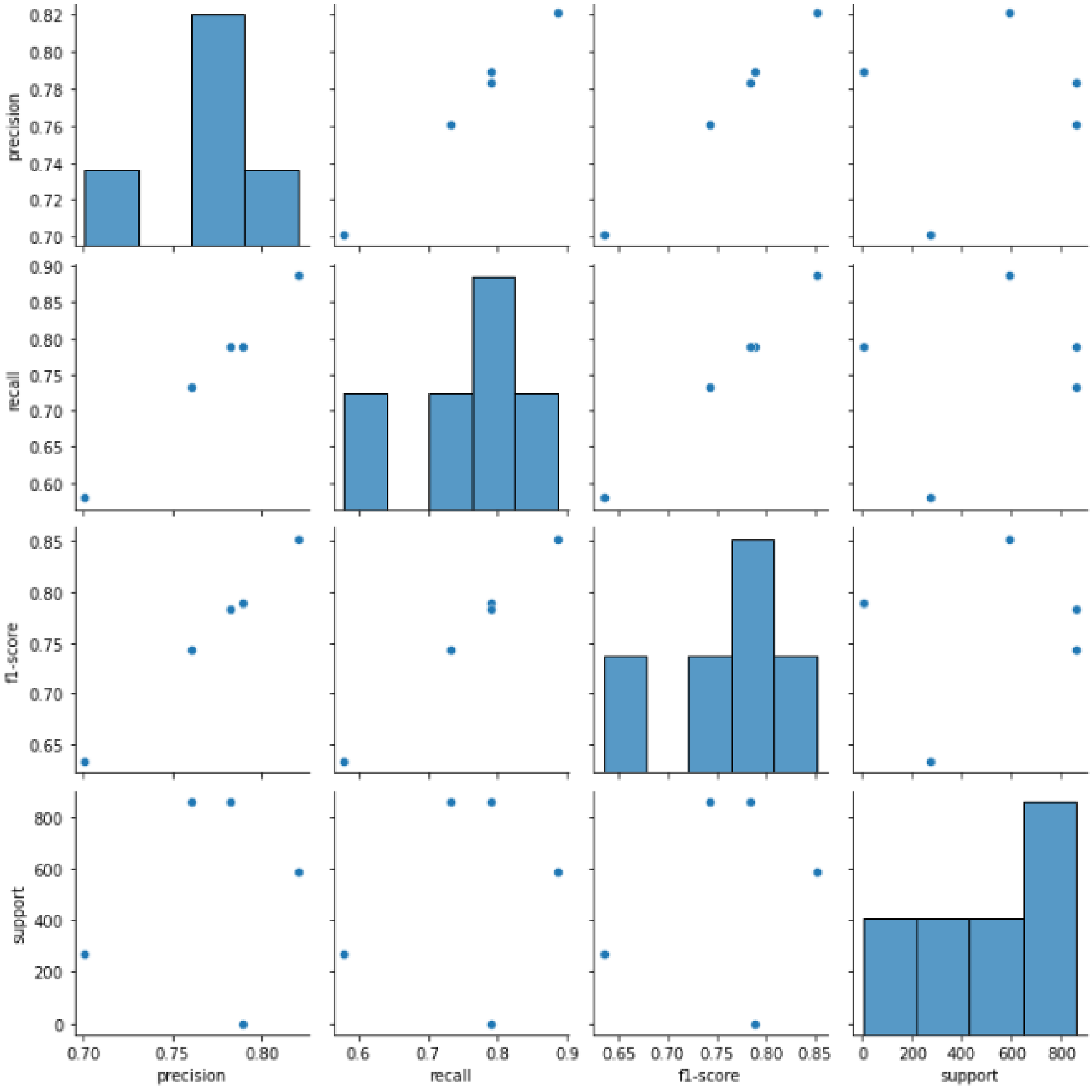


```
spending                        0.399889
advance_payments                0.386573
probability_of_full_payment    -0.537954
current_balance                 0.525482
credit_limit                    0.134378
min_payment_amt                 0.401667
max_spent_in_single_shopping    0.561897
dtype: float64
```

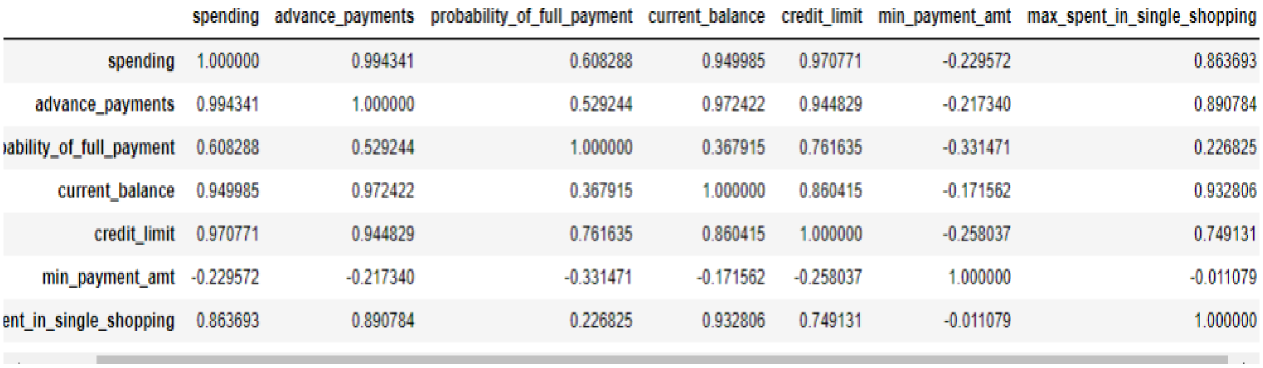
Probability of _full_payment is -0.537954 (negatively skewed).From,the displot Visualization the data distribution takes on from 0.80 to 0.92 which is a good factor.Only this Variable is negatively skewed apart from other variables which are positively skewed.

MULTIVARIVARIATE ANALYSIS:

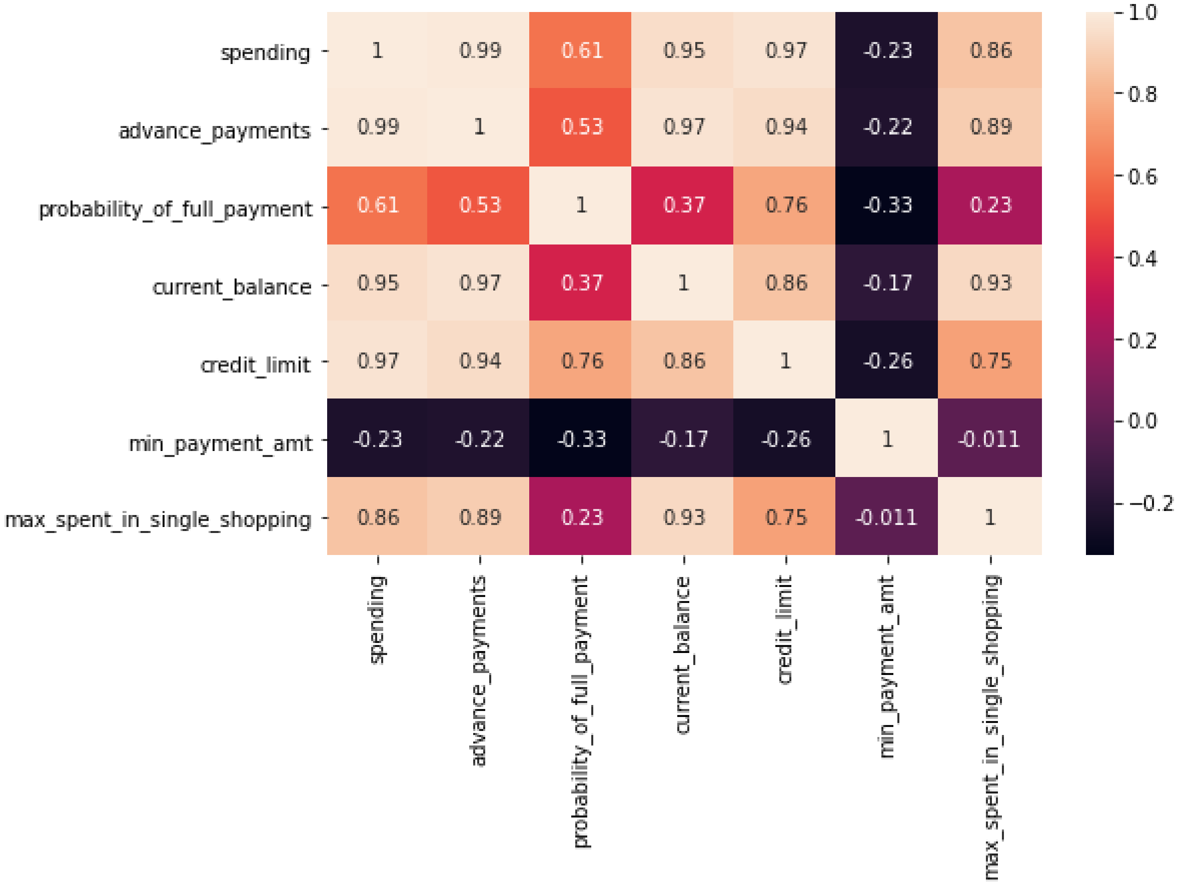
Check For Multi Collinearity



CORRELATION:



	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
ability_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



HEATMAP:

From the above HeatMap Visualization it can be seen that there is Strong Positive Correlation between Spending & advance_payments.Spending & current_balance are highly correlated.Spending & credit_limit are highly correlated.

Advance_payment and current_balance are highly correlated,Advance_payament and credit_limit are highly correlated. Advance_payment and max_spent_in_single_shopping are highly correlated current_balance and max_spent_in_single_shopping are highly correlated

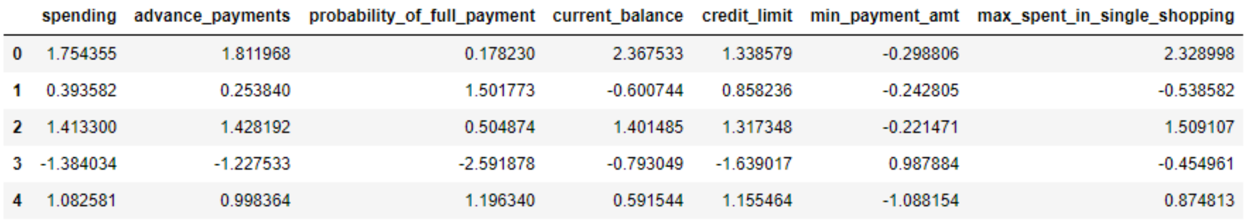
We can conclude based on the data that those who have high credit limit & those who have high Current Balance amount spends more.

Min_payment_amt is not correlated with any of the variables which wont influence changes in spending ,current balance or credit limit.

Probability of full payments are much higher for those customers who have a higher credit limit.

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

Yes, Scaling needs to be done as the values of the variables are in different scales.(I.E Values of the variables ranging from 100's to 1000's upto 10000's).In order to perform Analysis all of these different variables need to be converted to one scale .By performing Scaling,all of the values will fall in the same relative range

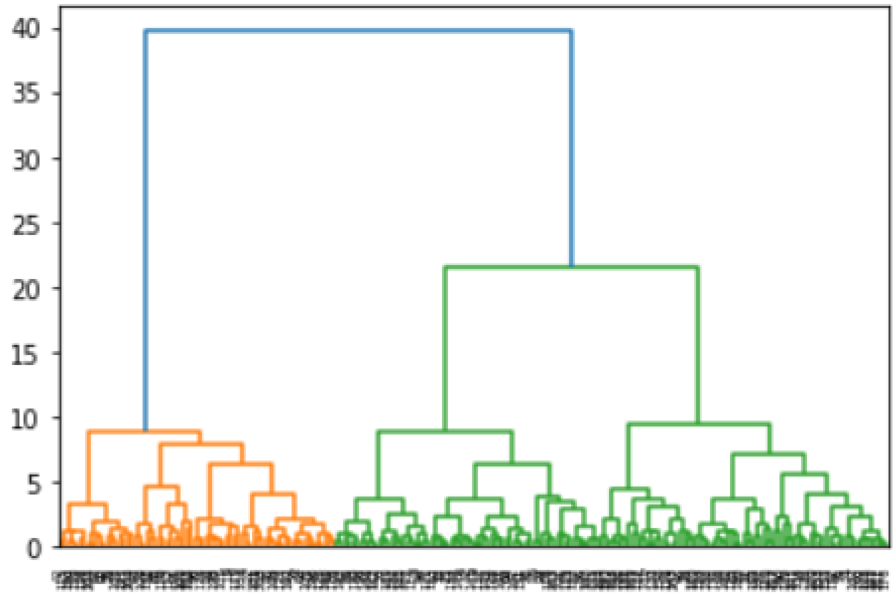


	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754355	1.811968	0.178230	2.367533	1.338579	-0.298806	2.328998
1	0.393582	0.253840	1.501773	-0.600744	0.858236	-0.242805	-0.538582
2	1.413300	1.428192	0.504874	1.401485	1.317348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

I have used Hierarchical clustering method to create optimum clusters & splitting the dataset based upon these clusters.

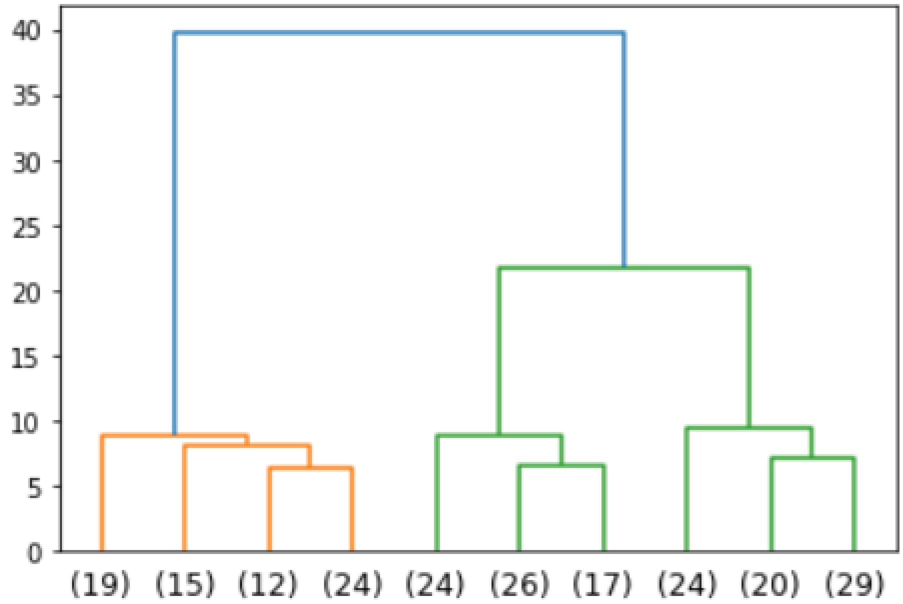
Hierarchical clustering relies using clustering techniques to find a hierarchy of clusters, where this hierarchy resembles a dendrogram.



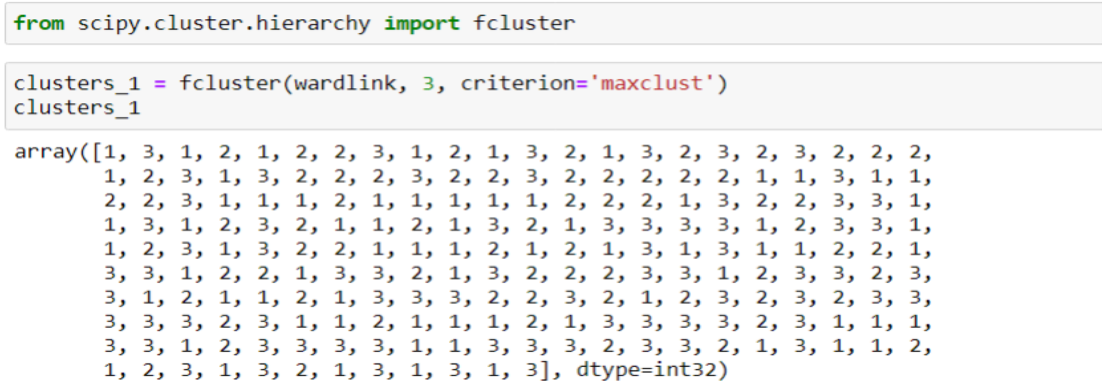
The above dendrogram indicates that all the data points have been clustered into different clusters by Ward's Method.

To find the optimum cluster to solve the problem further we can use truncatemode = lastp

<u>Cutting the Dendrogram with suitable clusters:</u>



Now,by using the above method we can see that the data has been clustered into 3 different Clusters.We can also visualize that the maximum customers fall into the green Cluster.

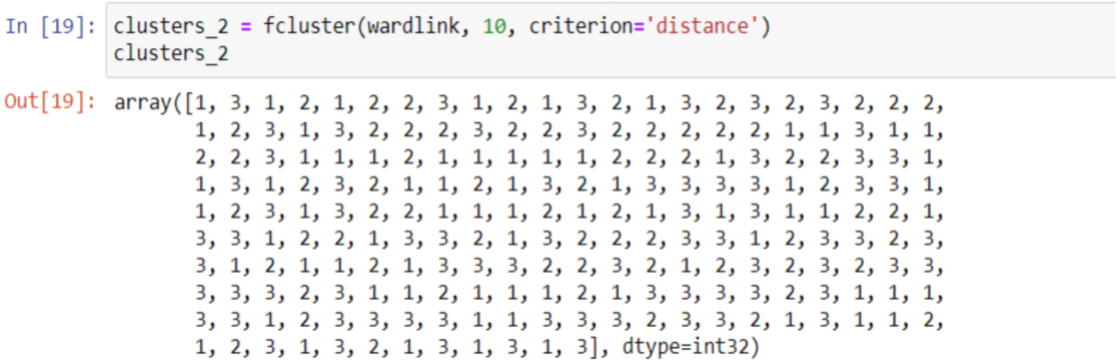


```
from scipy.cluster.hierarchy import fcluster
```

```
clusters_1 = fcluster(wardlink, 3, criterion='maxclust')
clusters_1
```

```
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)
```

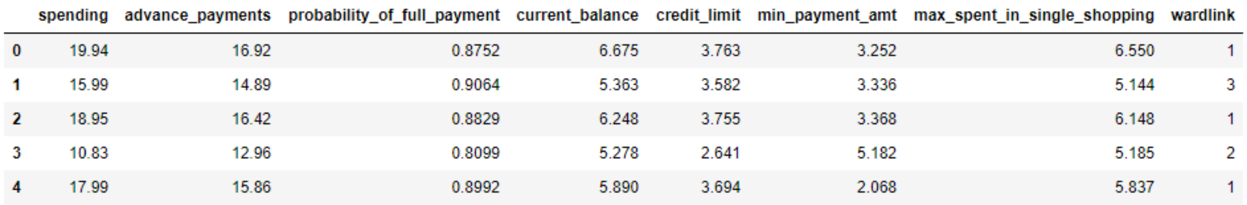
Next step is to append these clusters into the dataset. We can use : criterion = 'maxclust' where a cut is defined based on the number of clusters.



```
In [19]: clusters_2 = fcluster(wardlink, 10, criterion='distance')
         clusters_2

Out[19]: 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)
```

Next criterion = 'distance' where a cut is defined based on distance in the yaxis.

We can see that both the criterion have resulted in the same output.The distance criterion gives a glance to find the optimal no.of clusters.With the distance criterion we can see that the optimum no.of clusters would be 3.



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	wardlink
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	3
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	2
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837	1

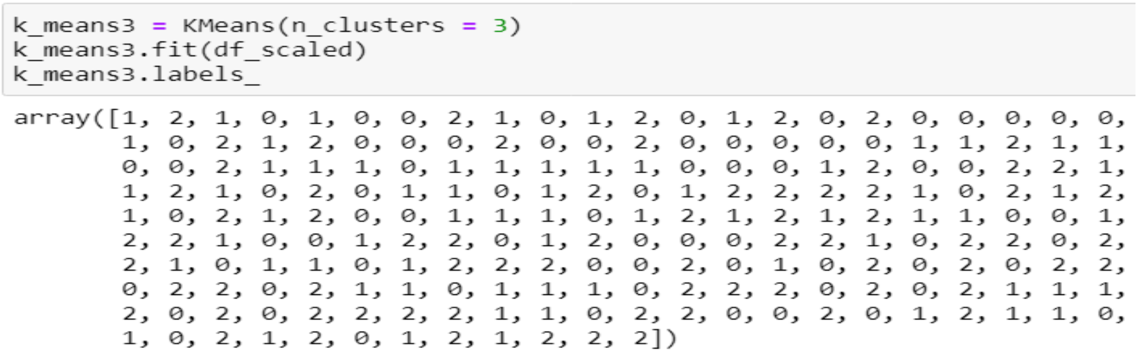
Appending these observations into the original data.

1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score.

In K-Means, each cluster is associated with a centroid. K-means is a centroid-based algorithm where we calculate the distances to assign a point to a cluster

We,apply K-Means Technique to the scaled dataset & identify the clusters formed. Then we will calculate the value of inertia and store it in WSS

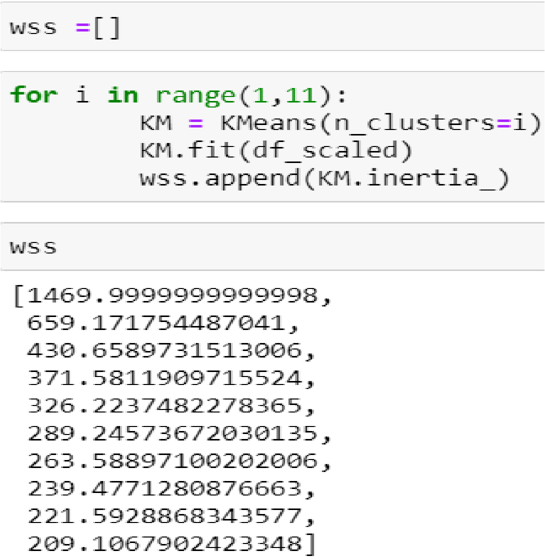
CLUSTER OUTPUT FOR ALL OBSERVATIONS:



```
k_means3 = KMeans(n_clusters = 3)
k_means3.fit(df_scaled)
k_means3.labels_

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])
```

We have 3 Clusters 0,1,2.

To further find the optimum number of clusters we can use k-elbow method.



```
wss =[]
```

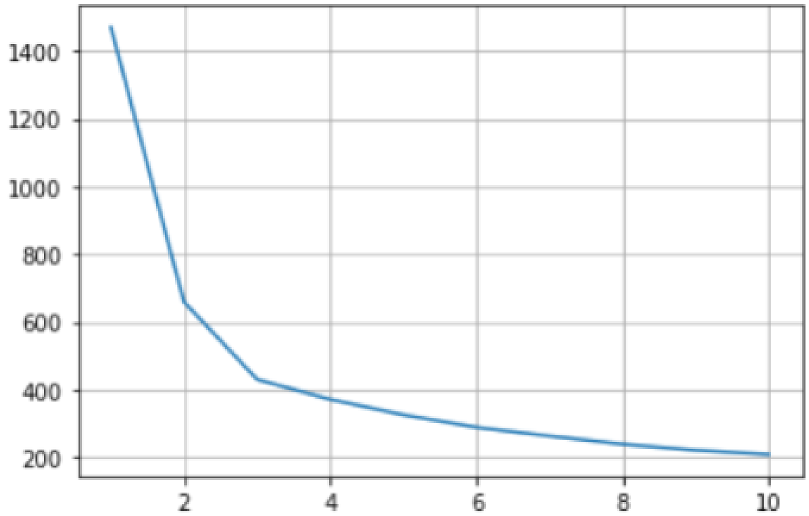
```
for i in range(1,11):
        KM = KMeans(n_clusters=i)
        KM.fit(df_scaled)
        wss.append(KM.inertia_)
```

```
wss
```

```
[1469.9999999999998,
 659.171754487041,
 430.6589731513006,
 371.5811909715524,
 326.2237482278365,
 289.24573672030135,
 263.58897100202006,
 239.4771280876663,
 221.5928868343577,
 209.1067902423348]
```

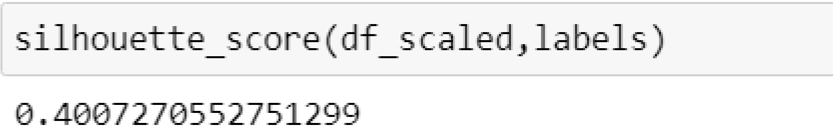
The above gives a clear idea on the inertia values of Clusters from 1 to 11.

The two methods to determine the optimal number of clusters are within sum of squares(wss) method and average silhouette scores method.



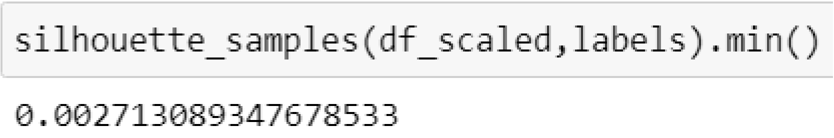
From the above visualisation the optimal number of clusters to be taken for k-means clustering is 3 since as per the elbow it can be easily seen in the curve that after 3 the curve gets flat.

Calculating the silhouette scores and silhouette width :



```
silhouette_score(df_scaled,labels)
```

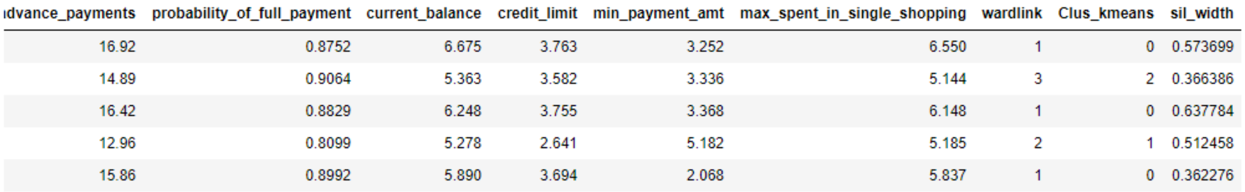
```
0.4007270552751299
```



```
silhouette_samples(df_scaled,labels).min()
```

```
0.002713089347678533
```

The Silhouette Score for 3 Clusters is better than nearest possible Optimum Clusters which is n_clusters = 4 , n_clusters=5.Also the elbow curve seen above shows,there is no huge drop in values so we can finalise on 3 clusters.Hence, As per wss method and silhouette score we can conclude that the optimal number of clusters is 3.The 3 Group Cluster Solution gives a pattern based on each categories of spending.



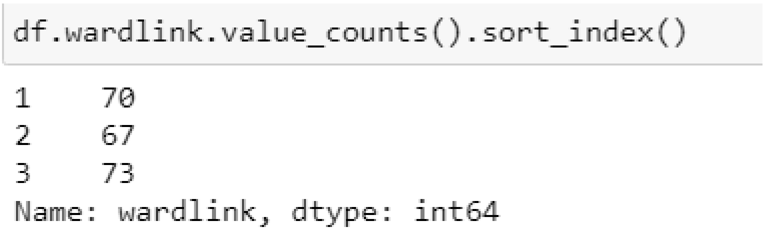
ıdvance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	wardlink	Clus_kmeans	sil_width
16.92	0.8752	6.675	3.763	3.252	6.550	1	0	0.573699
14.89	0.9064	5.363	3.582	3.336	5.144	3	2	0.366386
16.42	0.8829	6.248	3.755	3.368	6.148	1	0	0.637784
12.96	0.8099	5.278	2.641	5.182	5.185	2	1	0.512458
15.86	0.8992	5.890	3.694	2.068	5.837	1	0	0.362276

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

HIERARCHIAL CLUSTER PROFILING:

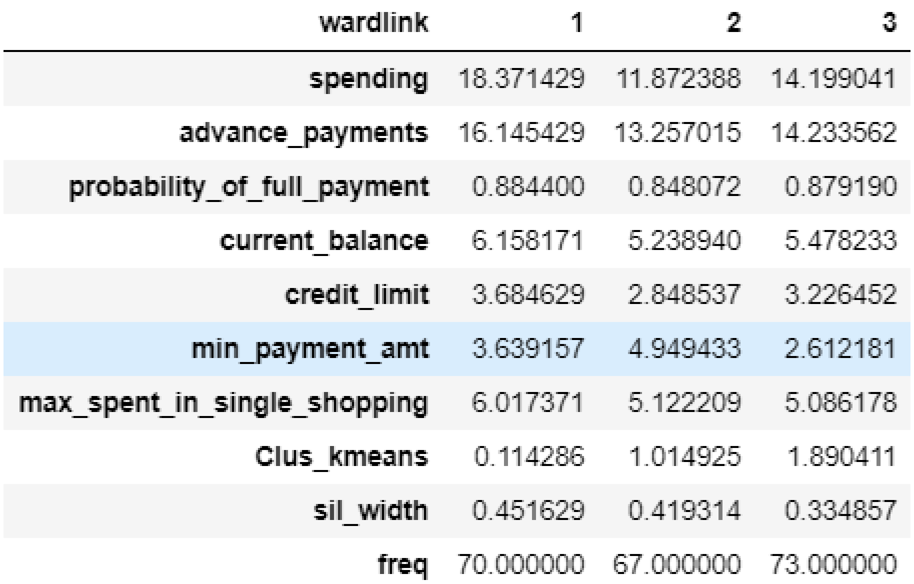
Cluster Group Profiles

Group 1 : High Spending ; Group 2 : Low Spending; Group 3 : Medium Spending



```
df.wardlink.value_counts().sort_index()
```

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



wardlink	1	2	3
spending	18.371429	11.872388	14.199041
advance_payments	16.145429	13.257015	14.233562
probability_of_full_payment	0.884400	0.848072	0.879190
current_balance	6.158171	5.238940	5.478233
credit_limit	3.684629	2.848537	3.226452
min_payment_amt	3.639157	4.949433	2.612181
max_spent_in_single_shopping	6.017371	5.122209	5.086178
Clus_kmeans	0.114286	1.014925	1.890411
sil_width	0.451629	0.419314	0.334857
freq	70.000000	67.000000	73.000000

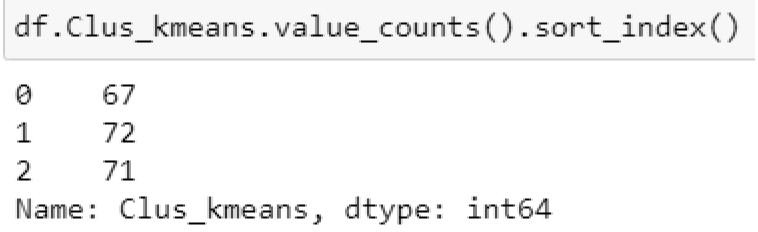
For Cluster 1,* *Spending is highest, averaging 18371 which is highest among three clusters. Highest advance payments around 1614 which is highest among three clusters Probability of Full Payment is very high, averaging around 0.8844 which is highest among three clusters.

For Cluster 2, The average Spending of this cluster is on lower side, averaging 11872. Highest advance payments around 1325 which is lowest among the three clusters. Probability of Full Payment is the least amongst other clusters, averaging around 0.848. Current Balance is around 5238 which is least among three clusters. Credit Limit is least for this cluster ranging around 28485. min_payment_amt is 494 which is max in this cluster. max_spent_in_single_shopping is around 5122.

For Cluster 3, Average Spending of this cluster is is 14199. Highest advance payments around 1423 Probability of Full Payment is on the higher side, averaging around 0.879. Current Balance is around 5478 which is average among three clusters. Credit Limit is around 32264 which is average among three clusters. min_payment_amt is 261 which is least among three cluster. Max_spent_in_single_shopping is the least around 5086.

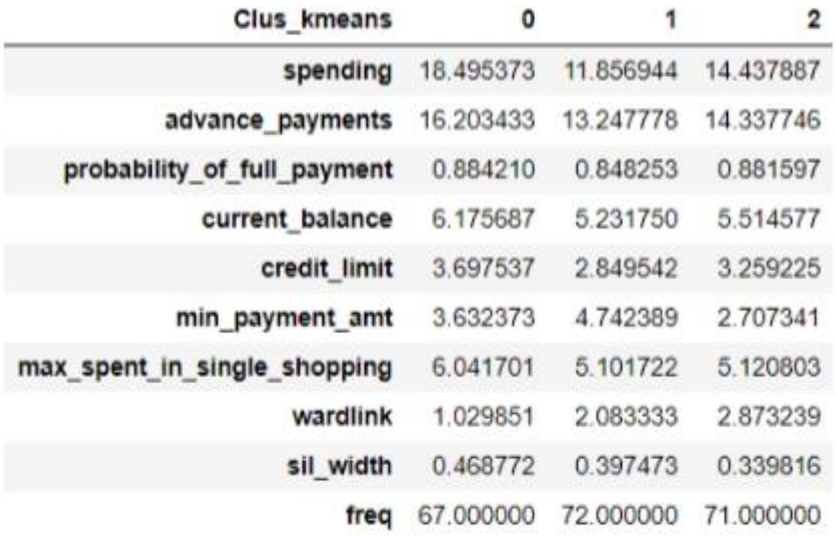
K_MEANS CLUSTERING PROFILE:

Group 0 : High Spending ; Group 1 : Low Spending; Group 2 : Medium Spending



```
df.Clus_kmeans.value_counts().sort_index()
```

```
0    67
1    72
2    71
Name: Clus_kmeans, dtype: int64
```



Clus_kmeans	0	1	2
spending	18.495373	11.856944	14.437887
advance_payments	16.203433	13.247778	14.337746
probability_of_full_payment	0.884210	0.848253	0.881597
current_balance	6.175687	5.231750	5.514577
credit_limit	3.697537	2.849542	3.259225
min_payment_amt	3.632373	4.742389	2.707341
max_spent_in_single_shopping	6.041701	5.101722	5.120803
wardlink	1.029851	2.083333	2.873239
sil_width	0.468772	0.397473	0.339816
freq	67.000000	72.000000	71.000000

PROMOTIONAL STRATEGY:

Hierarchial Clusters: ***Cluster 1 are premium customers .Bank should focus on Cluster 1 as the customers in this cluster have higher spending,highest advance payments,highest Current Balance and highest credit limit. This segment appears to be upper class and can be targeted using various offers such as cards with rewards and loyalty points for every spent . Increase there credit limit & Giving any reward points might increase their purchases.***

Cluster 2 are low spending customers ;Poor spending customers has the least Credit limit and so may be they spend least and also they have least current balance. Bank can also think of providing them offers for shopping at various websites & promotions that may increase their max_spent_in_single_shopping by tieing up with basic amenities groups such as groceries,utilities etc. Customers should be given remainders for payments often. This Cluster need huge promotions schemes & targets must be set in order to move customers from this cluster to Medium spending Cluster.

Cluster 3 medium spending customers . Bank should give customers in this Cluster more promotional offers because there are more chances that these customers may move to Cluster 1 of high spending. Increase spending habits by tieing with ecommerce sites ,premium hotels,luxury etc.Hence Bank should provide more promotional offers for customers & encourge them to spend more

For K-means Clusters:

Cluster 0 are premium customers .Bank should focus on Cluster 0 as the customers in this cluster have higher spending,highest advance payments,highest Current Balance and highest credit limit. This segment

appears to be upper class and can be targeted using various offers such as cards with rewards and loyalty purchases.

Cluster 1 are low spending customers ;Poor spending customers has the least Credit limit and so may be they spend least and also they have least current balance. Bank can also think of providing them offers for shopping at various websites & promotions that may increase their max_spent_in_single_shopping by tieing up with basic amenities groups such as groceries,utilities etc. Customers should be given remainders for payments often. This Cluster need huge promotions schemes & targets must be set in order to move customers from this cluster to Medium spending Cluster.

Cluster 2 medium spending customers . Bank should give customers in this Cluster more promotional offers because there are more chances that these customers may move high spending category of customers. Increase spending habits by tieing with ecommerce sites ,premium hotels,luxury etc.Hence Bank should provide more promotional offers for customers & encourge them to spend more

PROBLEM 2 : CART-RF-ANN

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.

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it.

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

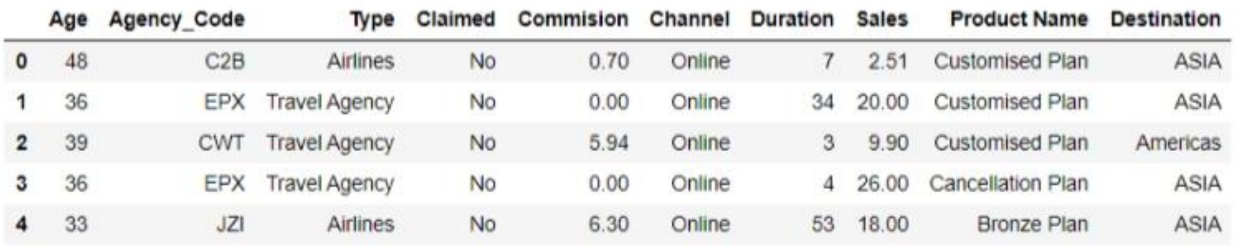
2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model

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

2.5 Inference: Basis on these predictions, what are the business insights and recommendations

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it.

Importing the Dataset and loading the necessary libraries.



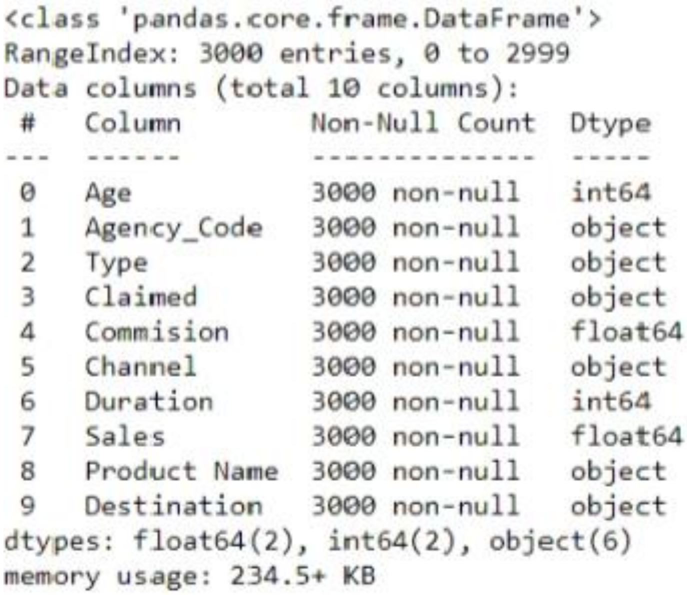
	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	C2B	Airlines	No	0.70	Online	7	2.51	Customised Plan	ASIA
1	36	EPX	Travel Agency	No	0.00	Online	34	20.00	Customised Plan	ASIA
2	39	CWT	Travel Agency	No	5.94	Online	3	9.90	Customised Plan	Americas
3	36	EPX	Travel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

SHAPE:

Shape : (3000, 9)

INFORMATION :

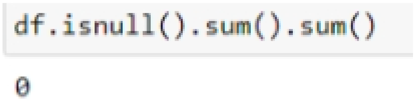
The Info of the dataset shows that the dataset contains integer,float & object values.Henceforth,we need to convert the object datatypes into a numeric value



```
<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
```

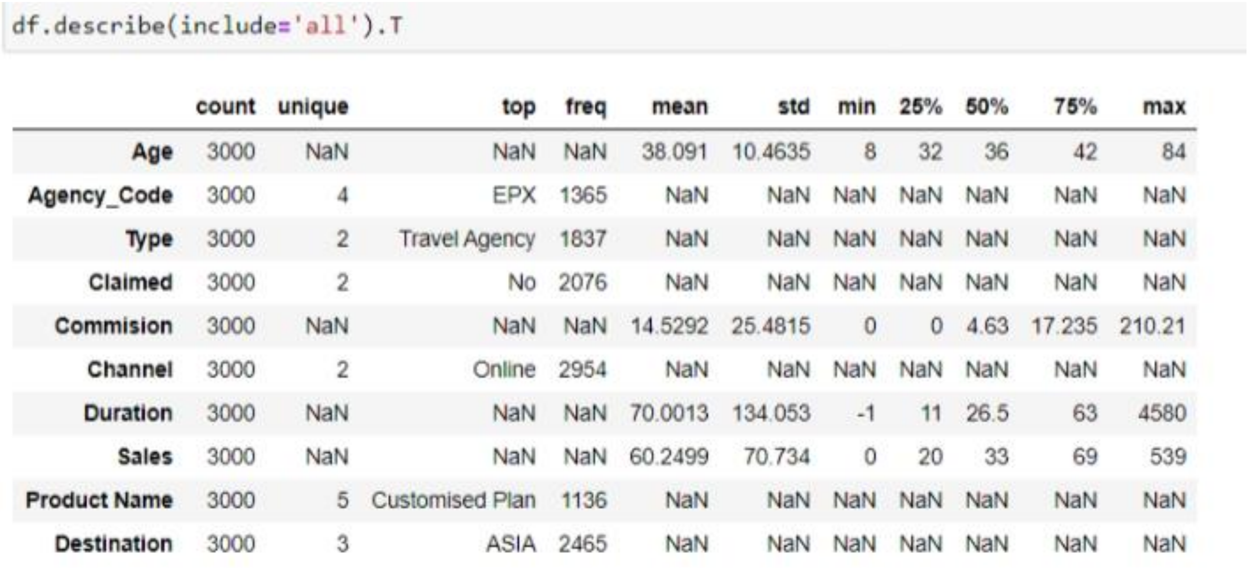
MISSING VALUES:

There were no missing values in the dataset;



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

0
```

SUMMARY :



```
df.describe(include='all').T
```

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	3000	NaN	NaN	NaN	38.091	10.4635	8	32	36	42	84
Agency_Code	3000	4	EPX	1365	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Type	3000	2	Travel Agency	1837	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Claimed	3000	2	No	2076	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Commision	3000	NaN	NaN	NaN	14.5292	25.4815	0	0	4.63	17.235	210.21
Channel	3000	2	Online	2954	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Duration	3000	NaN	NaN	NaN	70.0013	134.053	-1	11	26.5	63	4580
Sales	3000	NaN	NaN	NaN	60.2499	70.734	0	20	33	69	539
Product Name	3000	5	Customised Plan	1136	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Destination	3000	3	ASIA	2465	NaN	NaN	NaN	NaN	NaN	NaN	NaN

There are totally 4 numeric variables & 6 Categorical Variables.The Most Preferred type is Travel Agency;Channel is online;Customised Plans are most preffered by the customers.Destination ASIA is the preffered destination by the customers.

There are only 4 continuous variables Age,Commision,Duration and Sales, the result is shown for them only.

DUPLICATES:

Check for duplicate data:

Number of duplicate rows = 139

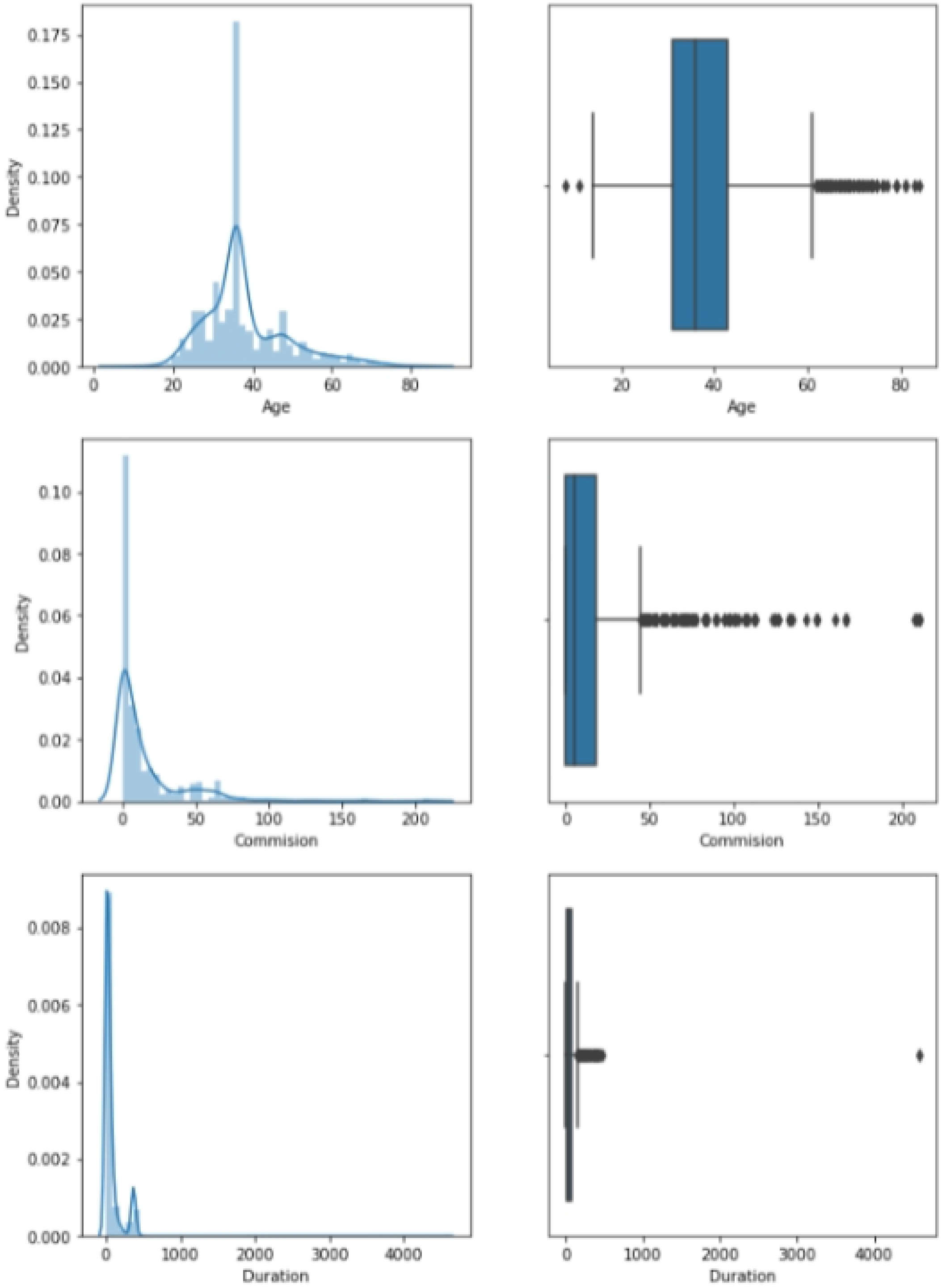
After Removing Duplicates:

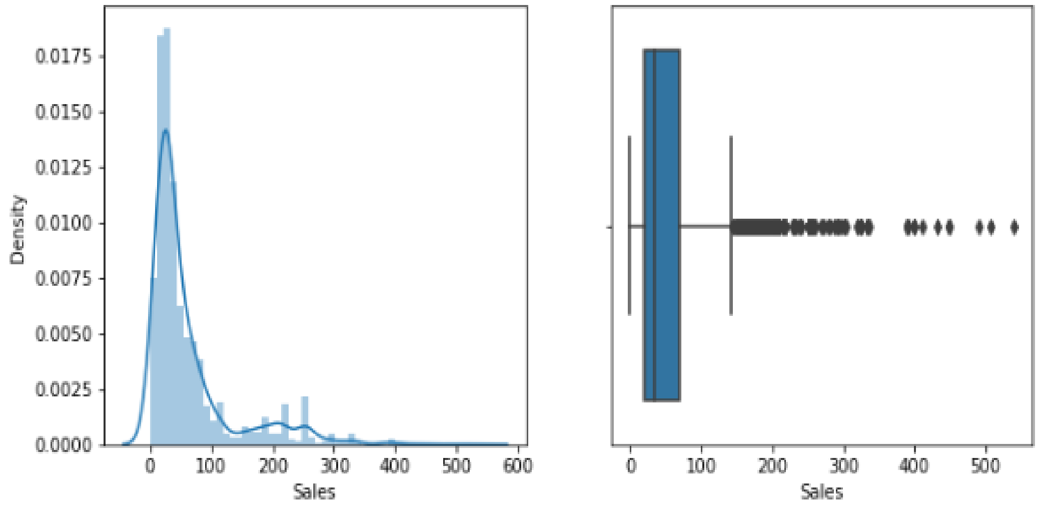
Number of duplicate rows = 0

SHAPE OF THE DATAFRAME:

(2861, 9)

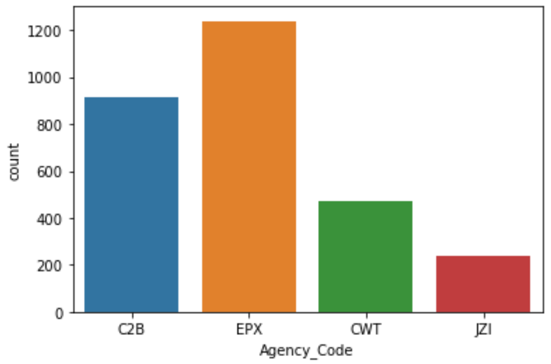
UNIVARIATE ANALYSIS & BIVARIATE ANALYSIS:


0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
Density
0
20
40
60
80
Age
20
40
60
80
Age
0.10
0.08
0.06
0.04
0.02
0.00
Density
0
50
100
150
200
Commision
0
50
100
150
200
Commision
0.008
0.006
0.004
0.002
0.000
Density
0
1000
2000
3000
4000
Duration
0
1000
2000
3000
4000
Duration

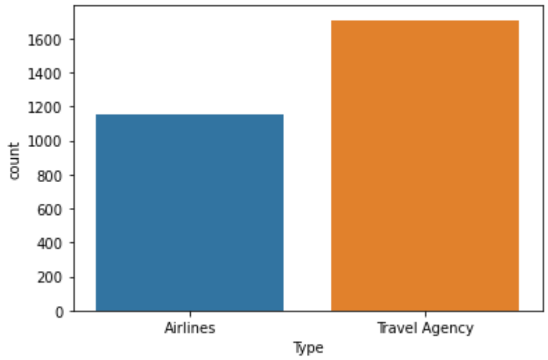


From the above visualization,we can see that all the variables has outliers & all the variables are positively skewed.Through,distplot we can see the pattern of data distribution.

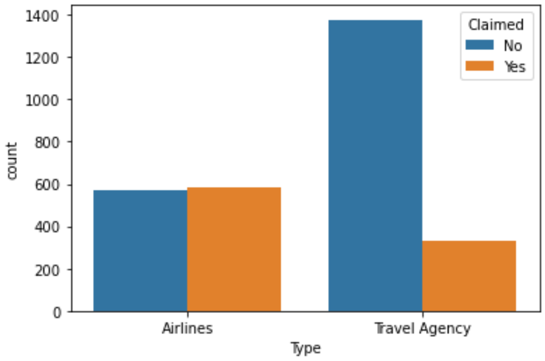
CATEGORICAL VARIABLES:



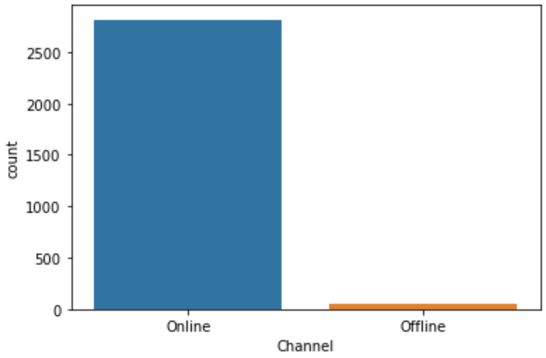
From the above,we can see that Agency EPX has higher frequency



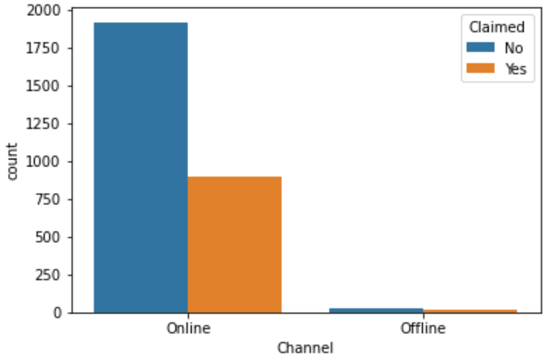
Travel Agencies have been more Sales over Airlines



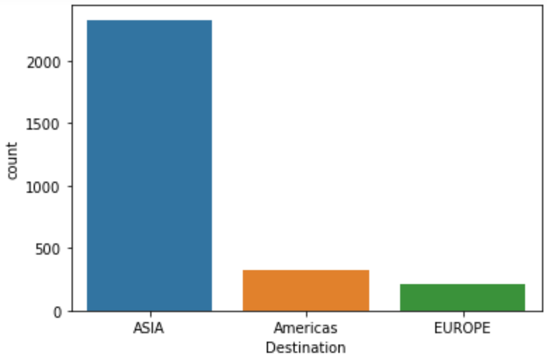
Insurance claim in Airlines,has been on higher side comparatively to Travel Agency which had higher Sale frequencies than Airlines.



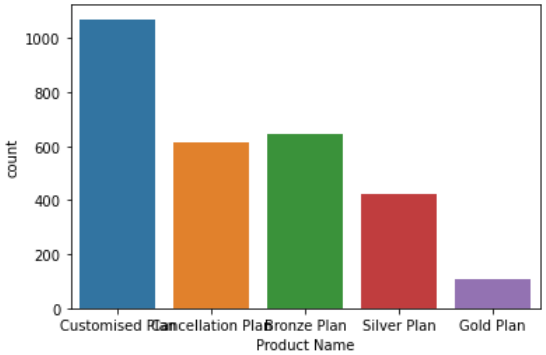
Channel Online is very much used than offline



Insurance Claim has been more in Online



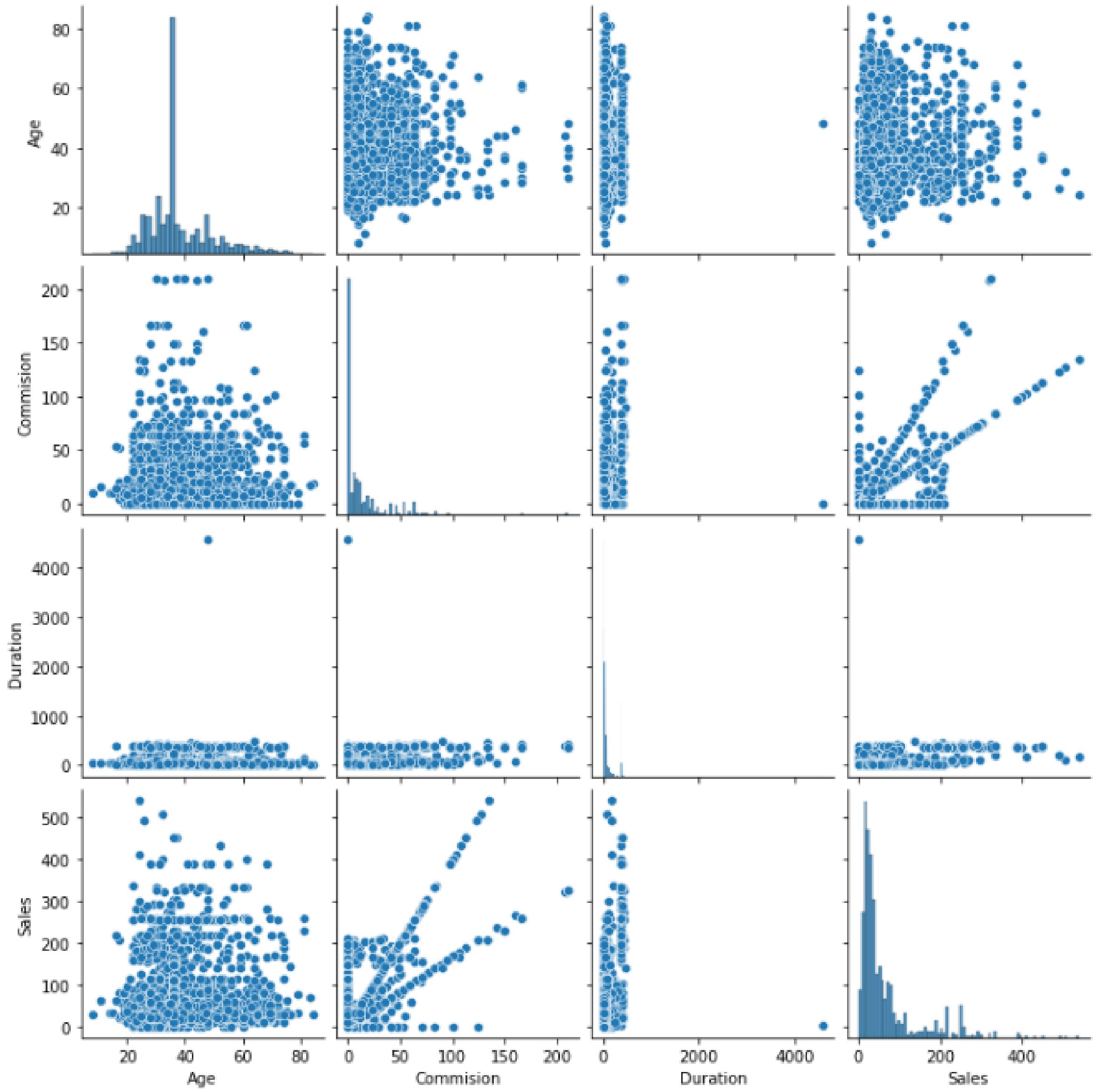
Most Sale & Most Claimed Destination is ASIA



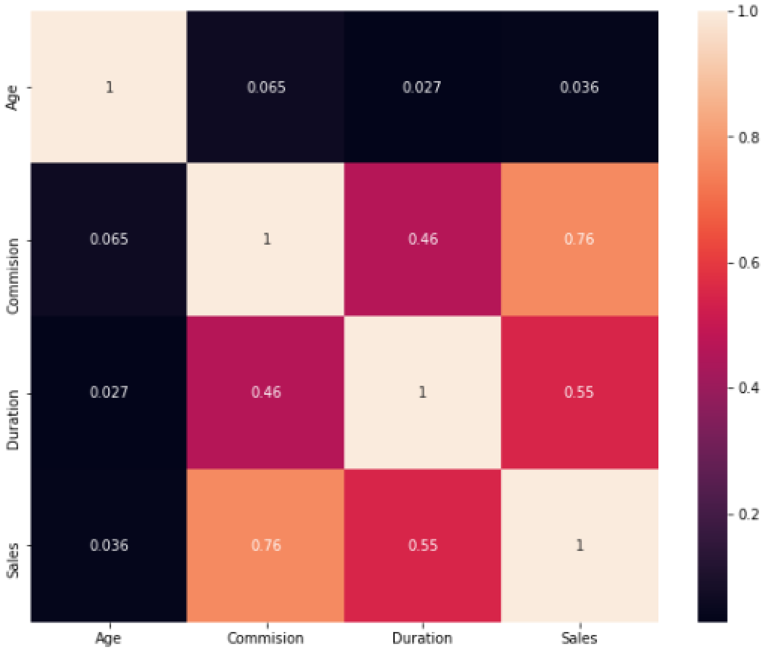
Customized Plans are more sought by Customers than the rest plans.

Further to our inference,we can see that maximum of the customers doing a claim in our data belong to age group of 30-50 years and it is observed that age group between 30-40 contribute to the highest number of claims , the type of Tour Agency was Travel Agency, Product name was Customised Plan , Channel was Online and Destination was Asia.

MULTIVARIATE ANALYSIS:

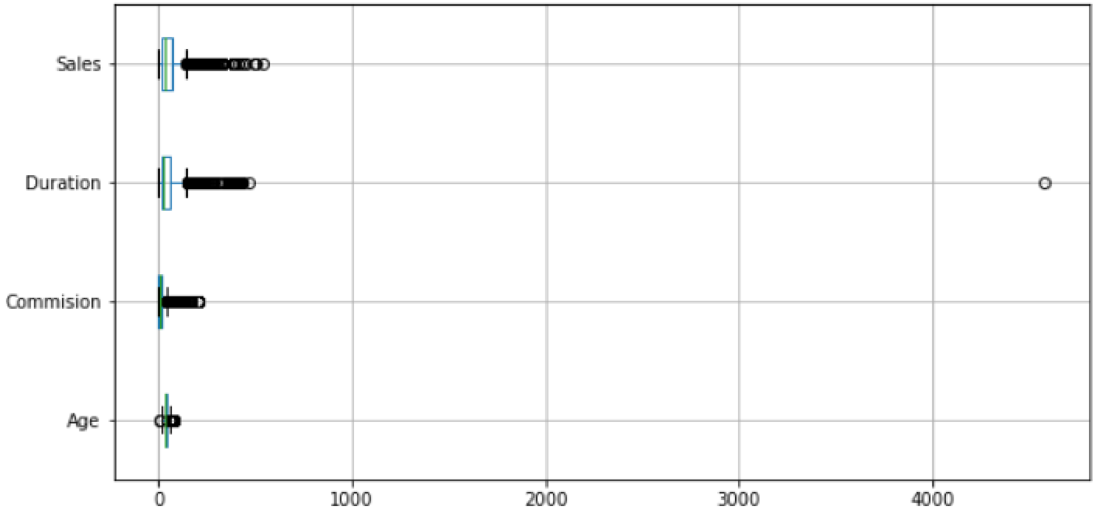

Age
80
60
40
20
Commision
200
150
100
50
0
Duration
4000
3000
2000
1000
0
Sales
500
400
300
200
100
0
20
40
60
80
Age
0
50
100
150
200
Commision
0
2000
4000
Duration
0
200
400
Sales

HEATMAP:

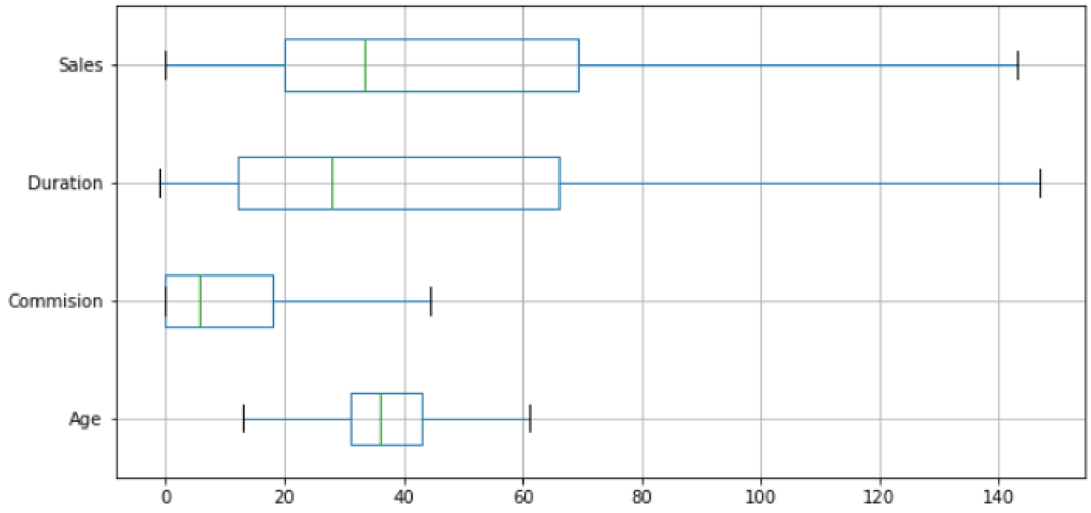


From the above Heatmap, we can see there is negative correlation & only positive correlation is observed. Not much multi-collinearity was observed.

CHECKING FOR OUTLIERS:



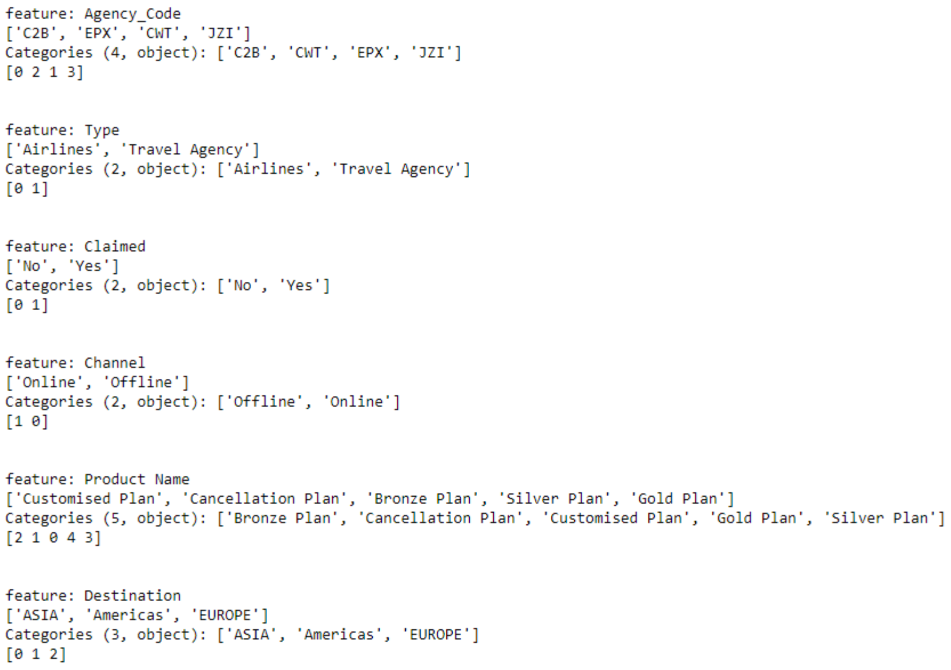
TREATING OUTLIERS:



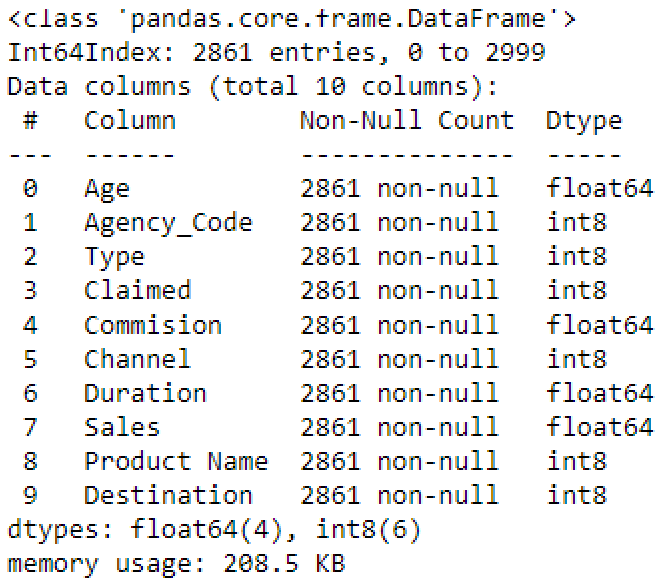
By checking the Boxplots for all the continuous variables we can conclude that a very high number of outliers are present in all the continuous variables .Hence, it is better to remove the present outliers which lies in our dataset to further build the models for the problem.

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

*The Models which we are going to build does not take in objects but only Numerics. Hence **object data types are converted into Categorical variables.***

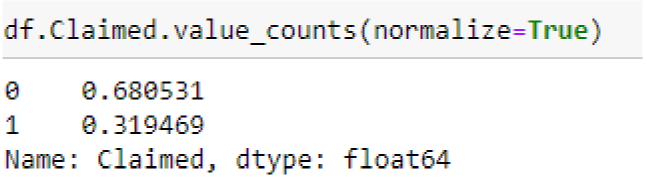


```
feature: Agency_Code
['C2B', 'EPX', 'CWT', 'JZI']
Categories (4, object): ['C2B', 'CWT', 'EPX', 'JZI']
[0 2 1 3]


feature: Type
['Airlines', 'Travel Agency']
Categories (2, object): ['Airlines', 'Travel Agency']
[0 1]


feature: Claimed
['No', 'Yes']
Categories (2, object): ['No', 'Yes']
[0 1]


feature: Channel
['Online', 'Offline']
Categories (2, object): ['Offline', 'Online']
[1 0]


feature: Product Name
['Customised Plan', 'Cancellation Plan', 'Bronze Plan', 'Silver Plan', 'Gold Plan']
Categories (5, object): ['Bronze Plan', 'Cancellation Plan', 'Customised Plan', 'Gold Plan', 'Silver Plan']
[2 1 0 4 3]


feature: Destination
['ASIA', 'Americas', 'EUROPE']
Categories (3, object): ['ASIA', 'Americas', 'EUROPE']
[0 1 2]
```



```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2861 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Age           2861 non-null   float64
 1   Agency_Code   2861 non-null   int8   
 2   Type          2861 non-null   int8   
 3   Claimed       2861 non-null   int8   
 4   Commision     2861 non-null   float64
 5   Channel       2861 non-null   int8   
 6   Duration      2861 non-null   float64
 7   Sales         2861 non-null   float64
 8   Product Name  2861 non-null   int8   
 9   Destination   2861 non-null   int8   
dtypes: float64(4), int8(6)
memory usage: 208.5 KB
```

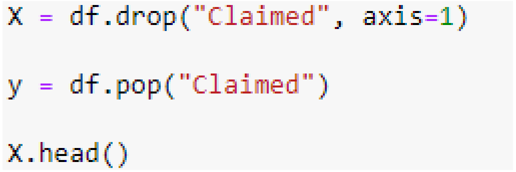
Now, all the objects variables have been converted into Categorical ones.



```
df.Claimed.value_counts(normalize=True)
```

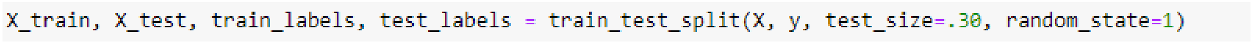
```
0    0.680531
1    0.319469
Name: Claimed, dtype: float64
```

<u>Claimed is our Target Variable. Extracting the target column into separate vectors for training set and test set</u>



```
X = df.drop("Claimed", axis=1)

y = df.pop("Claimed")

X.head()
```

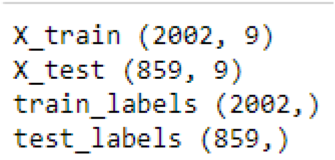
Splitting the data into Train and Test set:



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

Test size we have given as 0.30 as we want 30% of the data is to be tested. Random state we have given as 1,as it will give similar results with similar random state respectively.

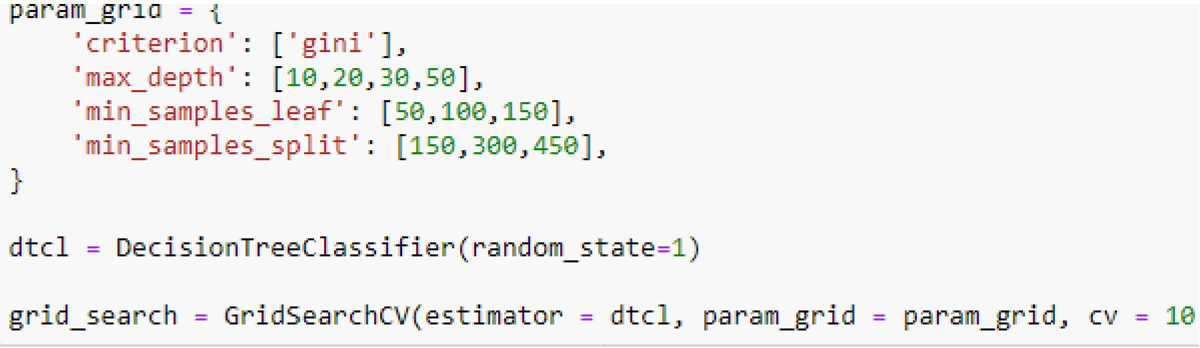
The dimensions of the training and test data are below ; the samples are almost equally distributed between the train and test datasets:



```
X_train (2002, 9)
X_test (859, 9)
train_labels (2002,)
test_labels (859,)
```

Decision Tree Classifier

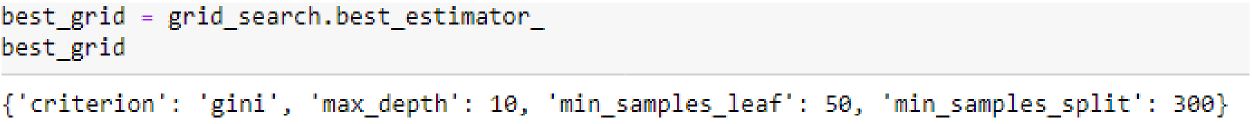
With GINI CRITERION



```
param_grid = {
    'criterion': ['gini'],
    'max_depth': [10,20,30,50],
    'min_samples_leaf': [50,100,150],
    'min_samples_split': [150,300,450],
}

dtcl = DecisionTreeClassifier(random_state=1)

grid_search = GridSearchCV(estimator = dtcl, param_grid = param_grid, cv = 10
```

Grid search is essentially an optimization algorithm which enables us to classify best parameters for the optimization of the problem from a list of parameter

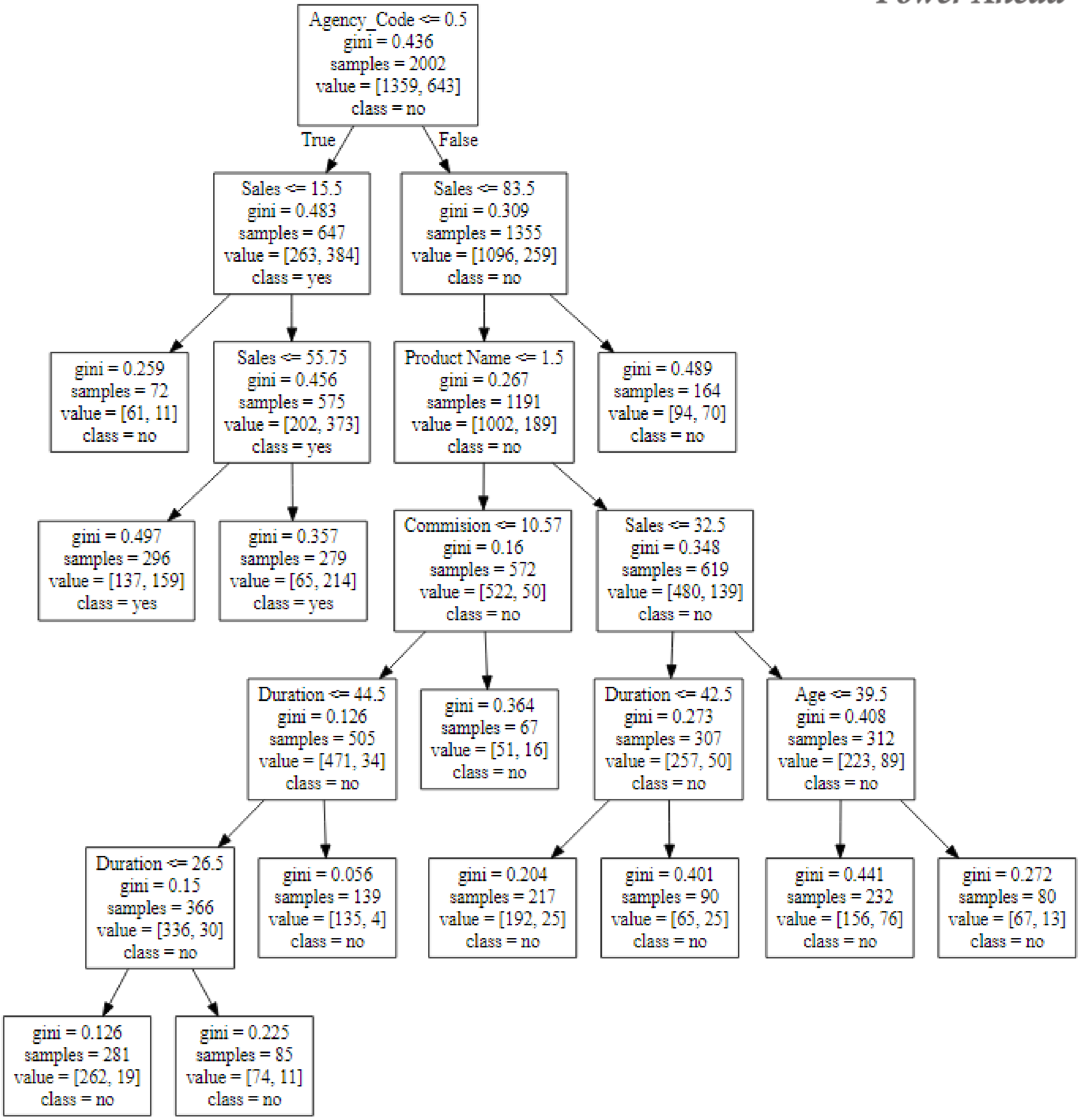
Fitting our train dataset to the grid search. After fitting the values, we will get the best parameters



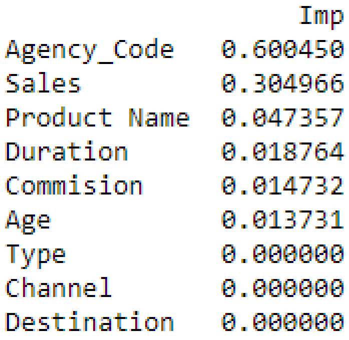
```
best_grid = grid_search.best_estimator_
best_grid
```

```
{'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 50, 'min_samples_split': 300}
```

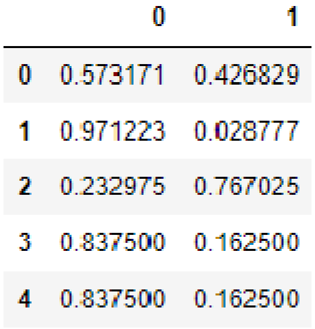
By adding the best grid parameters & saving it on to an output dot file & by pasting the code on http://webgraphviz.com/ to view the tree chart as follows: which will aid in carrying out the test & train data analysis.



Variable Importance & our train and test data with best parameters prediction results for Decision Tree Classifier are displayed below:



```
                   Imp
Agency_Code   0.600450
Sales         0.304966
Product Name  0.047357
Duration      0.018764
Commision     0.014732
Age           0.013731
Type          0.000000
Channel       0.000000
Destination   0.000000
```



	0	1
0	0.573171	0.426829
1	0.971223	0.028777
2	0.232975	0.767025
3	0.837500	0.162500
4	0.837500	0.162500

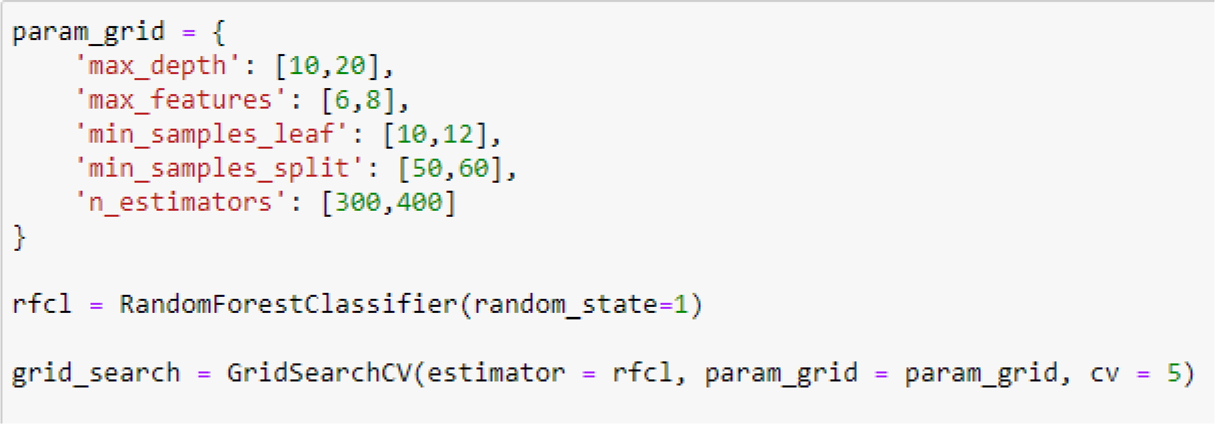
Random Forest Classifier:

The random forest classifier can use for both classification and the regression task

Random forest classifier will handle the missing values.

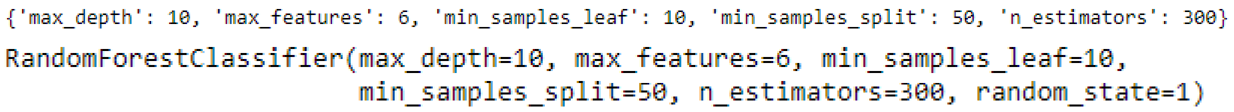
When we have more trees in the forest, random forest classifier won't over fit the model.We can model the random forest classifier for categorical values also.

Now that,Splitting of the data into Train and Test set is already done. RF Model can be built by getting the best parameters.



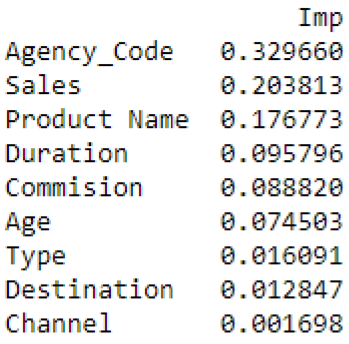
```
param_grid = {
    'max_depth': [10,20],
    'max_features': [6,8],
    'min_samples_leaf': [10,12],
    'min_samples_split': [50,60],
    'n_estimators': [300,400]
}

rfcl = RandomForestClassifier(random_state=1)

grid_search = GridSearchCV(estimator = rfcl, param_grid = param_grid, cv = 5)
```

We get the best parameters & best grid as shown below:

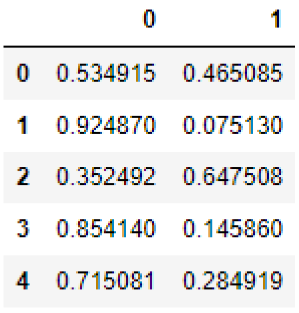


```
{'max_depth': 10, 'max_features': 6, 'min_samples_leaf': 10, 'min_samples_split': 50, 'n_estimators': 300}
RandomForestClassifier(max_depth=10, max_features=6, min_samples_leaf=10,
                       min_samples_split=50, n_estimators=300, random_state=1)
```

Variable Importance & train and test data with best parameters prediction results for Random Forest Classifier are displayed below:

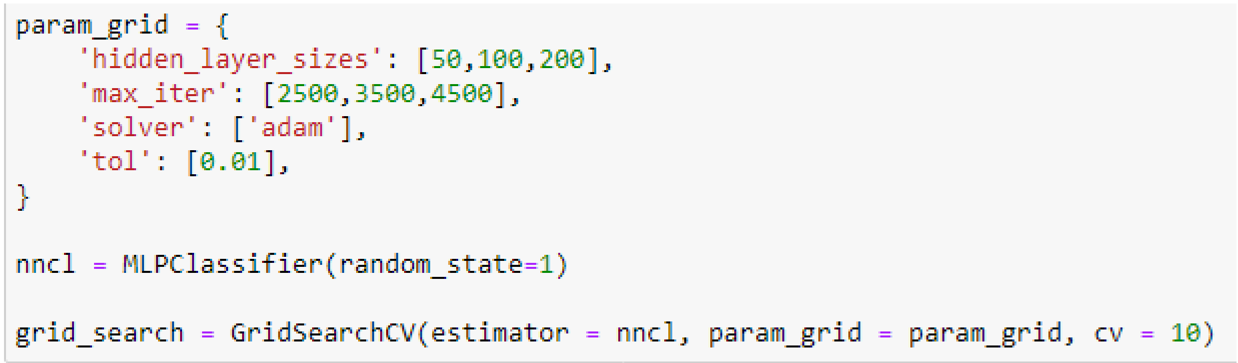


```
                   Imp
Agency_Code   0.329660
Sales         0.203813
Product Name  0.176773
Duration      0.095796
Commision     0.088820
Age           0.074503
Type          0.016091
Destination   0.012847
Channel       0.001698
```



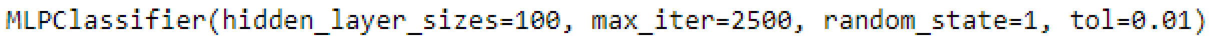
	0	1
0	0.534915	0.465085
1	0.924870	0.075130
2	0.352492	0.647508
3	0.854140	0.145860
4	0.715081	0.284919

Neural Network Classifier:



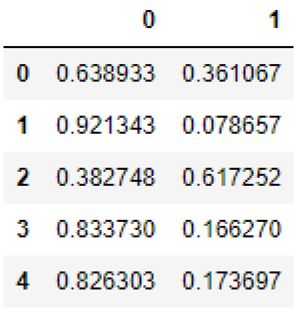
```
param_grid = {
    'hidden_layer_sizes': [50,100,200],
    'max_iter': [2500,3500,4500],
    'solver': ['adam'],
    'tol': [0.01],
}

nncl = MLPClassifier(random_state=1)

grid_search = GridSearchCV(estimator = nncl, param_grid = param_grid, cv = 10)
```

We get the best parameters & best grid as shown below:



```
MLPClassifier(hidden_layer_sizes=100, max_iter=2500, random_state=1, tol=0.01)
```

Train and test data with best parameters prediction results for Neural Network Classifier are displayed below:



	0	1
0	0.638933	0.361067
1	0.921343	0.078657
2	0.382748	0.617252
3	0.833730	0.166270
4	0.826303	0.173697

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

CART MODEL PERFORMANCE EVALUATION

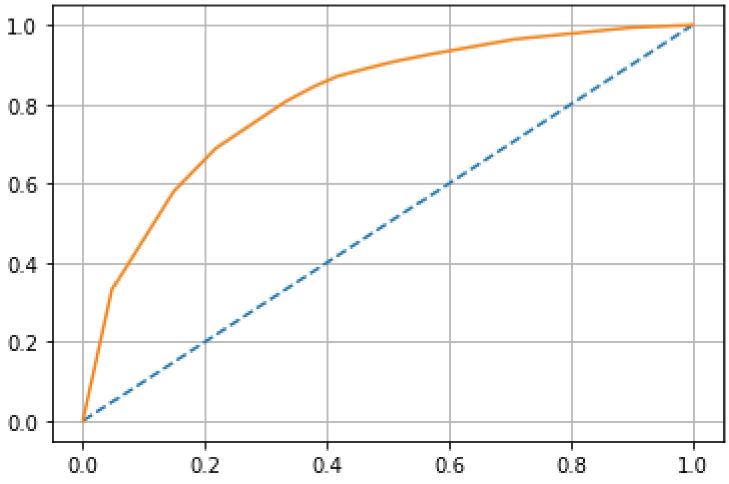
Decision Trees are commonly used in data mining with the objective of creating a model that predicts the value of dependent variable based on the values of several independent variables.

AUC and ROC for the training data

To Predict the probabilities of train data.

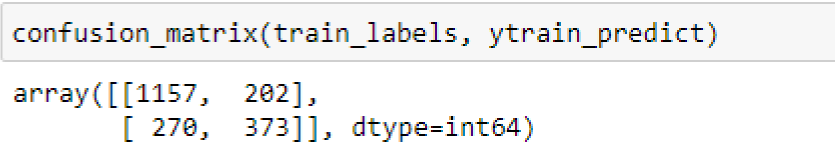
Calculate the AUC as well as the ROC curve and plotting them.

```
AUC: 0.810
```



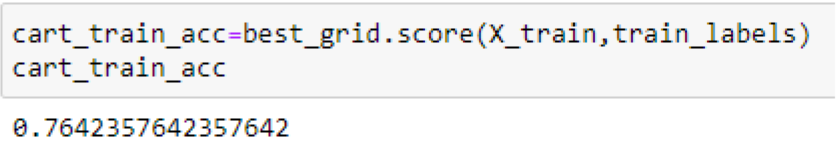
From the Graph it can be be seen that we have derived the AUC Values as 0.810 & ROC Plot for the Train Data.

Confusion Matrix and Accuracy for the training data :



```
confusion_matrix(train_labels, ytrain_predict)
```

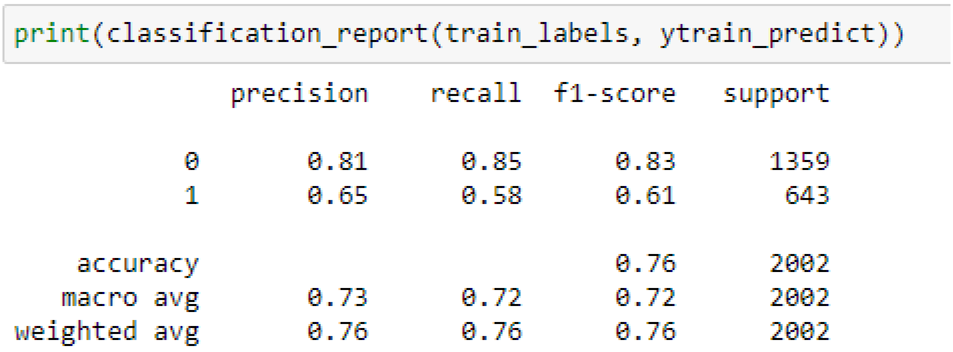
```
array([[1157,  202],
       [ 270,  373]], dtype=int64)
```



```
cart_train_acc=best_grid.score(X_train,train_labels)
cart_train_acc
```

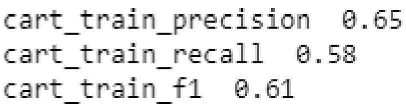
```
0.7642357642357642
```

Classification Report for the training data:



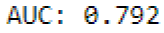
```
print(classification_report(train_labels, ytrain_predict))
```

```
              precision    recall  f1-score   support

           0       0.81      0.85      0.83      1359
           1       0.65      0.58      0.61       643

    accuracy                           0.76      2002
   macro avg       0.73      0.72      0.72      2002
weighted avg       0.76      0.76      0.76      2002
```

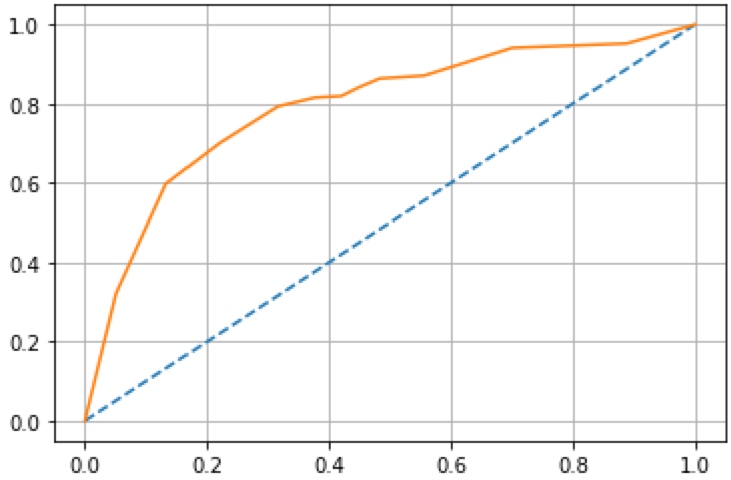


```
cart_train_precision  0.65
cart_train_recall  0.58
cart_train_f1  0.61
```

AUC and ROC for the testing data :

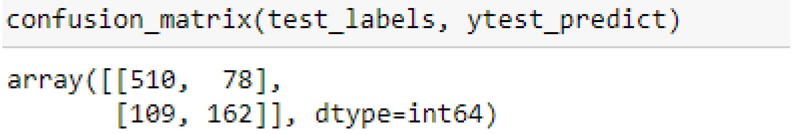


```
AUC: 0.792
```



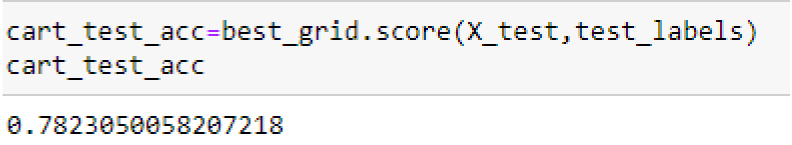
From the Graph it can bee be seen that we have derived the AUC Values as 0.792 & ROC PLot for the Test Data.

Confusion Matrix and Accuracy for the testing data :



```
confusion_matrix(test_labels, ytest_predict)
```

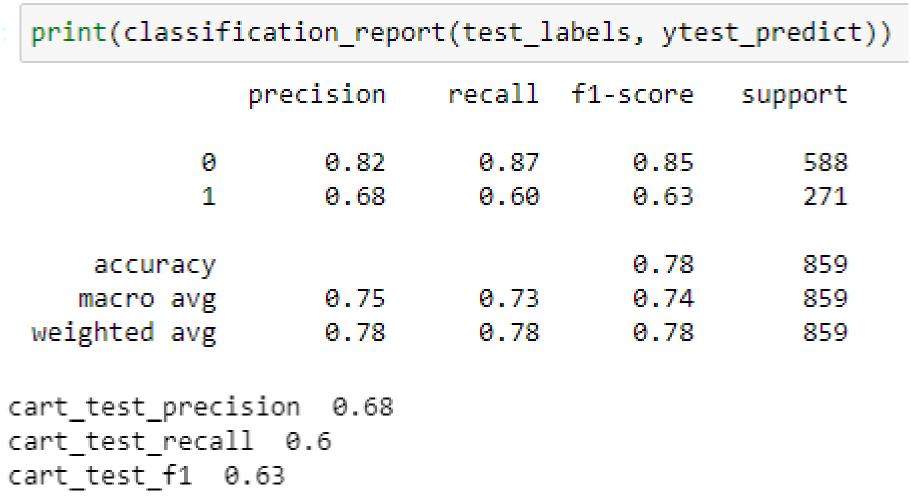
```
array([[510,  78],
       [109, 162]], dtype=int64)
```



```
cart_test_acc=best_grid.score(X_test,test_labels)
cart_test_acc
```

```
0.7823050058207218
```

Classification Report for the testing data:



```
print(classification_report(test_labels, ytest_predict))

              precision    recall  f1-score   support

           0       0.82      0.87      0.85       588
           1       0.68      0.60      0.63       271

    accuracy                           0.78       859
   macro avg       0.75      0.73      0.74       859
weighted avg       0.78      0.78      0.78       859

cart_test_precision  0.68
cart_test_recall  0.6
cart_test_f1  0.63
```

Train Data Accuracy : 76% ; Test Data Accuracy : 78%

Train Data Precision : 65 % ; Test Data Precison : 68%

Train Data f1-score : 61% ; Test Data f1-score: 63%

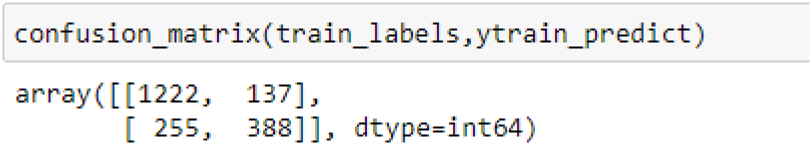
Train Data AUC : 81% ; Test Data AUC : 79%

From observing the characteristics of the CART training & testing data set,we can observe that the results are close & almost similar.Overall,the model is a good model.

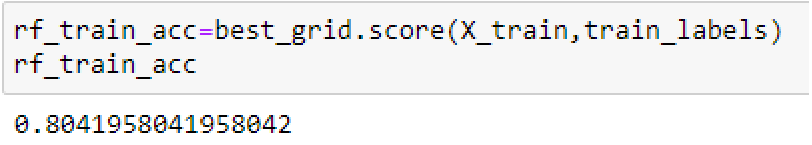
RF MODEL PERFORMANCE EVALUATION

Random forest classifier can handle the missing values. When there are more trees in the forest, random forest classifier won't over fit the model.We Can model the random forest classifier for categorical values also.The model is built with dependant variable as Claimed, and considering all independent variables.

Confusion Matrix and Accuracy for the training data :

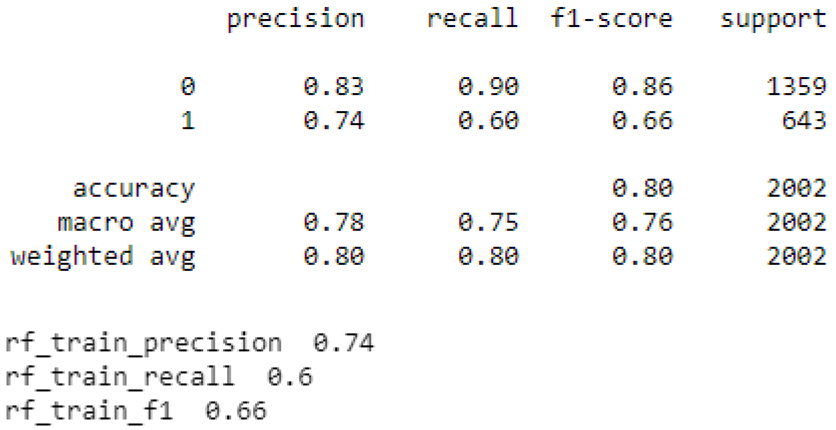


```
confusion_matrix(train_labels,ytrain_predict)

array([[1222,  137],
       [ 255,  388]], dtype=int64)
```



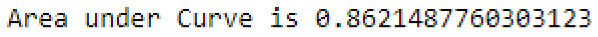
```
rf_train_acc=best_grid.score(X_train,train_labels)
rf_train_acc

0.8041958041958042
```

Classification Report for the training data:

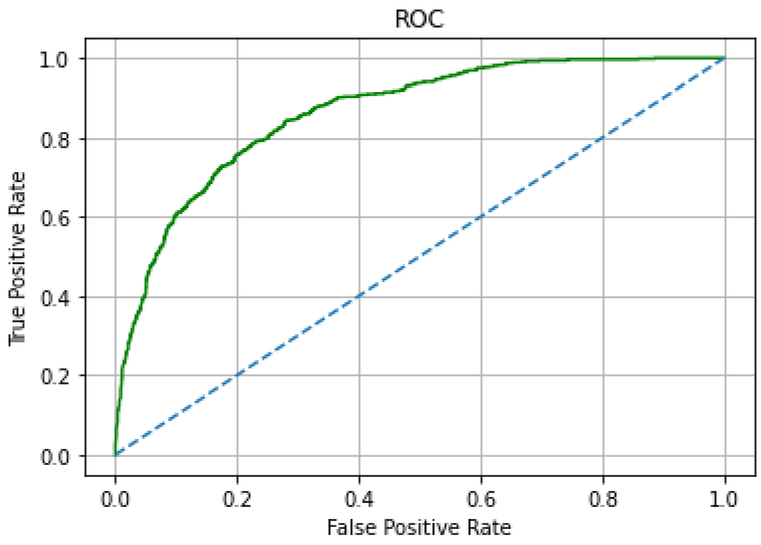


```
              precision    recall  f1-score   support

           0       0.83      0.90      0.86      1359
           1       0.74      0.60      0.66       643

    accuracy                           0.80      2002
   macro avg       0.78      0.75      0.76      2002
weighted avg       0.80      0.80      0.80      2002

rf_train_precision  0.74
rf_train_recall  0.6
rf_train_f1  0.66
```

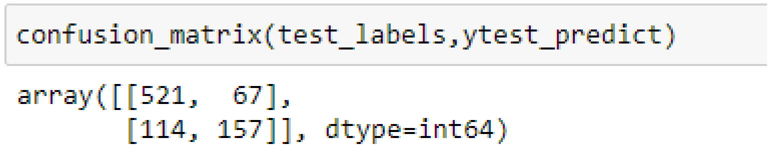
AUC & ROC PLOT FOR TRAINING DATA:



```
Area under Curve is 0.8621487760303123
```

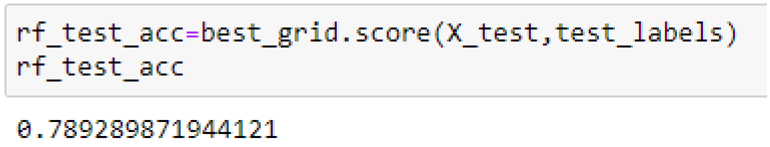


Confusion Matrix and Accuracy for the testing data :



```
confusion_matrix(test_labels,ytest_predict)
```

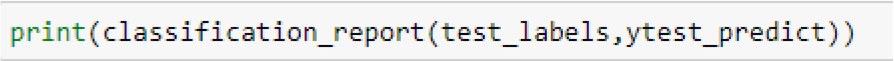
```
array([[521,  67],
       [114, 157]], dtype=int64)
```



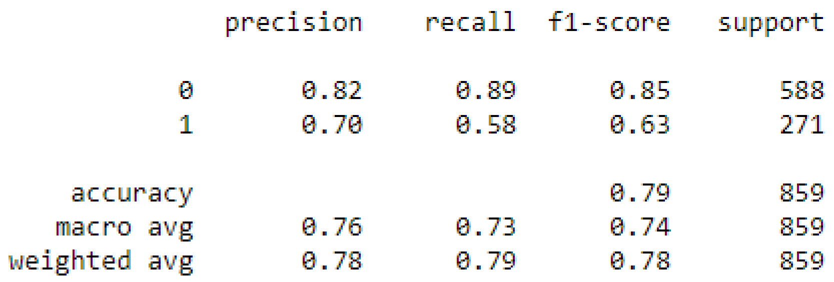
```
rf_test_acc=best_grid.score(X_test,test_labels)
rf_test_acc
```

```
0.789289871944121
```

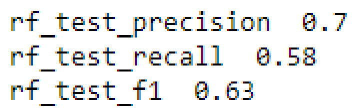
Classification Report for the testing data:



```
print(classification_report(test_labels,ytest_predict))
```

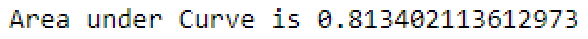


```
              precision    recall  f1-score   support

           0       0.82      0.89      0.85       588
           1       0.70      0.58      0.63       271

    accuracy                           0.79       859
   macro avg       0.76      0.73      0.74       859
weighted avg       0.78      0.79      0.78       859
```

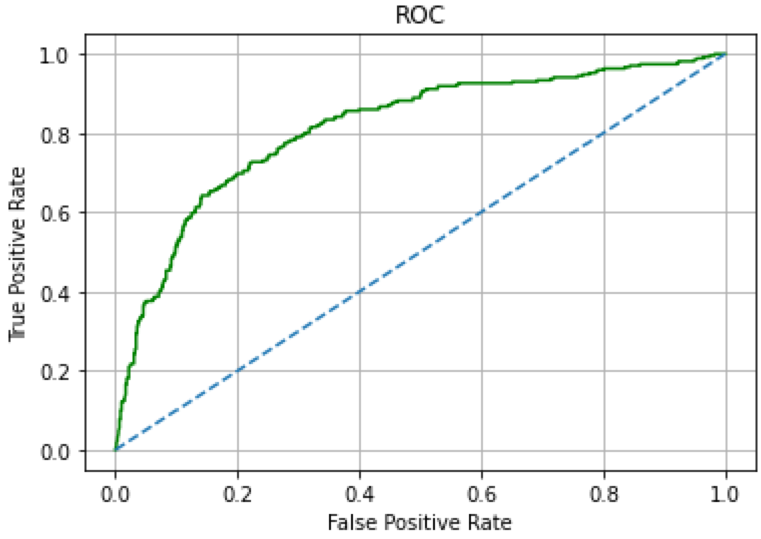


```
rf_test_precision  0.7
rf_test_recall  0.58
rf_test_f1  0.63
```

ROC PLOT FOR TEST DATA



```
Area under Curve is 0.813402113612973
```



Train Data Accuracy : 80% ; Test Data Accuracy : 79%

Train Data Precision : 74 % ; Test Data Precison : 70%

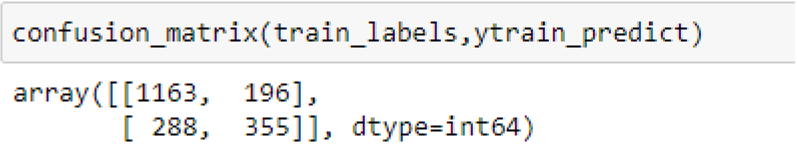
Train Data f1-score : 66% ; Test Data f1-score: 63%

Train Data AUC : 86% ; Test Data AUC : 81%

From observing the characteristics of the RF training & testing data set, RF model has better accuracy,precision,recall & f1 score than the CART model.

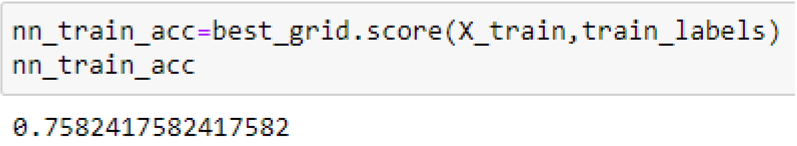
ARTIFICIAL NEURAL NETWORK MODEL PERFORMANCE EVALUATION

Confusion Matrix and Accuracy for the training data :



```
confusion_matrix(train_labels,ytrain_predict)
```

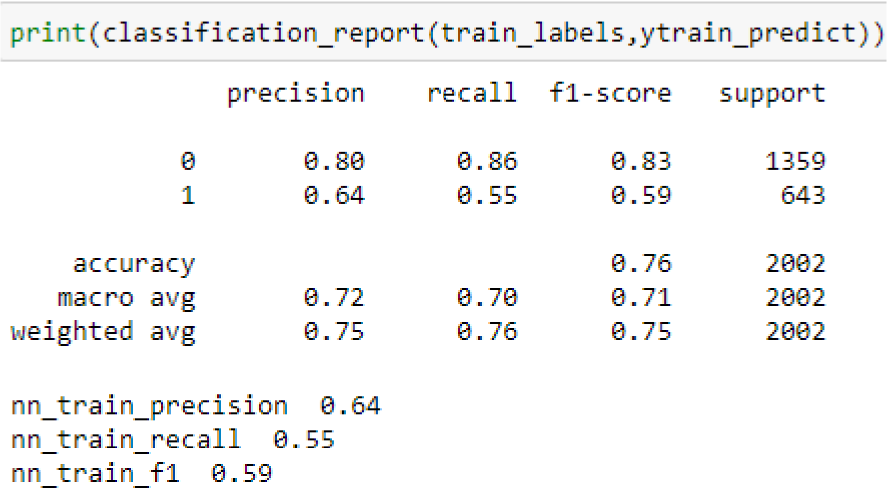
```
array([[1163,  196],
       [ 288,  355]], dtype=int64)
```



```
nn_train_acc=best_grid.score(X_train,train_labels)
nn_train_acc
```

```
0.7582417582417582
```

Classification Report for the training data:

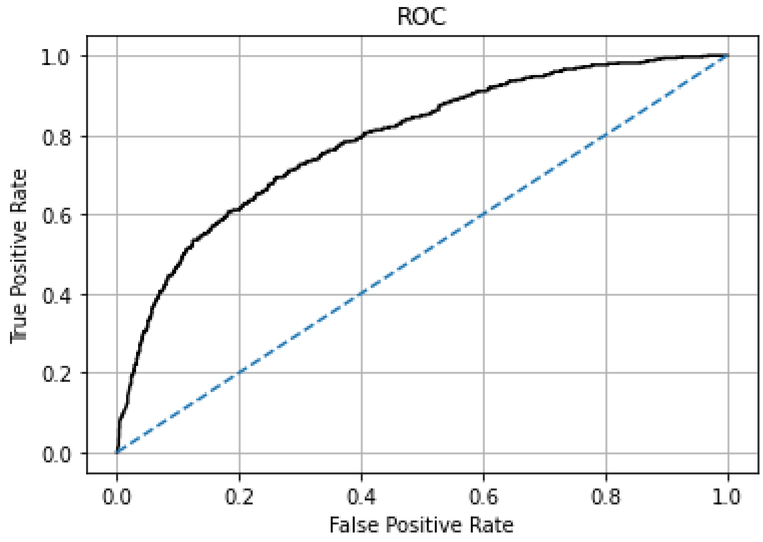


```
print(classification_report(train_labels,ytrain_predict))
```

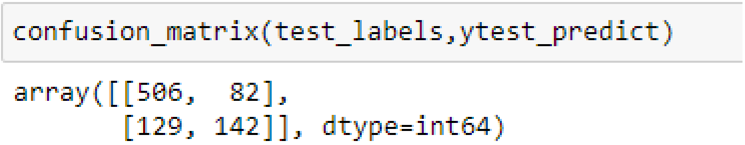
```
              precision    recall  f1-score   support

           0       0.80      0.86      0.83      1359
           1       0.64      0.55      0.59       643

    accuracy                           0.76      2002
   macro avg       0.72      0.70      0.71      2002
weighted avg       0.75      0.76      0.75      2002

nn_train_precision  0.64
nn_train_recall  0.55
nn_train_f1  0.59
```

ROC PLOT FOR TRAINING DATA:

```
Area under Curve is 0.7871576735707002
```

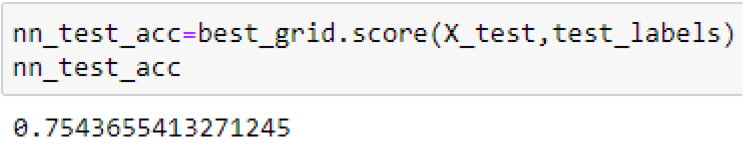


Confusion Matrix and Accuracy for the testing data :



```
confusion_matrix(test_labels,ytest_predict)
```

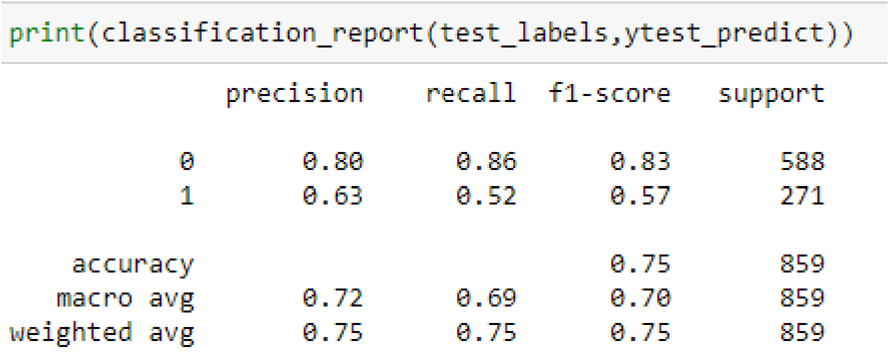
```
array([[506,  82],
       [129, 142]], dtype=int64)
```



```
nn_test_acc=best_grid.score(X_test,test_labels)
nn_test_acc
```

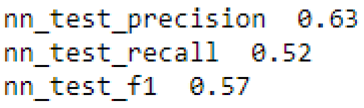
```
0.7543655413271245
```

Classification Report for the testing data:



```
print(classification_report(test_labels,ytest_predict))
```

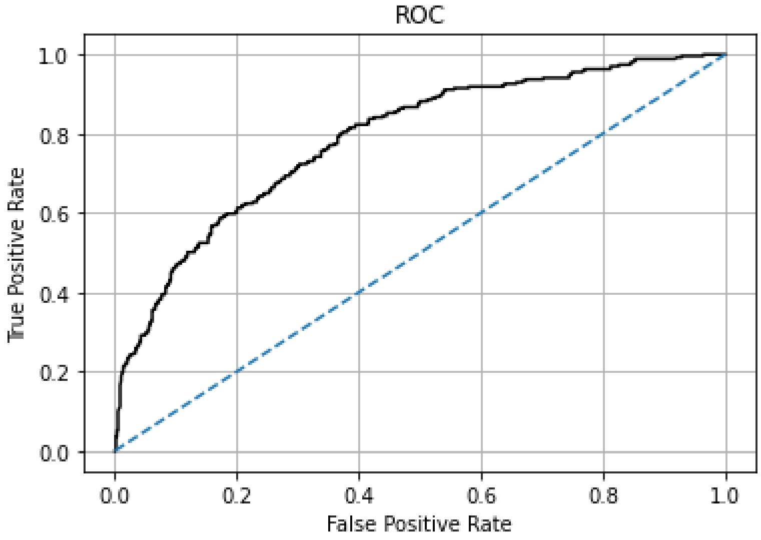
```
              precision    recall  f1-score   support

           0       0.80      0.86      0.83       588
           1       0.63      0.52      0.57       271

    accuracy                           0.75       859
   macro avg       0.72      0.69      0.70       859
weighted avg       0.75      0.75      0.75       859
```



```
nn_test_precision  0.63
nn_test_recall  0.52
nn_test_f1  0.57
```

ROC PLOT FOR TESTING DATA:

```
Area under Curve is 0.7869317468684891
```



Train Data Accuracy : 76% ; Test Data Accuracy : 75%

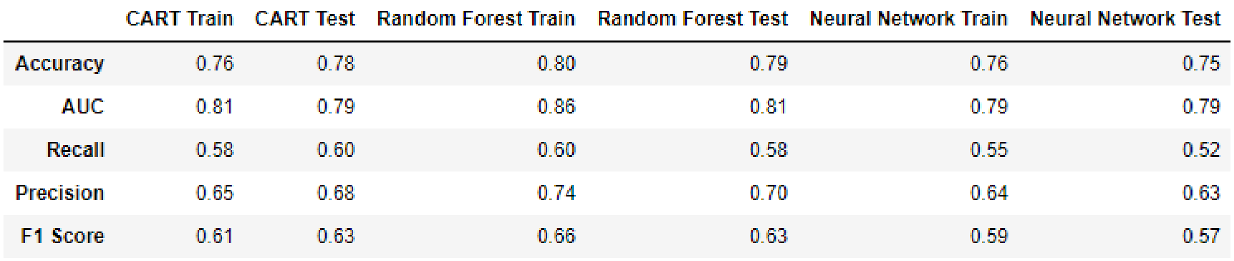
Train Data Precision : 64 % ; Test Data Precison : 63%

Train Data f1-score : 59% ; Test Data f1-score: 57%

Train Data AUC : 79% ; Test Data AUC : 79%

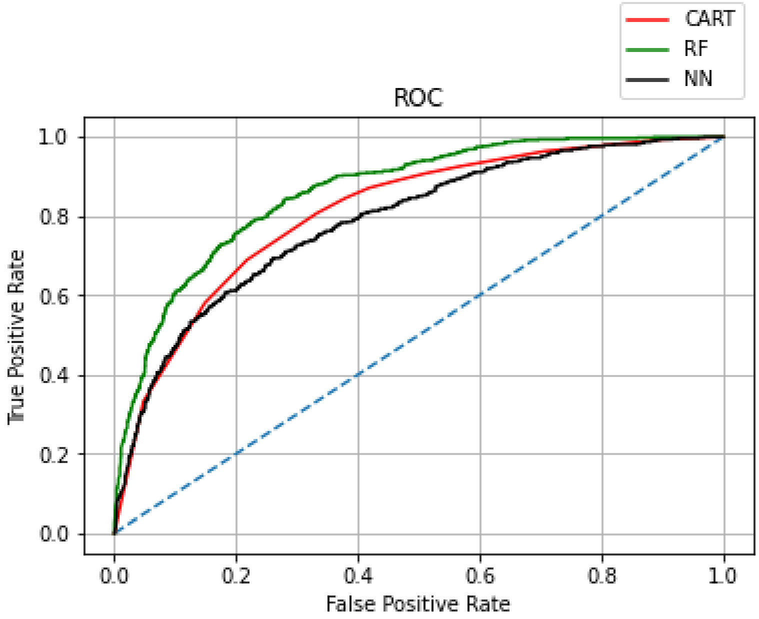
From observing the characteristics of the ANN training & testing data set, RF model has better accuracy,precision,recall & f1 score than the model.So we can finalise on RF Model

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

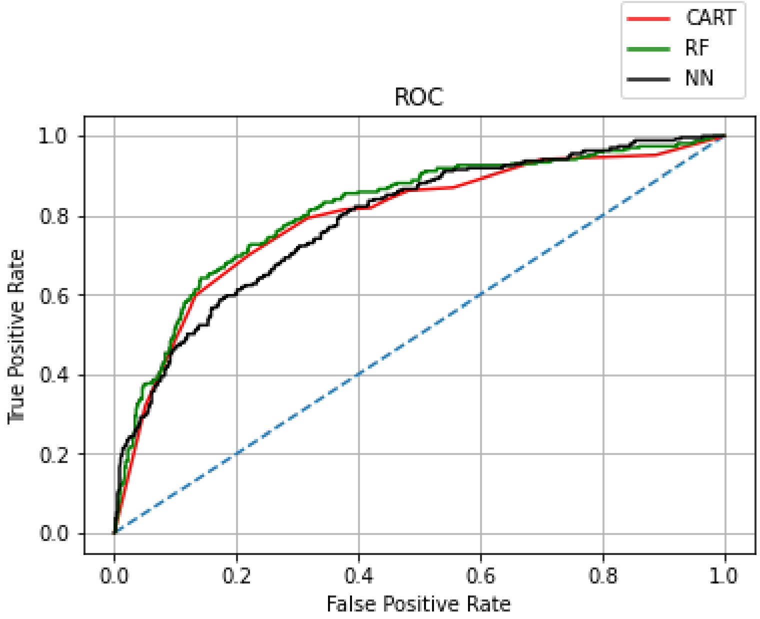


	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.76	0.78	0.80	0.79	0.76	0.75
AUC	0.81	0.79	0.86	0.81	0.79	0.79
Recall	0.58	0.60	0.60	0.58	0.55	0.52
Precision	0.65	0.68	0.74	0.70	0.64	0.63
F1 Score	0.61	0.63	0.66	0.63	0.59	0.57

ROC Curve for all the 3 models (Training data):



ROC Curve for all the 3 models (Testing data):



From observing the characteristics of the CART,RF,ANN training & testing data set, RF model has better accuracy,precision,recall & f1 score than the other two models.

The Random Forest method has the bestperformance I.E best accuracy compared to all the three models. The percentage deviation between Training and Testing Dataset also is reasonably under control, classifying a good model. The Percentage deviation between Training & Testing Dataset is very minimal among the three models.

2.5 Inference: Basis on these predictions, what are the business insights and recommendations.

1. ***The Basic objective for building the predictive model was to see & also classify if an insurance firm providing insurance is facing low/medium/high claim frequency.***
2. ***The data had Outliers.After,performing & running the 3 models we inferred that the data is well balanced for conducting the models but ,more data will assist & help understand to predict the models better.Past Data could also help in understanding & structuring the data to dive into deeper business problems. Overall all the models are stable enough for making any future predictions because there is no overfitting.***
3. ***Claims are higher for ASIA destination ;the management could take actions & follow stipulated protocols before structuring a policy in ASIA.This,could be done by increasing rates via Premium on the policy for recovering the Cost which has been claimed***
4. ***The Management should make more Customers opt for an insurance policy more affordable & increase complexion of the policy in such a way that more customers would choose the insurance policy for the competitive rate.This would attract more customers & also certainly reduce fraud claims by the set of complexions***
5. ***As we could observe from the data maximum of the insurance claim has been done by online channel than offline channel,reason being more convinient & good experience which is a good factor genreating profits.But,subsequently,management should also promote offline channels additional promotions ,offers , low cost than online to pull customers which could generate more customerbase***
6. ***We could observe that most of the Sales was generated from Type Sales than Airlines & also we could observe that Insurance Claim are more at Airlines than Agencies.This could be because of more customer satisfaction & customer trust at Airlines than at Agencies.This could be put into better use by suggestive selling insurance plans at Airlines. Management could deploy better customer service at Agencies to match the likeliness of Airlines which could give better Sales Performance.Low performing Agencies such as JZI,CWT could be given more promotional drives for better sales conversion rates.***
7. ***Customized Plans have been chosen more compared to other plans.Targets & incentives could be given to promote Silver Plans & Gold plans which would generate more sales via Promo offers, Ad Camapaigns ,Marketing drives etc.***
8. ***Management should focus more on Customer Satisfaction,Customer Claim Turn Around Times ,methods to actively reduce false transactions via strong Artificial Intelligence Methods.Fousing on making the process efficient & affordable & reducing overall operational costs.***