A

Report

On

Data Mining

Submitted By –

Ritu Utkarsha

Problem 1: Clustering

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

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

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. Explain the results properly. Interpret and write inferences on the finalized clusters.

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

Dataset for Problem 1: bank_marketing_part1_Data.csv

Data Dictionary for Market Segmentation:

1. spending: Amount spent by the customer per month (in 1000s)
2. advance payments: Amount paid by the customer in advance by cash (in 100s)
3. probability_of_full_payment: Probability of payment done in full by the customer to the bank
4. current_balance: Balance amount left in the account to make purchases (in 1000s)
5. credit_limit: Limit of the amount in credit card (10000s)
6. min_payment_amt: minimum paid by the customer while making payments for purchases made monthly (in 100s)
7. max_spent_in_single_shopping: Maximum amount spent in one purchase (in 1000s)

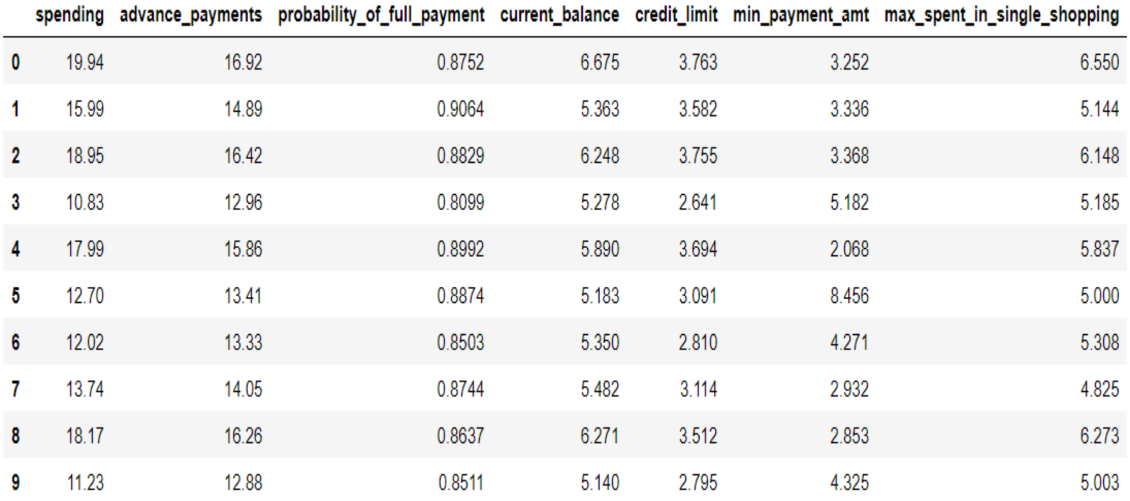
Here, we have to segment the market based on credit card usage, using clustering methods.

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

Solution: EXPLORATORY DATA ANALYSIS:

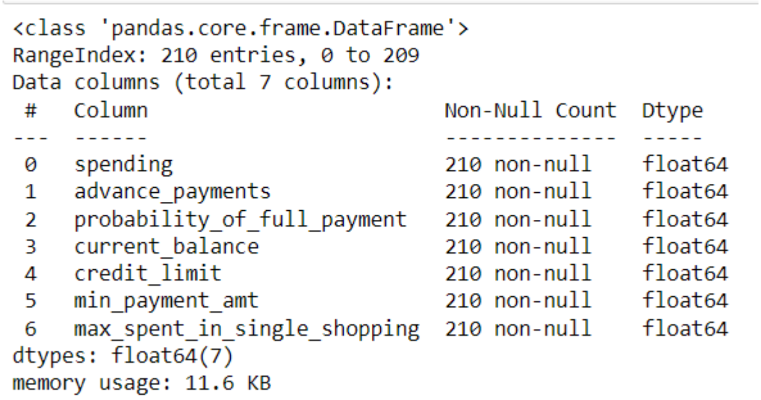
Exploratory Data Analysis (EDA) is an analysis approach that identifies general patterns in the data**. These patterns include outliers and features of the data that might be unexpected. EDA is an important first step in any data analysis.**

Let’s look at the first 10 rows of the dataset.



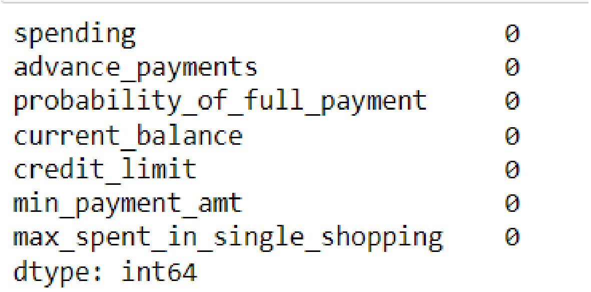
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837
5	12.70	13.41	0.8874	5.183	3.091	8.456	5.000
6	12.02	13.33	0.8503	5.350	2.810	4.271	5.308
7	13.74	14.05	0.8744	5.482	3.114	2.932	4.825
8	18.17	16.26	0.8637	6.271	3.512	2.853	6.273
9	11.23	12.88	0.8511	5.140	2.795	4.325	5.003

CHECKING INFO AND NULL VALUES:



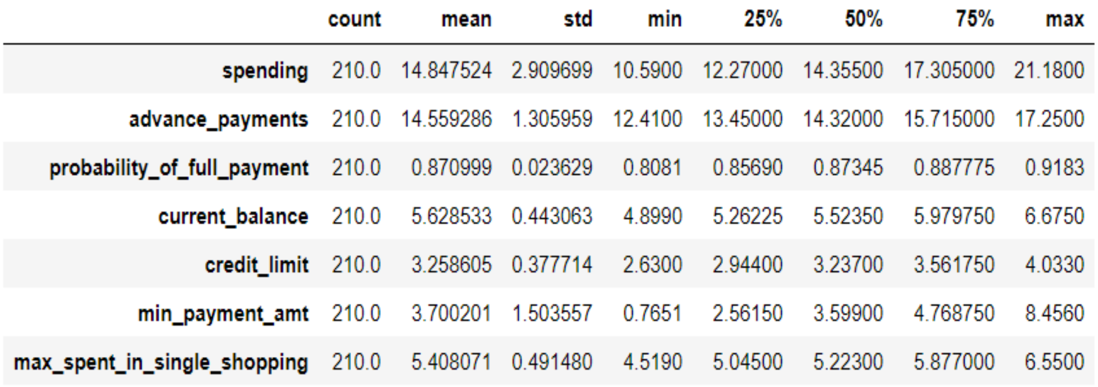
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   spending                      210 non-null    float64
 1   advance_payments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

There are 210 rows and 7 columns in the dataset. Out of 7, all the columns are of float data type.



```
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
```

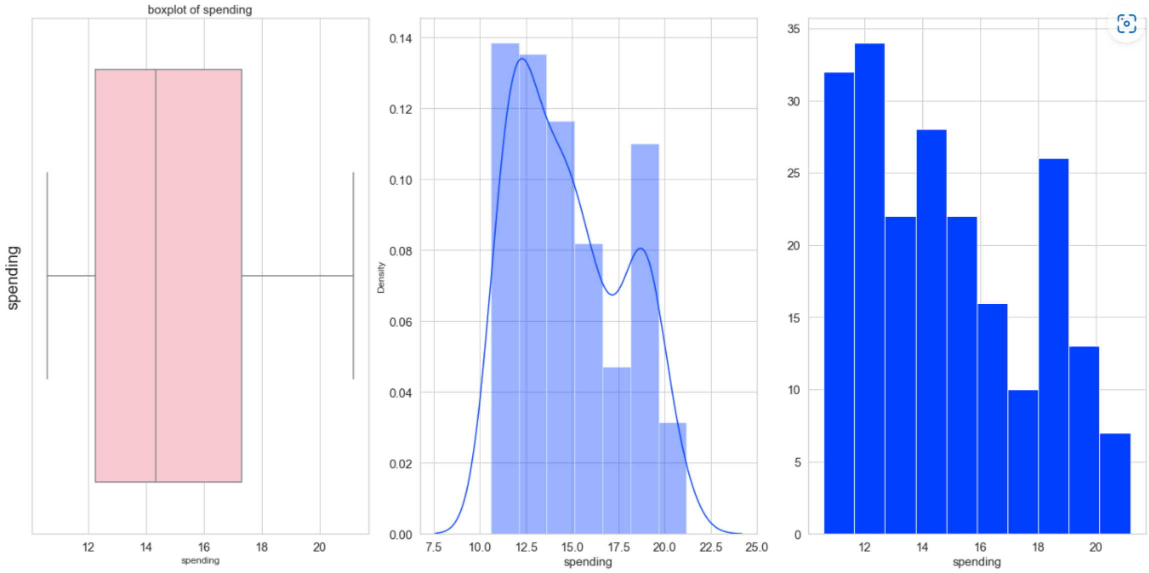
There is no null value and no duplicate in the dataset.



	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

Spending Variable –

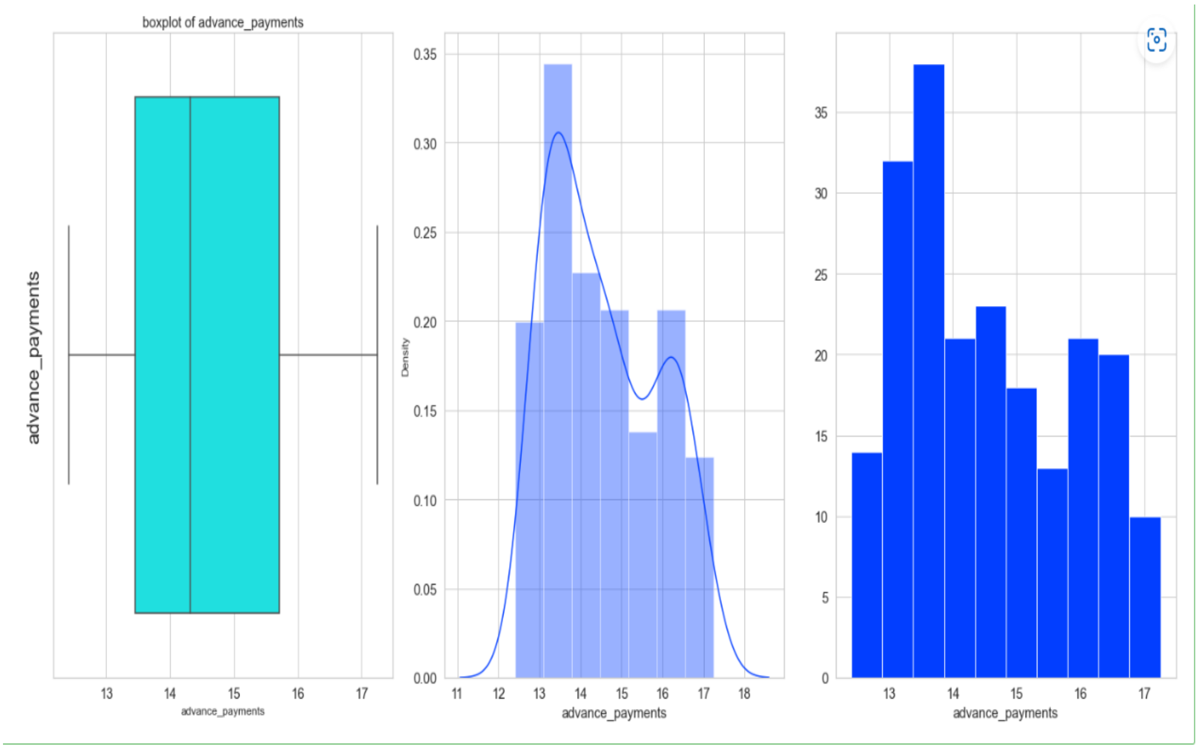
- Range of values: 10.59
- Minimum spending: 10.59
- Maximum spending: 21.18
- Mean value: 14.847523809523818
- Median value: 14.355
- Standard deviation: 2.909699430687361
- Null values: False
- 1st Quartile (Q1) of spending is: 12.27
- 3rd Quartile (Q3) of spending is: 17.305
- Interquartile range (IQR) of spending is 5.035
- Lower outliers in spending: 4.717499999999999
- Upper outliers in spending: 24.8575
- Number of outliers in spending upper: 0
- Number of outliers in spending lower: 0
- % of Outlier in spending upper: 0 %
- % Of Outlier in spending lower: 0 %



Its data is normally distributed, mean and median are almost same. The mean for spending is 14.85 and the median is 14.35. The minimum amount spent by the customers per month in 1000s is 10.59 and the maximum amount spent by the customers per month in 1000s is 21.18. No outlier is detected.

advance_payments variable-

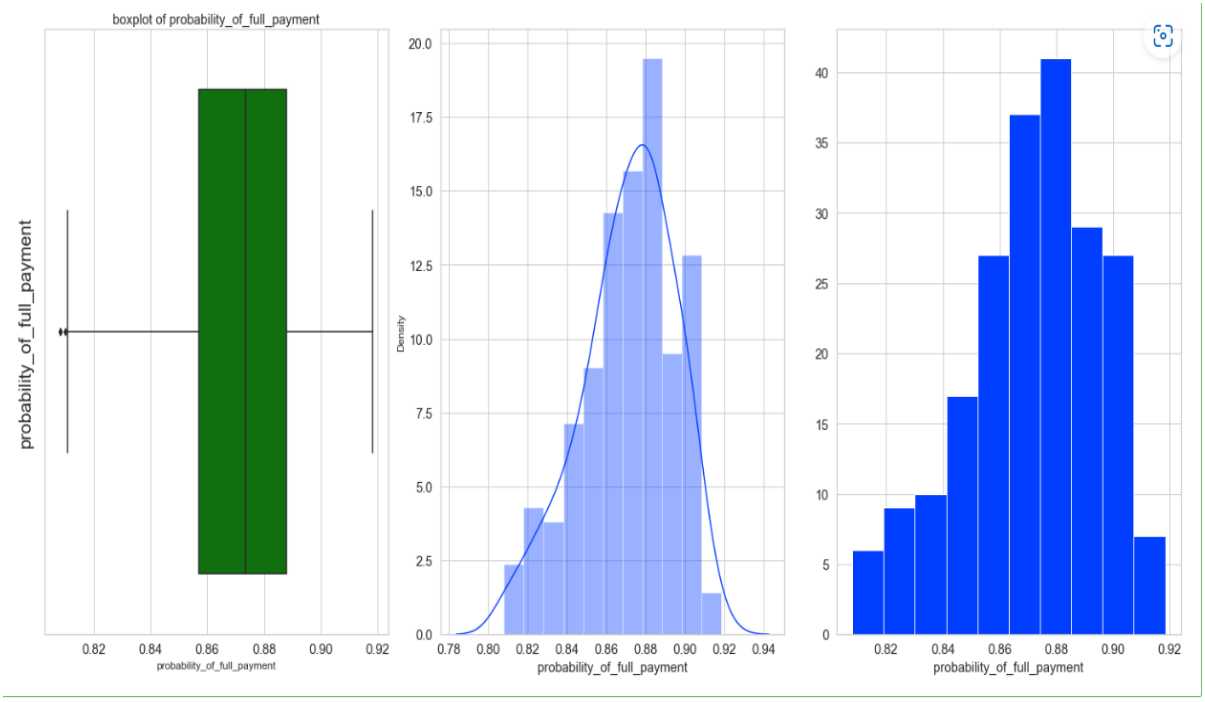
Range of values: 6.66
Minimum advance payments: 12.41
Maximum advance payments: 17.25
Mean value: 14.559285714285727
Median value: 14.32
Standard deviation: 1.305958726564022
Null values: False
1st Quartile (Q1) of advance_payments is: 13.45
3rd Quartile (Q3) of advance_payments is: 15.715
Interquartile range (IQR) of advance_payments is 5.035
Lower outliers in advance_payments: 10.052499999999998
Upper outliers in advance_payments: 19.1125
Number of outliers in advance_payments upper: 0
Number of outliers in advance_payments lower: 0
% Of Outlier in advance_payments upper: 0 %
% Of Outlier in advance_payments lower: 0 %



Data is normally distributed, and the mean and median are almost same. The mean for advance payments is 14.55 and the median is 14.32. The minimum amount paid by the customers in advance in 100s is 12.41 and the maximum amount paid by the customers in advance in 100s is 17.25. No outlier is detected.

probability_of_full_payment –

Range of values: -9.6717
Minimum probability of full payment: 0.8081
Maximum probability of full payment: 0.9183
Mean value: 0.8709985714285714
Median value: 0.8734500000000001
Standard deviation: 0.0236294165838465
Null values: False
1st Quartile (Q1) of probability_of_full_payment is: 0.8569
3rd Quartile (Q3) of probability_of_full_payment is: 0.887775
Interquartile range (IQR) of probability_of_full_payment is 0.03087
Lower outliers in probability_of_full_payment: 0.8105875
Upper outliers in probability_of_full_payment: 0.9340875
Number of outliers in probability_of_full_payment upper: 0
Number of outliers in probability_of_full_payment lower: 3
% Of Outlier in probability_of_full_payment upper: 0 %
% Of Outlier in probability_of_full_payment lower: 1 %

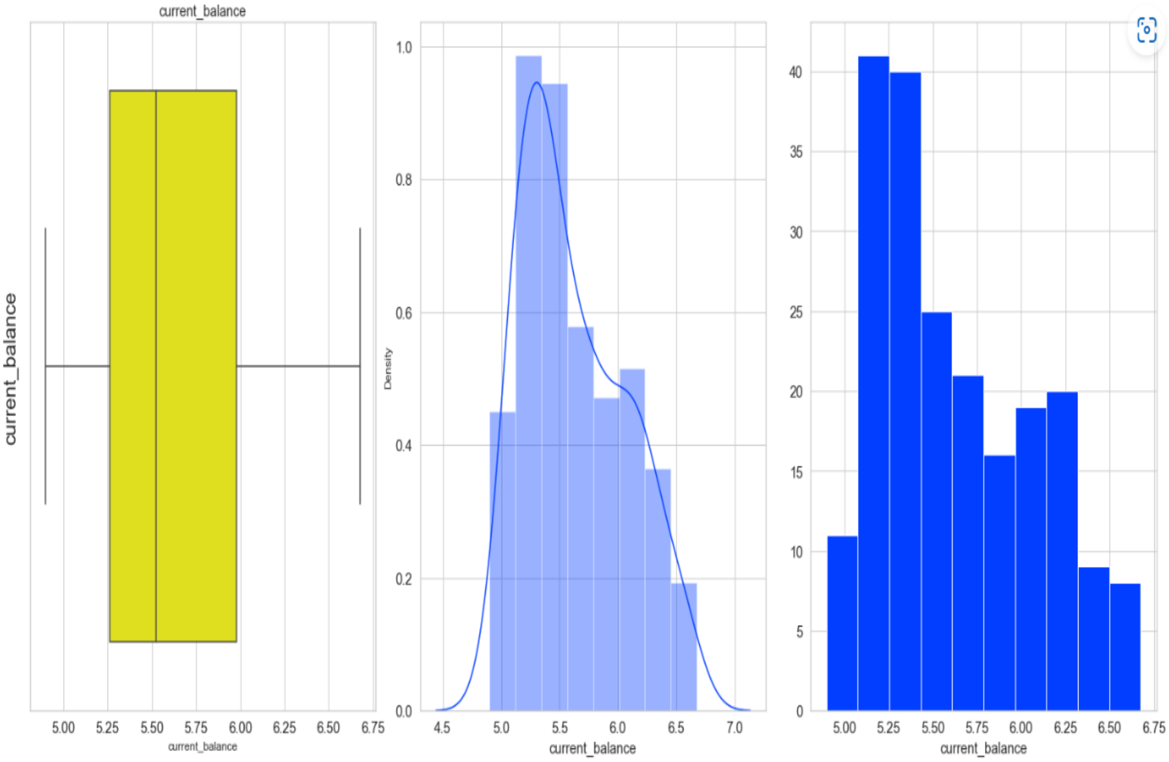


Here median is little bit greater than mean so we cannot say it is fully normally distributed. The mean is 0.870 and the median is 0.8734. It is negatively skewed. The minimum probability of payment done in full by the customer to the bank is 0.80 and the maximum probability of payment done in full by the customer to the bank is 0.91. Outliers are present.

current_balance –

Range of values: 1.7759999999999998
Minimum current_balance: 4.899
Maximum current_balance: 6.675
Mean value: 5.628533333333335
Median value: 5.5235
Standard deviation: 0.44306347772644944
Null values: False
1st Quartile (Q1) of current_balance 5.26225
3rd Quartile (Q3) of current_balance is: 5.97975
Interquartile range (IQR) of is current_balance 0.7175000000000002
Lower outliers in current_balance: 4.186
Upper outliers in current_balance: 7.056000000000001

Number of outliers in current_balance: 0
Number of outliers in current_balance: 0
% Of Outlier in current_balance: 0 %
% Of Outlier in current_balance: 0 %

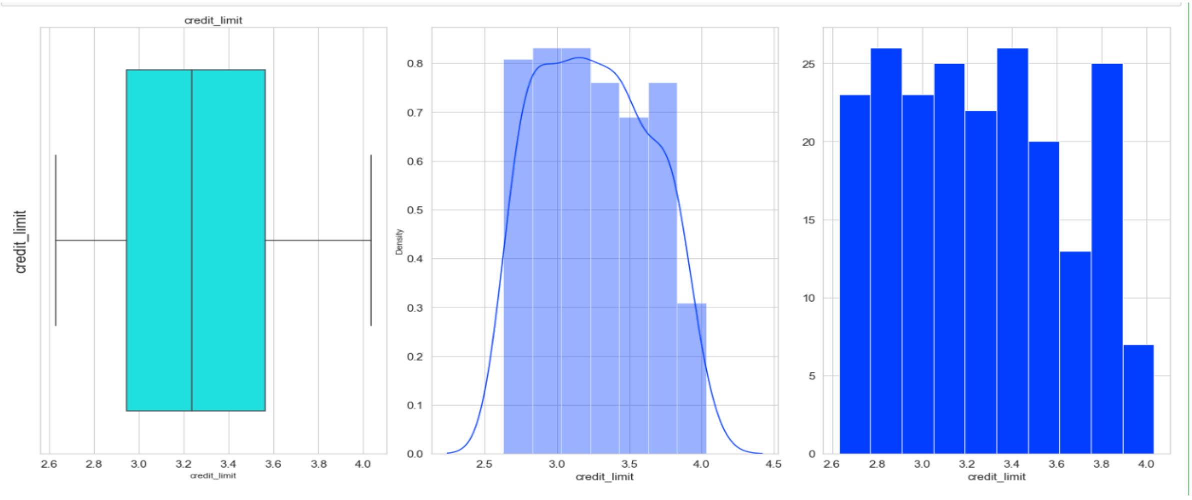


The mean is being little bit greater than median; the data is little bit positively skewed. The mean for current balance is 5.63 and the median is 5.52. The minimum balance amount left in the account to make purchases in 1000s is 4.9 and the maximum amount left in the account to make purchases in 1000s is 6.7. No outlier is detected.

credit_limit-

Range of values: 1.4030000000000005
Minimum credit_limit: 2.63
Maximum credit_limit: 4.033
Mean value: 3.258604761904763
Median value: 3.237
Standard deviation: 0.37771444490658734
Null values: False
1st Quartile (Q1) of credit_limit 2.944
3rd Quartile (Q3) of credit_limit is: 3.56175
Interquartile range (IQR) of is credit_limit 0.61775
Lower outliers in credit_limit: 2.017375
Upper outliers in credit_limit: 4.488375

Number of outliers in credit_limit: 0
Number of outliers in credit_limit: 0
% Of Outlier in credit_limit: 0 %
% Of Outlier in credit_limit: 0 %

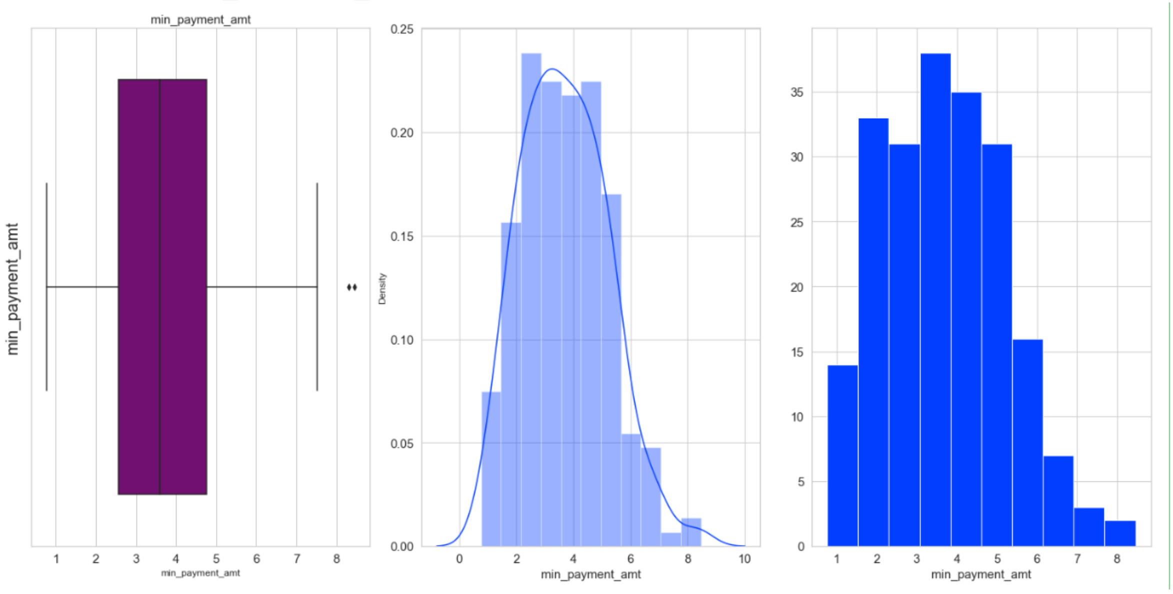


Data is normally distributed, and the mean and median are almost same. The mean of credit limit is 3.26 and the median is 3.24. The minimum limit of amount in credit card 10000s is 2.63 and the maximum limit of amount in credit card in 10000s is 4.0. No outlier is detected.

min_payment_amt-

Range of values: 7.690899999999999
Minimum min_payment_amt: 0.7651
Maximum min_payment_amt: 8.456
Mean value: 3.7002009523809503

Median value: 3.599
Standard deviation: 1.5035571308217792
Null values: False
1st Quartile (Q1) of min_payment_amt 2.5615
3rd Quartile (Q3) of min_payment_amt is: 4.76875
Interquartile range (IQR) of is min_payment_amt 2.2072499999999997
Lower outliers in min_payment_amt -0.7493749999999992
Upper outliers in min_payment_amt 8.079625
Number of outliers in min_payment_amt: 2
Number of outliers in min_payment_amt: 0
% Of Outlier in min_payment_amt: 1 %
% Of Outlier in min_payment_amt: 0 %

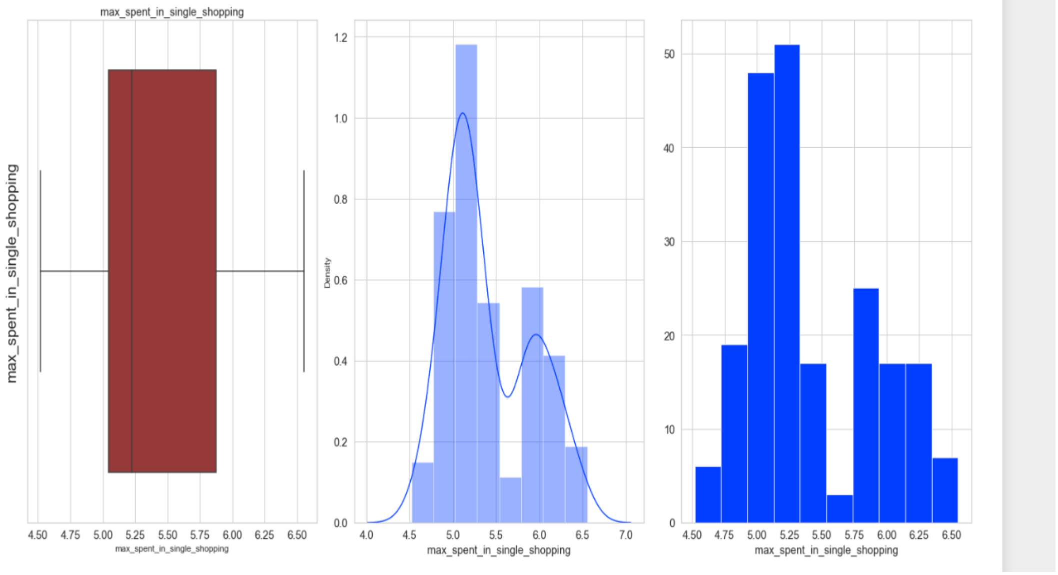


Data is positively skewed, and the mean is little bit greater than median. The mean is 3.7 and the median is 3.6. The minimum paid by the customer while making payments for purchases made monthly (in 100s) ranges from 0.77 to 8.45. The Outliers are detected.

max_spent_in_single_shopping-

Range of values: 2.0309999999999997
Minimum max_spent_in_single_shopping: 4.519
Maximum max_spent_in_single_shopping: 6.55
Mean value: 5.408071428571429
Median value: 5.223000000000001
Standard deviation: 0.49148049910240543
Null values: False
1st Quartile (Q1) of max_spent_in_single_shopping 5.045
3rd Quartile (Q3) of max_spent_in_single_shopping is: 5.877
Interquartile range (IQR) of is max_spent_in_single_shopping 0.8319999
Lower outliers in max_spent_in_single_shopping 3.797

Upper outliers in max_spent_in_single_shopping 7.125
Number of outliers in max_spent_in_single_shopping: 0
Number of outliers in max_spent_in_single_shopping: 0
% Of Outlier in max_spent_in_single_shopping: 0 %
% Of Outlier in max_spent_in_single_shopping: 0 %



Data is positively skewed. The mean is 5.41 and the median is 5.22. The maximum amount spent in single shopping presented ranges from 4.52 to 6.55. No outlier is detected.

SKEWNESS:

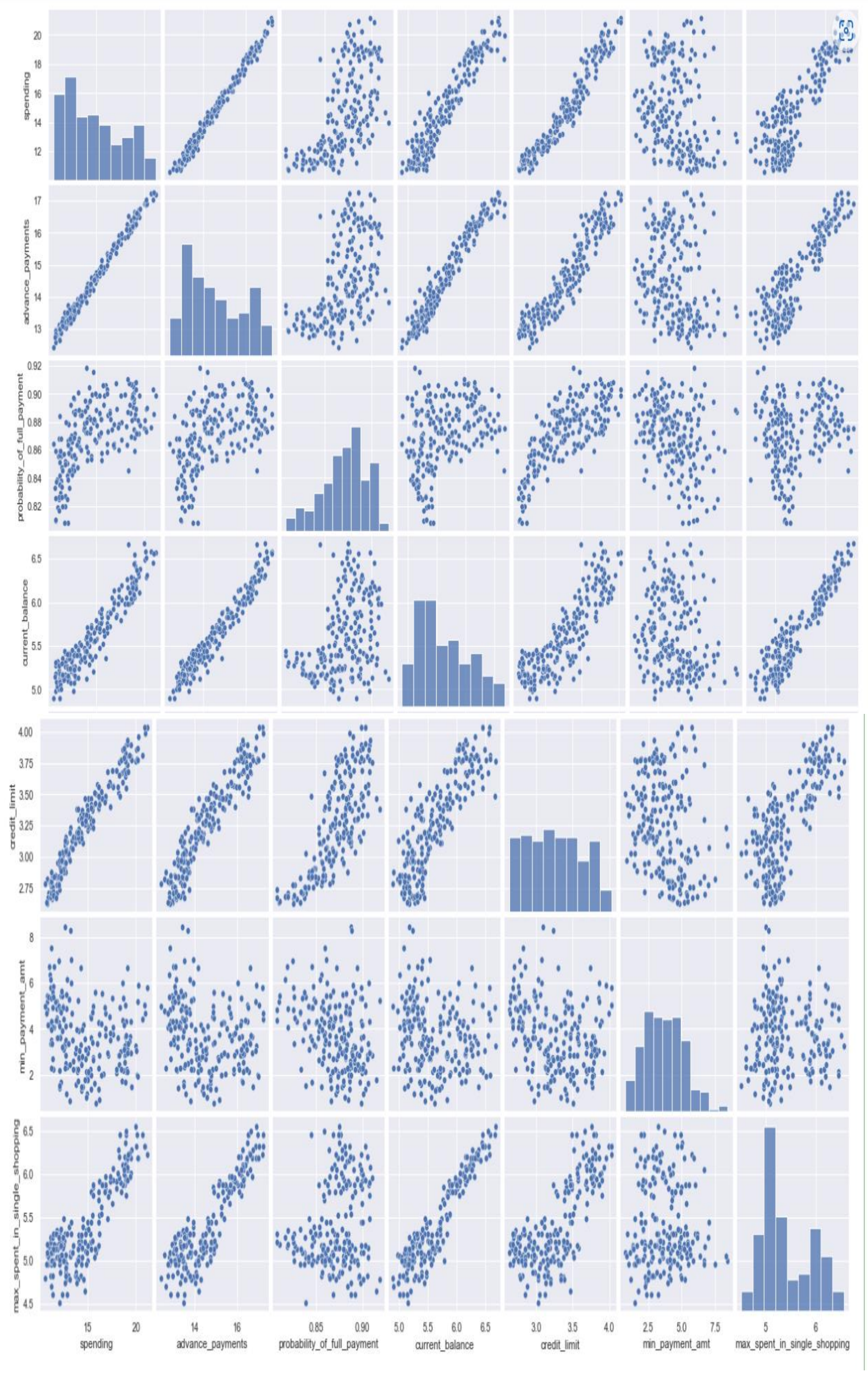
```
max_spent_in_single_shopping    0.561897
current_balance                 0.525482
min_payment_amt                 0.401667
spending                        0.399889
advance_payments                0.386573
credit_limit                    0.134378
probability_of_full_payment    -0.537954
dtype: float64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

MULTIVARIATE ANALYSIS

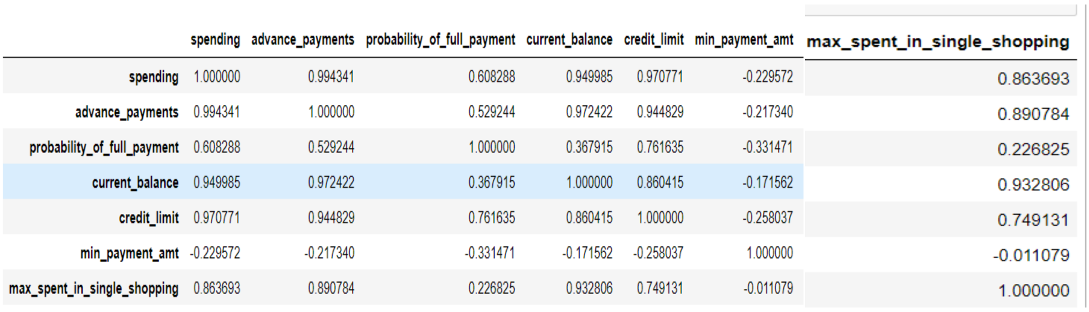
Multivariate analysis (MVA) is a statistical procedure for analysis of data involving more than one type of measurement or observation.

PAIRPLOT: Pair plot shows the relationship between the variables in the form of scatterplot and the distribution between the variables in the form of histogram.

From the pair plot, we can see that there is linear positive relationship between advance_payments and spending, credit_limit and spending, current_balance and advance_payments, credit_limit and advance_payments, current_balance and spending etc.


spending
12
14
16
18
20
advance_payments
13
14
15
16
17
probability_of_full_payment
0.82
0.84
0.86
0.88
0.90
0.92
current_balance
5.0
5.5
6.0
6.5
credit_limit
2.75
3.00
3.25
3.50
3.75
4.00
min_payment_amt
2
4
6
8
max_spent_in_single_shopping
4.5
5.0
5.5
6.0
6.5
15
20
spending
14
16
advance_payments
0.85
0.90
probability_of_full_payment
5.0
5.5
6.0
6.5
current_balance
3.0
3.5
4.0
credit_limit
2.5
5.0
7.5
min_payment_amt
5
6
max_spent_in_single_shopping

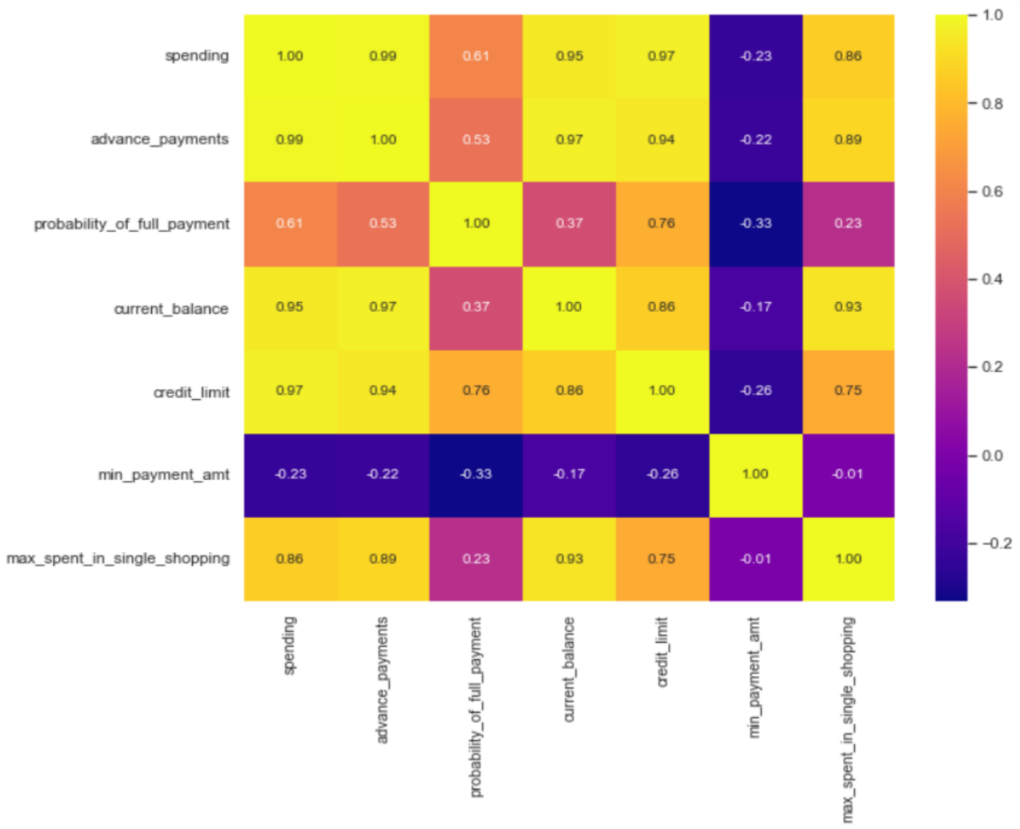
CORRELATION MATRIX:



	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
probability_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
max_spent_in_single_shopping	0.863693	0.890784	0.226825	0.932806	0.749131	-0.011079	1.000000

A correlation is a statistical measure of the relationship between two variables. The measure is best used in variables that demonstrate a linear relationship between each other. The fit of the data can be visually represented in a scatterplot. Using a scatterplot, we can generally assess the relationship between the variables and determine whether they are correlated or not. We can see from the correlation matrix that the various aspects of credit card usage have high positive correlation with each other. The correlation coefficient is a value that indicates the strength of the relationship between variables. The coefficient can take any values from -1 to 1. Values near to 0 have no correlation.

VISUAL REPRESENTATATION USING HEATMAP:



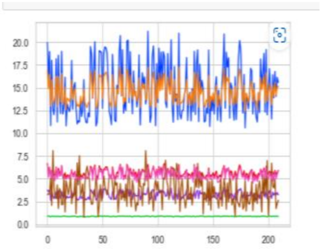
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
spending	1.00	0.99	0.61	0.95	0.97	-0.23	0.86
advance_payments	0.99	1.00	0.53	0.97	0.94	-0.22	0.89
probability_of_full_payment	0.61	0.53	1.00	0.37	0.76	-0.33	0.23
current_balance	0.95	0.97	0.37	1.00	0.86	-0.17	0.93
credit_limit	0.97	0.94	0.76	0.86	1.00	-0.26	0.75
min_payment_amt	-0.23	-0.22	-0.33	-0.17	-0.26	1.00	-0.01
max_spent_in_single_shopping	0.86	0.89	0.23	0.93	0.75	-0.01	1.00

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

Solution:

As the values of the variables are different, we should do scaling otherwise "spending", advance_payments" are in different values and this may get more weightage than the other variables. Scaling will have all the values in the relative same range. Clustering on the non-normalised data fails. Clustering on the normalised data works very well. k-means clustering is very sensitive to scale due to its reliance on Euclidean distance so we should normalize the data. I have used minmax scaling to standardise the data to relative same scale -3 to +3.

Minmax Scaler Transform features by scaling each feature to a given range. This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g., between zero and one.



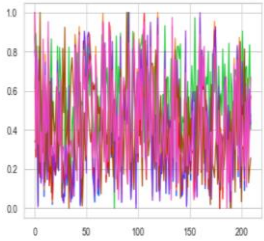
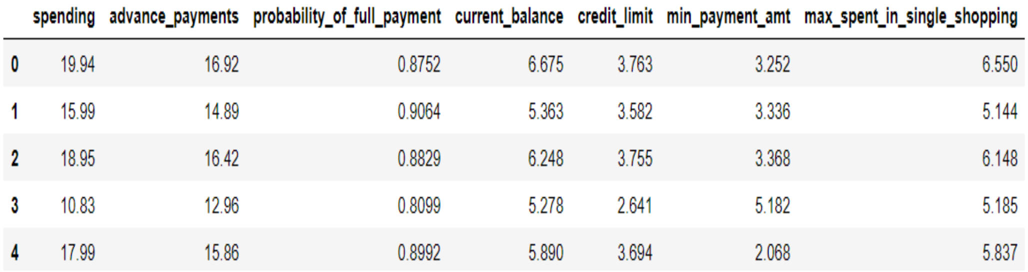
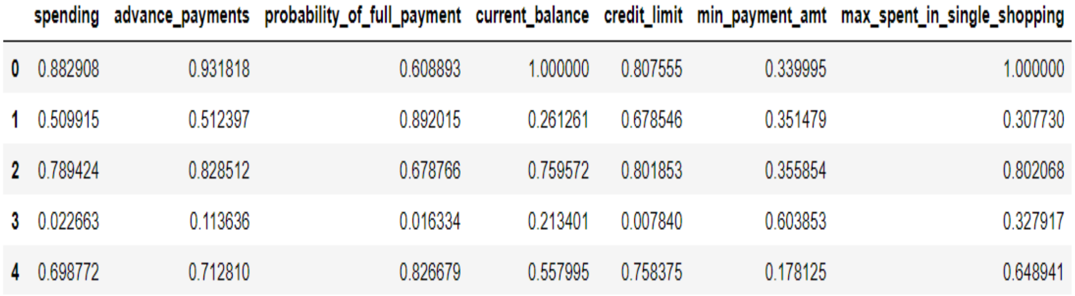


Fig: Before scaling Fig: after scaling



	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



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	0.882908	0.931818	0.608893	1.000000	0.807555	0.339995	1.000000
1	0.509915	0.512397	0.892015	0.261261	0.678546	0.351479	0.307730
2	0.789424	0.828512	0.678766	0.759572	0.801853	0.355854	0.802068
3	0.022663	0.113636	0.016334	0.213401	0.007840	0.603853	0.327917
4	0.698772	0.712810	0.826679	0.557995	0.758375	0.178125	0.648941

Fig: scaled data

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

Solution: Hierarchical clustering is a unsupervised machine learning algorithm, which is used to group the unlabelled datasets into a cluster and is also known as **hierarchical cluster analysis**. In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the **dendrogram**.

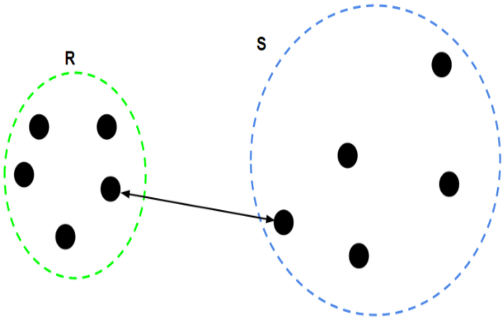
The hierarchical clustering technique has two approaches:

1)**Agglomerative algorithm**: Agglomerative is a bottom-up approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.

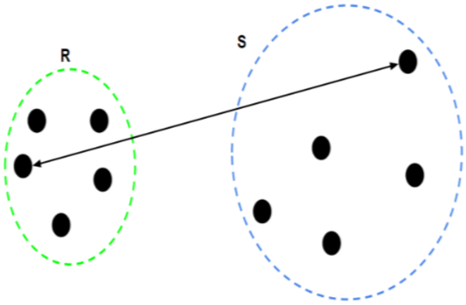
2) Divisive algorithm: It is the reverse of the agglomerative algorithm as it is a top-down approach.

Dendrogram is a tree like diagram that summarizes the process of clustering in a visual format. There are different types of linkages: –

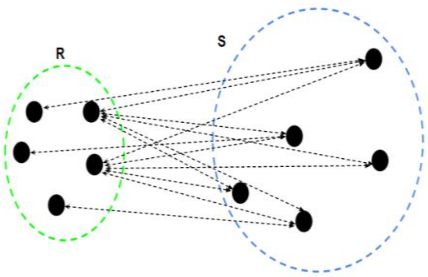
Single Linkage: – In single linkage the distance between the two clusters is the shortest distance between points in those two clusters.



Complete Linkage: – In complete linkage, the distance between the two clusters is the farthest distance between points in those two clusters.



Average Linkage: – In average linkage the distance between the two clusters is the average distance of every point in the cluster with every point in another cluster.



Ward Linkage: - Ward´s linkage is a method for hierarchical cluster analysis . The idea has much in common with analysis of variance (ANOVA). The linkage function specifying the distance between two clusters is computed as the increase in the "error sum of squares" (ESS) after fusing two clusters into a single cluster. Ward´s Method seeks to choose the successive clustering steps to minimize the increase in ESS at each step.

To understand different clusters, visualization is very important. For our dataset, we have used both average and ward linkage with Euclidean distances.

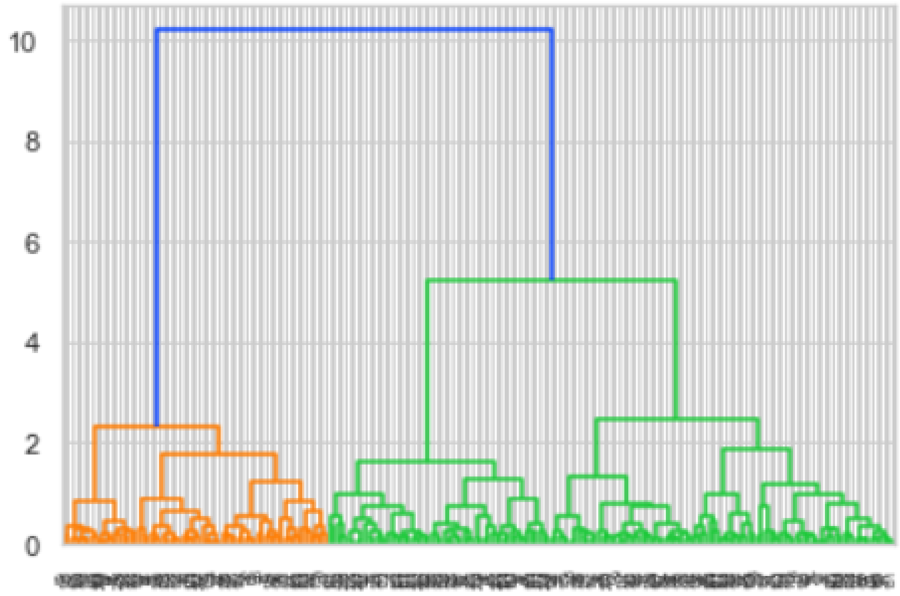


Fig: Dendrogram with ward linkage`

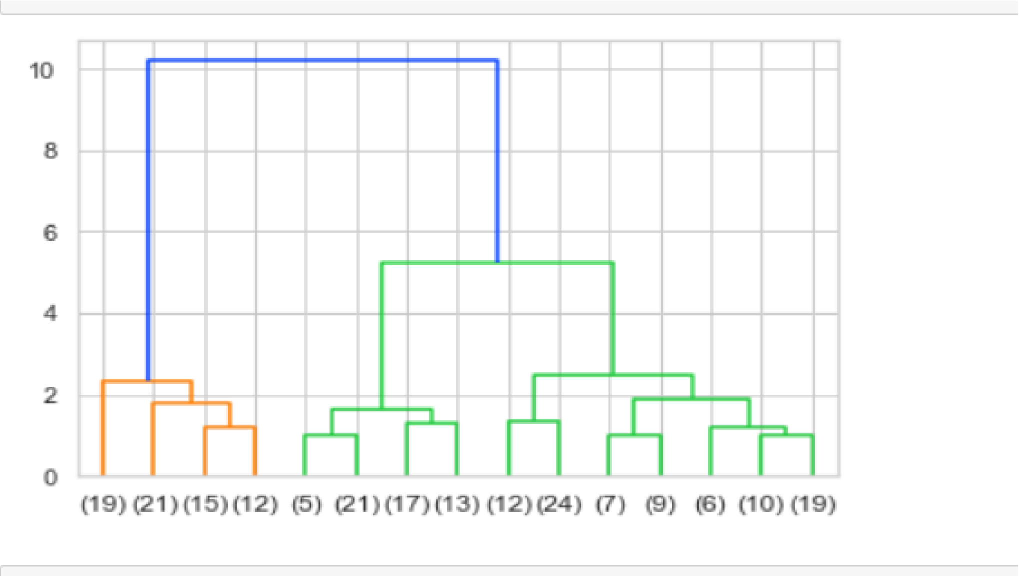


Fig: truncated dendrogram with the ward linkage with the last 15 clusters.

Cut-off 1 (at 4 of y axis) looks to be more suitable for this kind of data since the vertical line that passes through the very first cut-off is of highest length.

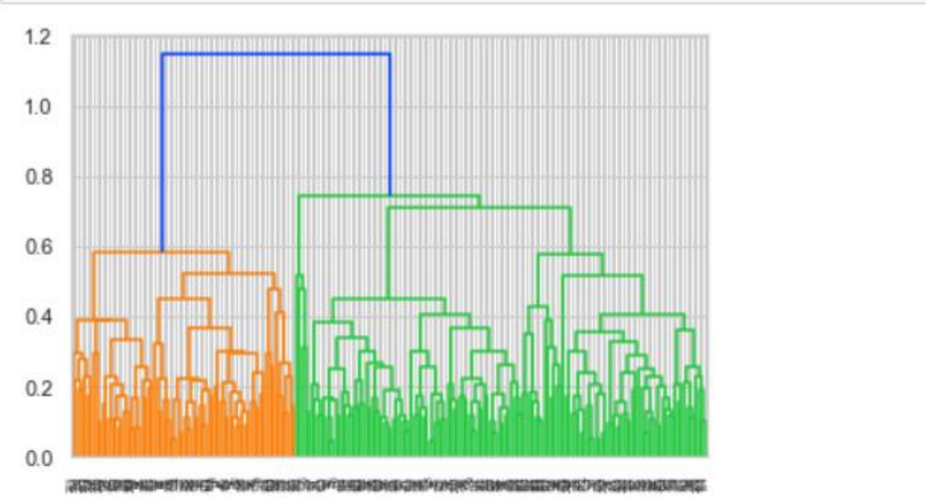


Fig: Dendrogram with average linkage

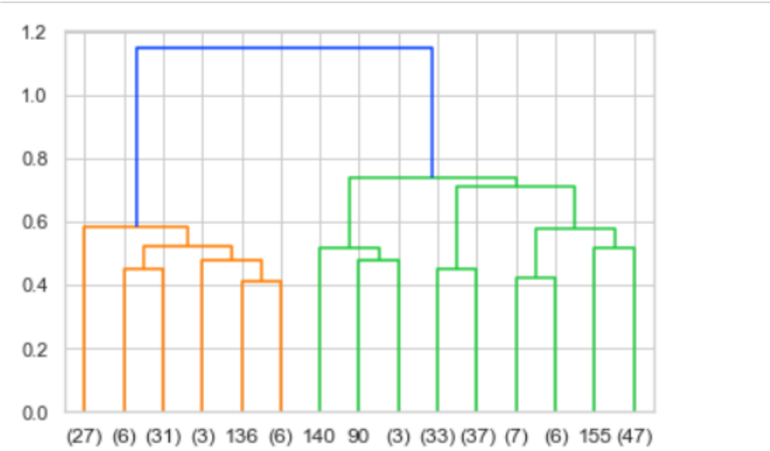
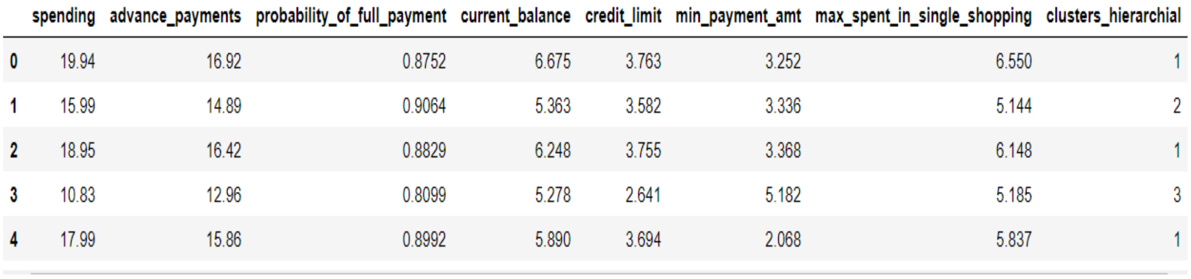


Fig: Truncated dendrogram with the average linkage with the last 15 clusters.

From the figure, we can see that the data is truncated into three clusters by colours (red, blue and green) using both ward method and average linkage method. Since we were not able to see the data points, we truncated the dendrogram.



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

Fig: Final Dataset after merging the clusters

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

Solution:

K-Means clustering is an unsupervised learning algorithm. There is no labeled data for this clustering, unlike in supervised learning. K-Means performs the division of objects into clusters that share similarities and are dissimilar to the objects belonging to another cluster. The term ‘K’ is a number. You need to tell the system how many clusters you need to create. For example, K = 2 refers to two clusters. There is a way of finding out what is the best or optimum value of K for a given data.

There are techniques to find optimum number of k values.

1)**The Elbow method:** In cluster analysis, the **elbow method** is a heuristic used in determining the number of clusters in a data set. The method consists of plotting the explained variation as a function of the number of clusters and picking the elbow of the curve as the number of clusters to use.

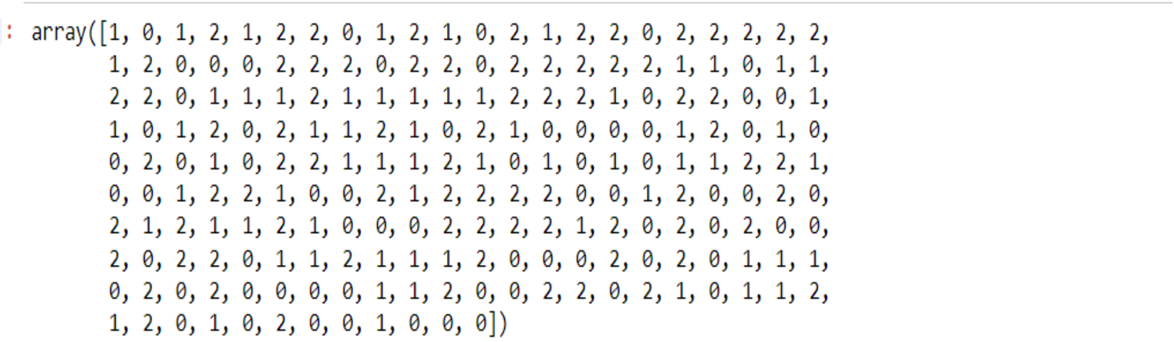
2) **Silhouette Method-**It is used for finding an optimal number of clusters. It is better than the elbow method to find optimal clusters. Silhouette distance ranges from -1 to 1. Higher value within the range -1 to 1 indicates that the object is well matched to its cluster and poorly to the neighbouring cluster. Value closer to 1 is better.

To calculate the average silhouette width for the dataset, we use the below formula:

$$s_{\text{all}} = \frac{1}{N}\sum_{i=1}^{N} s(i).$$

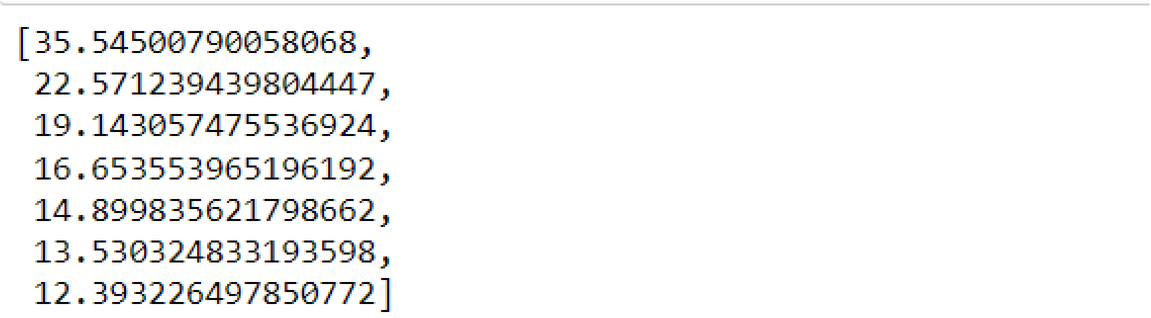
We applied k means clustering on the scaled data followed by calculating WSS scores and then plotted them to check the optimum k value using Elbow method.

We tried applying k means clustering with the number of clusters as 3. Below are the label after applying k means clustering.

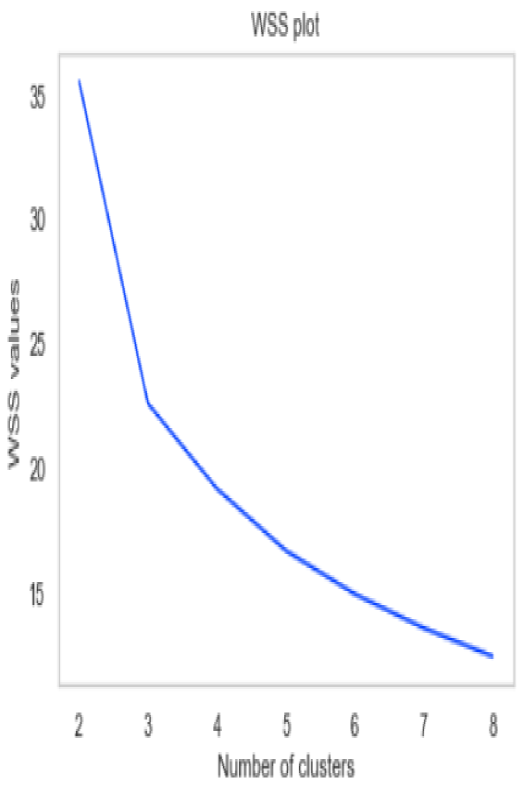


```
: array([1, 0, 1, 2, 1, 2, 2, 0, 1, 2, 1, 0, 2, 1, 2, 2, 0, 2, 2, 2, 2, 2,
         1, 2, 0, 0, 0, 2, 2, 2, 0, 2, 2, 0, 2, 2, 2, 2, 2, 1, 1, 0, 1, 1,
         2, 2, 0, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 0, 2, 2, 0, 0, 1,
         1, 0, 1, 2, 0, 2, 1, 1, 2, 1, 0, 2, 1, 0, 0, 0, 0, 1, 2, 0, 1,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

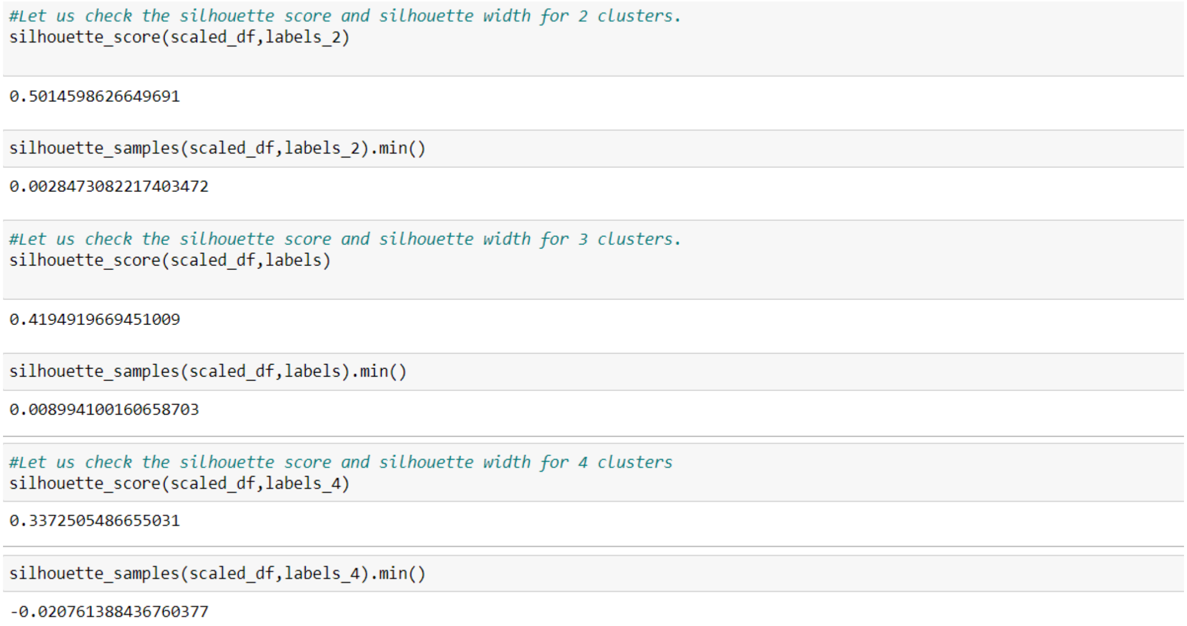
When we try applying k means clustering with k values ranging from 2 to 9, we will get the below inertia/wss score for each cluster.



```
[35.54500790058068,
 22.571239439804447,
 19.143057475536924,
 16.653553965196192,
 14.899835621798662,
 13.530324833193598,
 12.393226497850772]
```

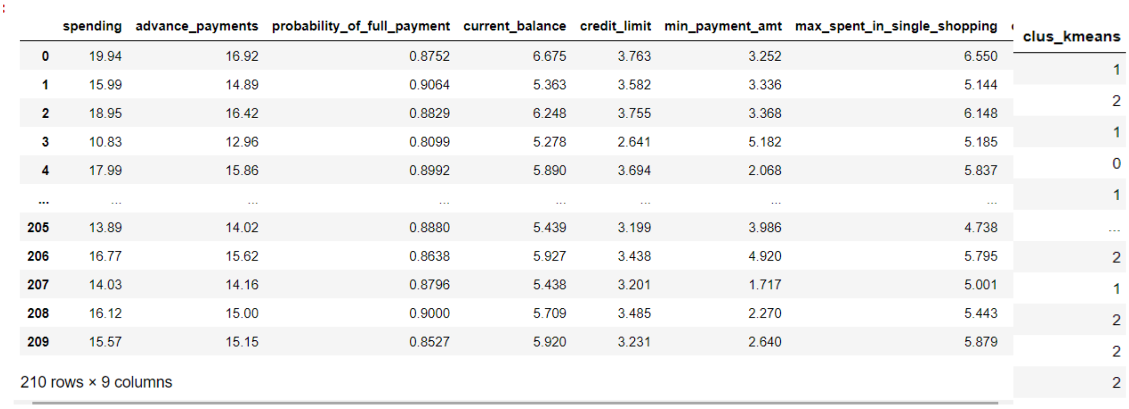


The ideal WSS plot must be a sharp elbow like structure. The number of clusters corresponding to that elbow like graph is the most optimum. Having said that, we'll go for 3 clusters. WSS decreases as value of k keeps increasing.



```
#Let us check the silhouette score and silhouette width for 2 clusters.
silhouette_score(scaled_df,labels_2)

0.5014598626649691

silhouette_samples(scaled_df,labels_2).min()

0.0028473082217403472

#Let us check the silhouette score and silhouette width for 3 clusters.
silhouette_score(scaled_df,labels)

0.4194919669451009

silhouette_samples(scaled_df,labels).min()

0.008994100160658703

#Let us check the silhouette score and silhouette width for 4 clusters
silhouette_score(scaled_df,labels_4)

0.3372505486655031

silhouette_samples(scaled_df,labels_4).min()

-0.020761388436760377
```

Silhouette score for 2 clusters is 0.5 which is closer to 1 than for 3 and 4 clusters (0.41 and 0.33 respectively). But selection of 2 clusters do not give any insights so we can say 3 clusters are well separated from each other on an average.



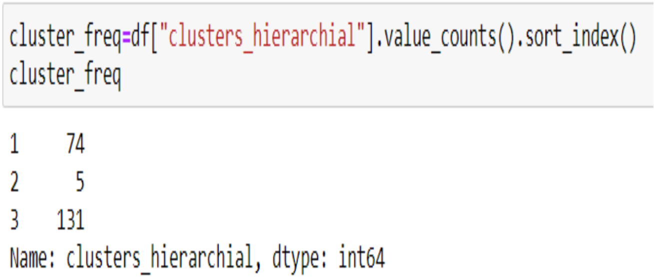
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	clus_kmeans
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	1
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	2
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	0
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837	1
...	...	...	...	...	...	...	...	...
205	13.89	14.02	0.8880	5.439	3.199	3.986	4.738	2
206	16.77	15.62	0.8638	5.927	3.438	4.920	5.795	1
207	14.03	14.16	0.8796	5.438	3.201	1.717	5.001	2
208	16.12	15.00	0.9000	5.709	3.485	2.270	5.443	2
209	15.57	15.15	0.8527	5.920	3.231	2.640	5.879	2

210 rows × 9 columns

Fig: k means clusters merged with the original data frame

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

Solution:



```
cluster_freq=df["clusters_hierarchial"].value_counts().sort_index()
cluster_freq
```

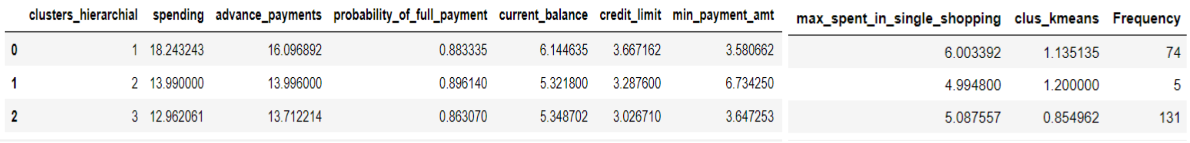
```
1     74
2      5
3    131
Name: clusters_hierarchial, dtype: int64
```

Fig: Cluster frequency

Cluster 0(74 customers): Customers under cluster 0 have high spending, current balance, credit limit, current balance, maximum spend in single shopping, probability of full payment. This indicates they are premium high net worth customers who make expensive purchases on their credit card.

Cluster 1(5 customers): Customers here have comparatively lesser spending, current balance, credit limit, current balance, maximum spend in single shopping, probability of full payment. They can be upper middle-class customers. The bank can help them with offers so that they can increase their spending and make a transition into premium customers.

Cluster 2(131 customers): They have the least spending and credit limit. It indicates they have recently bought credit cards, or they have started working recently. They can increase their spending habits by tying up with grocery stores, utilities etc.



	clusters_hierarchial	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	clus_kmeans	Frequency
0	1	18.243243	16.096892	0.883335	6.144635	3.667162	3.580662	6.003392	1.135135	74
1	2	13.990000	13.996000	0.896140	5.321800	3.287600	6.734250	4.994800	1.200000	5
2	3	12.962061	13.712214	0.863070	5.348702	3.026710	3.647253	5.087557	0.854962	131

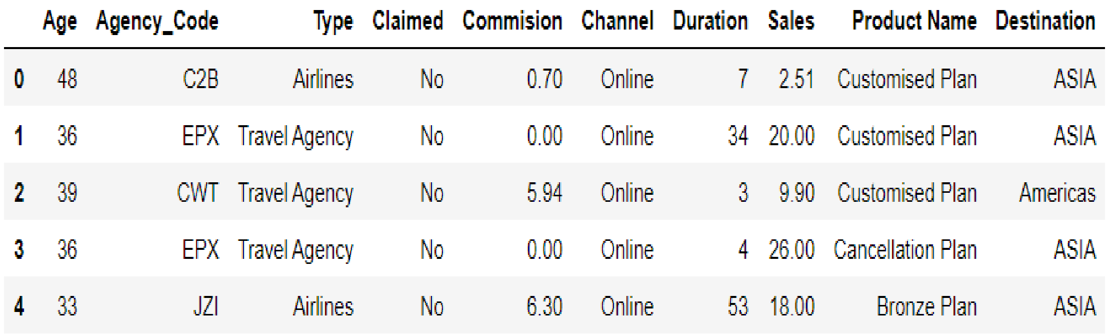
Problem 2: CART-RF

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 and compare the models' performances in train and test sets.

2.1 Read the data and do exploratory data analysis (4 pts). Describe the data briefly. Interpret the inferences for each (2 pts). Initial steps like head () .info(), Data Types, etc . Null value check. Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Summary stats,

Skewness, Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this, but the code should be able to represent the correct output and inferences should be logical and correct.

Solution: Lets see the head of the data.

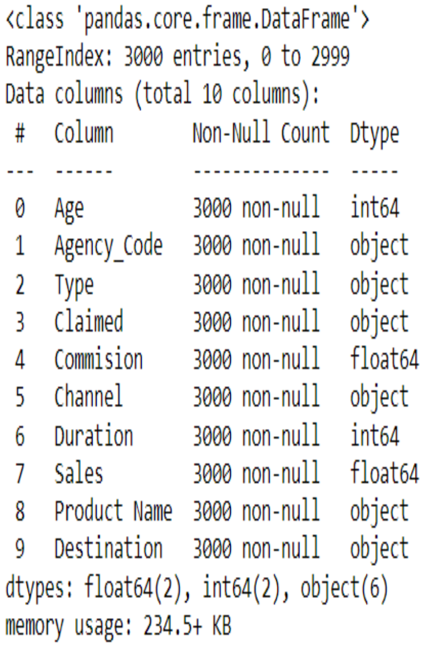


	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	C2B	Airlines	No	0.70	Online	7	2.51	Customised Plan	ASIA
1	36	EPX	Travel Agency	No	0.00	Online	34	20.00	Customised Plan	ASIA
2	39	CWT	Travel Agency	No	5.94	Online	3	9.90	Customised Plan	Americas
3	36	EPX	Travel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

3000 rows and 10 columns

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
```

Numerical columns:

Age, Commission, Duration, Sales

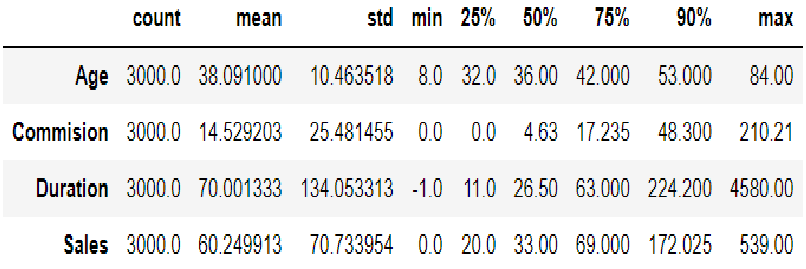
Categorical columns:

Agency code, type, claimed, channel, product name, destination

Null values:

0 null values

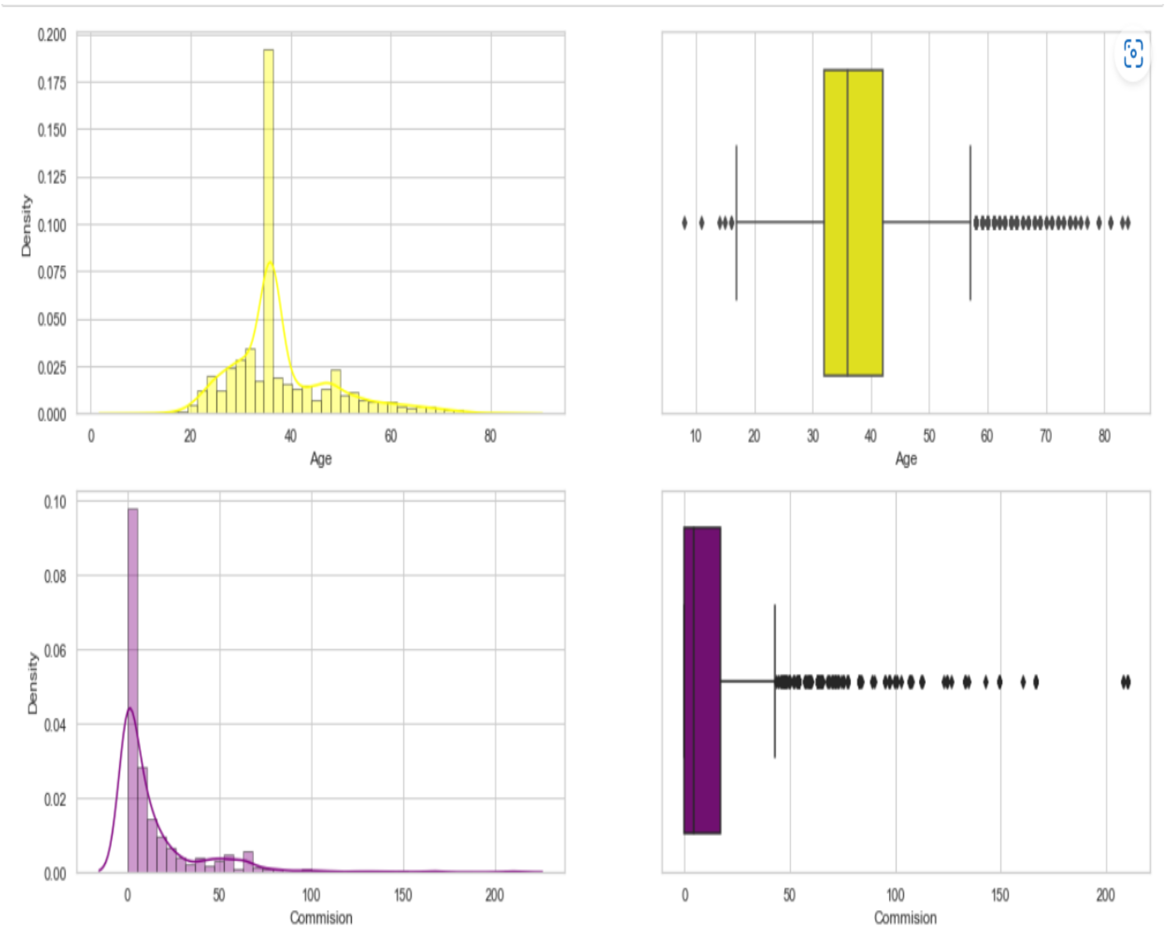
Descriptive Analysis of the dataset:

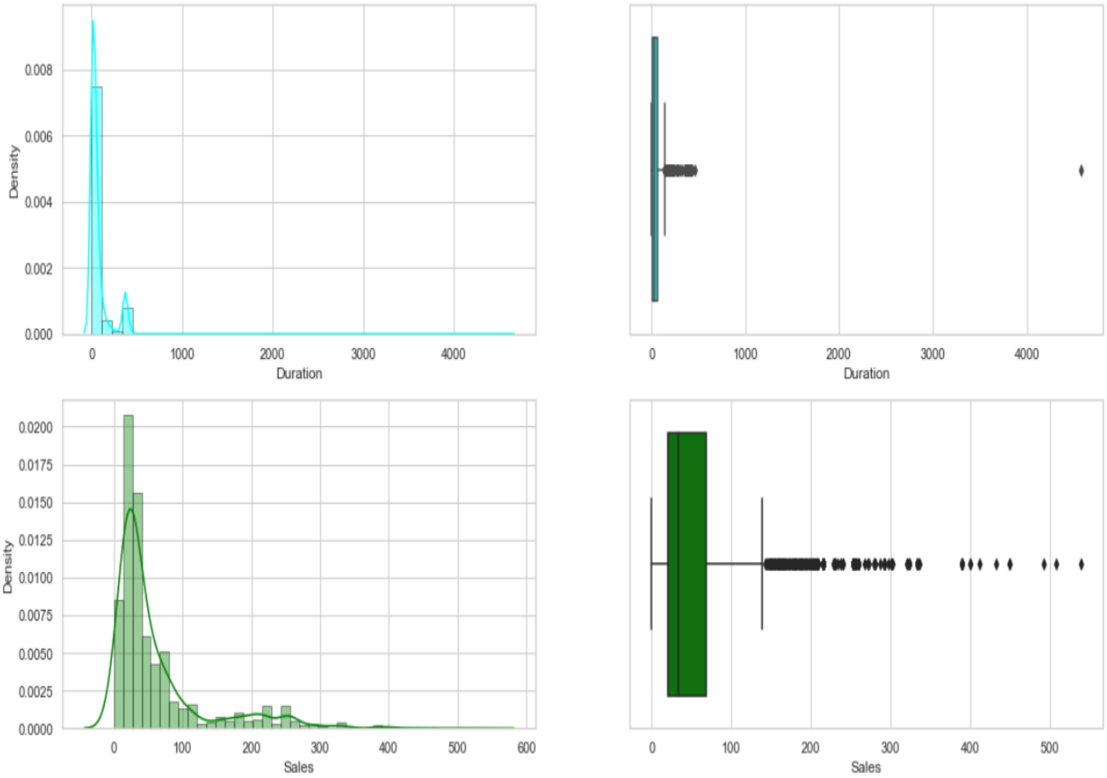


	count	mean	std	min	25%	50%	75%	90%	max
Age	3000.0	38.091000	10.463518	8.0	32.0	36.00	42.000	53.000	84.00
Commision	3000.0	14.529203	25.481455	0.0	0.0	4.63	17.235	48.300	210.21
Duration	3000.0	70.001333	134.053313	-1.0	11.0	26.50	63.000	224.200	4580.00
Sales	3000.0	60.249913	70.733954	0.0	20.0	33.00	69.000	172.025	539.00

Duplicates: There are 139 duplicate records present in the dataset. There is no unique identity of customers present, so this data can be of different customers of the company.

UNIVARIATE ANALYSIS:

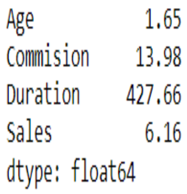




SKEWNESS:

```
Age            1.149713
Commision      3.148858
Duration      13.786136
Sales          2.381148
dtype: float64
```

KURTOSIS:



```
Age           1.65
Commision    13.98
Duration    427.66
Sales         6.16
dtype: float64
```

OBSERVATIONS:

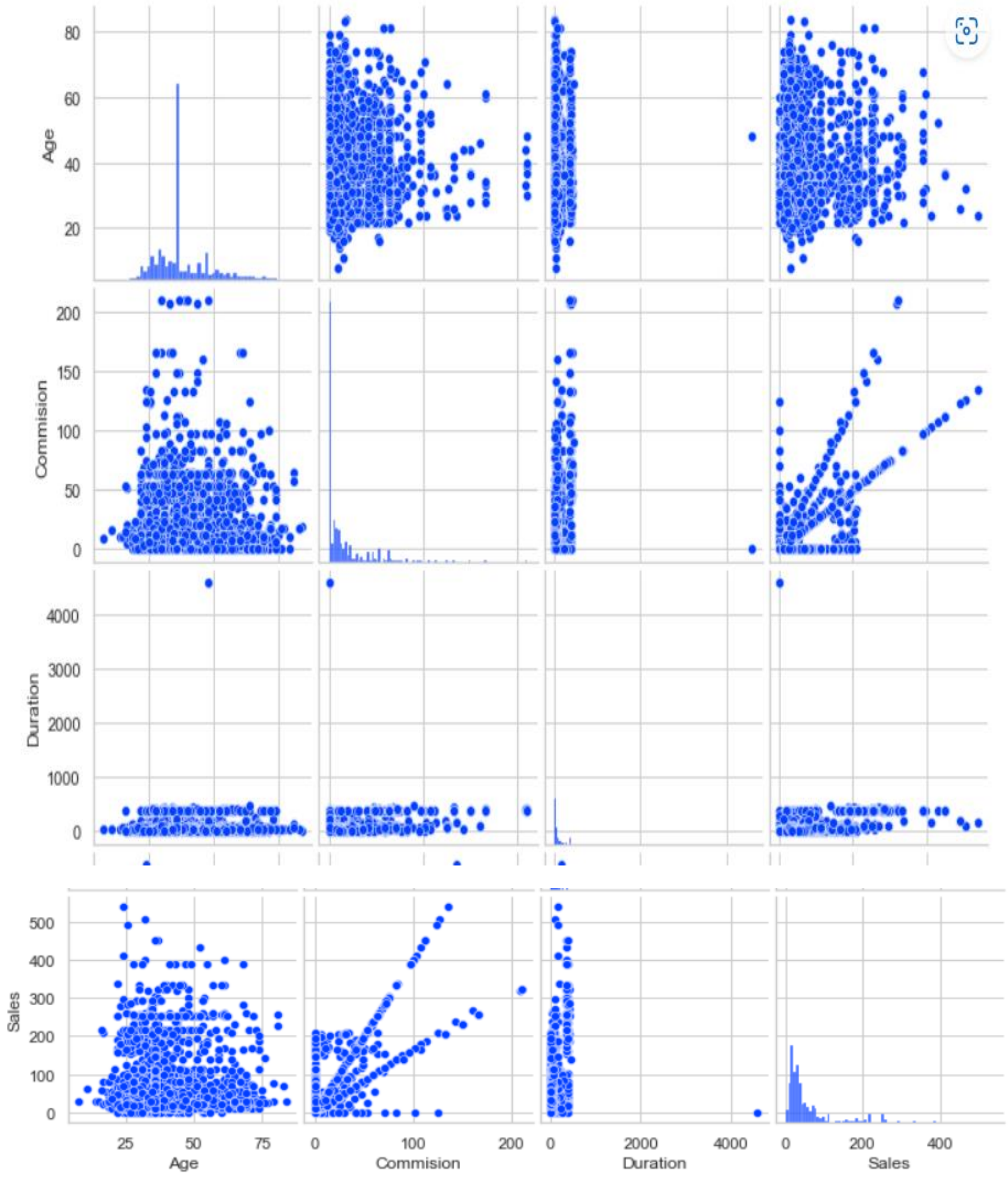
For all the above variables, mean is greater than median, hence it is positively skewed.

"Sales", "Commission" and "Duration" all 3 variables are positively Skewed with too many outliers

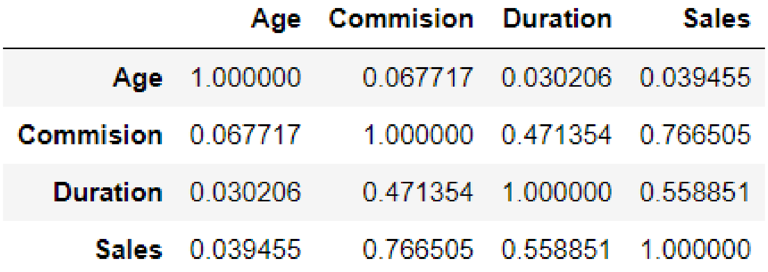
"Age" variable is little right skewed with too many outliers. There are outliers in all the variables.

CART and Random Forest are robust to outliers. Neural Networks can handle outliers if there are more hidden layers and if the number of outliers is lesser. For now, we are treating the outliers using IQR.

MULTIVARIATE ANALYSIS:

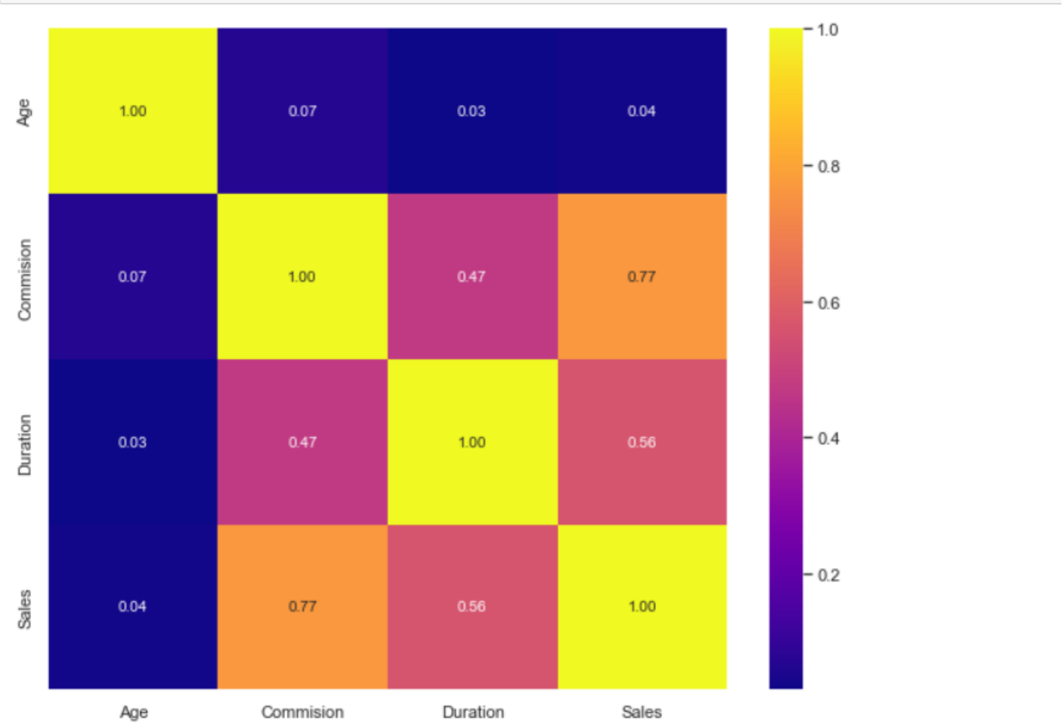


CORRELATION:



	Age	Commision	Duration	Sales
Age	1.000000	0.067717	0.030206	0.039455
Commision	0.067717	1.000000	0.471354	0.766505
Duration	0.030206	0.471354	1.000000	0.558851
Sales	0.039455	0.766505	0.558851	1.000000

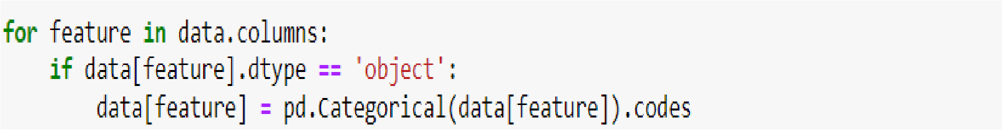
HEATMAP:



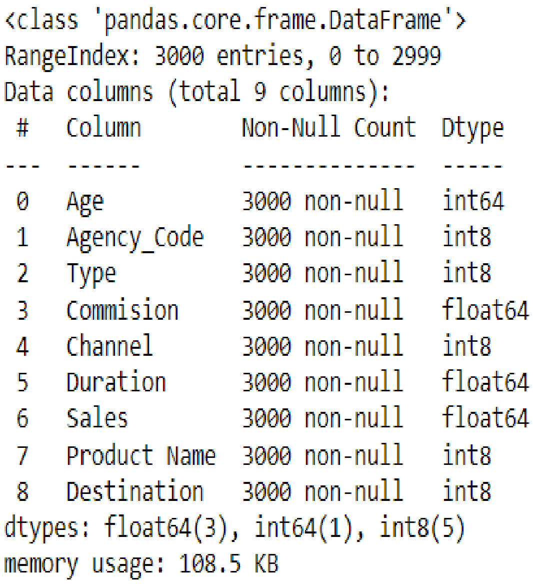
Looking at the correlation matrix, we can say that sales and commission are highly correlated, sales and duration are also correlated. While rest the variables are very less correlated.

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

Solution: Decision tree in Python can take only numerical / categorical columns. It cannot take string / object types. We applied loop through each column and checked if the column type is object, then converted those object columns into either integer or float.

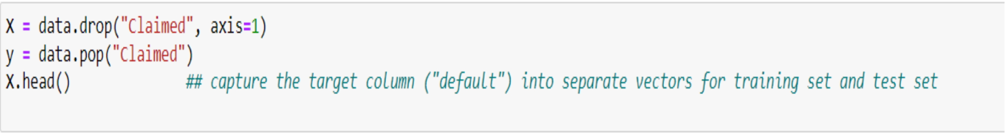


```
for feature in data.columns:
    if data[feature].dtype == 'object':
        data[feature] = pd.Categorical(data[feature]).codes
```

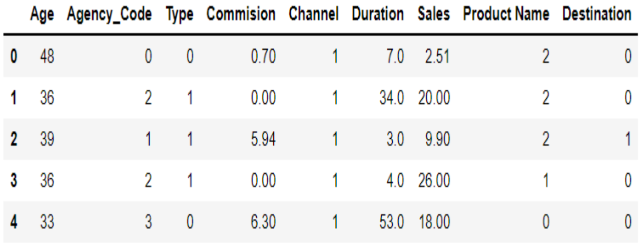


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Age           3000 non-null   int64  
 1   Agency_Code   3000 non-null   int8   
 2   Type          3000 non-null   int8   
 3   Commision     3000 non-null   float64
 4   Channel       3000 non-null   int8   
 5   Duration      3000 non-null   float64
 6   Sales         3000 non-null   float64
 7   Product Name  3000 non-null   int8   
 8   Destination   3000 non-null   int8   
dtypes: float64(3), int64(1), int8(5)
memory usage: 108.5 KB
```

Now,we split data into independent variable X and dependent variable Y (Claimed).

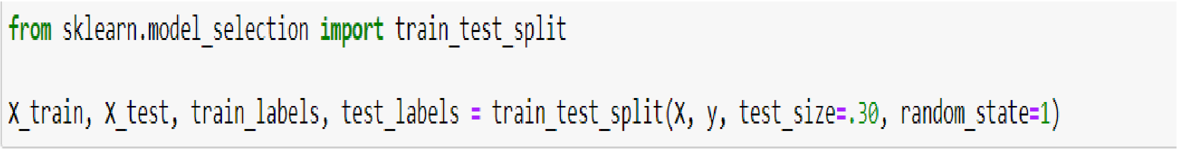


```
X = data.drop("Claimed", axis=1)
y = data.pop("Claimed")
X.head()            ## capture the target column ("default") into separate vectors for training set and test set
```



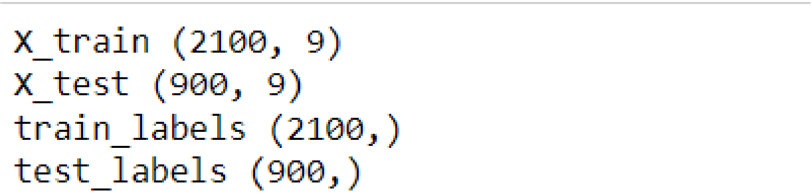
	Age	Agency_Code	Type	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	0	0	0.70	1	7.0	2.51	2	0
1	36	2	1	0.00	1	34.0	20.00	2	0
2	39	1	1	5.94	1	3.0	9.90	2	1
3	36	2	1	0.00	1	4.0	26.00	1	0
4	33	3	0	6.30	1	53.0	18.00	0	0

Then, we split the data into train and test sets by using:



```
from sklearn.model_selection import train_test_split

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

Training data has 2100 rows and 9 columns, and test data has 900 rows and 9 columns.

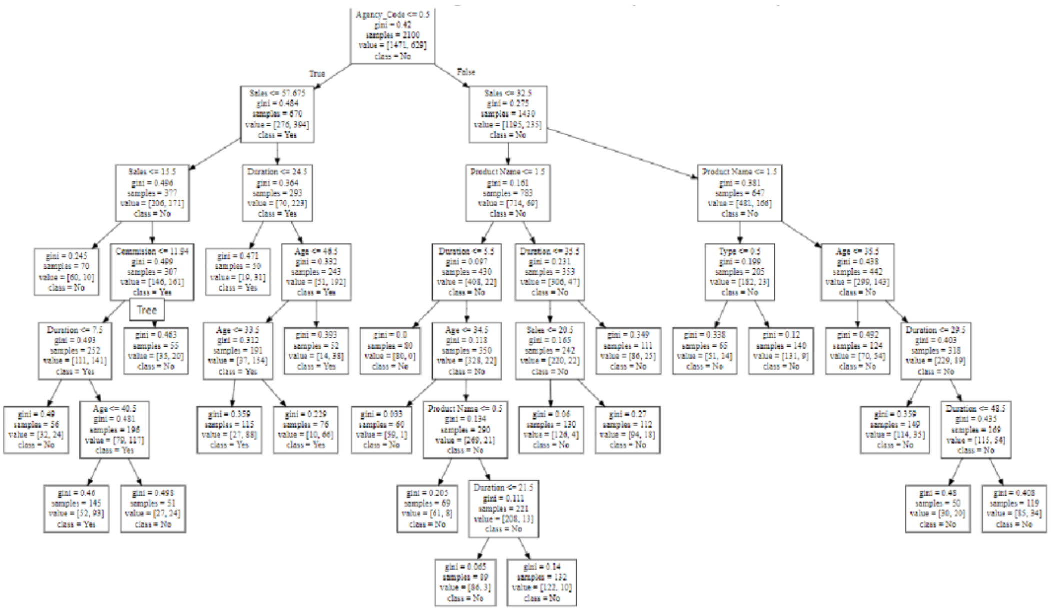


```
X_train (2100, 9)
X_test (900, 9)
train_labels (2100,)
test_labels (900,)
```

BUILDING CLASSIFICATION MODEL CART, RANDOM FOREST, ARTIFICIAL NEURAL NETWORK-

The next step is to build classification model for CART, Random Forest, ANN by using various submodules of sklearn library.

CART: In CART, we made the model by fitting and testing data to DecisionTreeClassifier and checked the result using http://webgraphviz.com/. We saw that overfitting of data occurs. To overcome this problem, we pruned the tree. Below is the snapshot of the pruned decision tree.



Agency_Code <= 0.5 gini = 0.42 samples = 2100 value = [1471, 629] class = No
True
False
Sales <= 57.675 gini = 0.484 samples = 670 value = [276, 394] class = Yes
Sales <= 32.5 gini = 0.275 samples = 1430 value = [1195, 235] class = No
Sales <= 15.5 gini = 0.496 samples = 377 value = [206, 171] class = No
Duration <= 24.5 gini = 0.364 samples = 293 value = [70, 223] class = Yes
Product Name <= 1.5 gini = 0.161 samples = 783 value = [714, 69] class = No
Product Name <= 1.5 gini = 0.381 samples = 647 value = [481, 166] class = No
gini = 0.245 samples = 70 value = [60, 10] class = No
Commision <= 11.94 gini = 0.499 samples = 307 value = [146, 161] class = Yes
gini = 0.471 samples = 50 value = [19, 31] class = Yes
Age <= 46.5 gini = 0.332 samples = 243 value = [51, 192] class = Yes
Duration <= 5.5 gini = 0.097 samples = 430 value = [408, 22] class = No
Duration <= 35.5 gini = 0.231 samples = 353 value = [306, 47] class = No
Type <= 0.5 gini = 0.199 samples = 205 value = [182, 23] class = No
Age <= 35.5 gini = 0.438 samples = 442 value = [299, 143] class = No
Tree
Duration <= 7.5 gini = 0.493 samples = 252 value = [111, 141] class = Yes
gini = 0.463 samples = 55 value = [35, 20] class = No
Age <= 33.5 gini = 0.312 samples = 191 value = [37, 154] class = Yes
gini = 0.393 samples = 52 value = [14, 38] class = Yes
gini = 0.0 samples = 80 value = [80, 0] class = No
Age <= 34.5 gini = 0.118 samples = 350 value = [328, 22] class = No
Sales <= 20.5 gini = 0.165 samples = 242 value = [220, 22] class = No
gini = 0.349 samples = 111 value = [86, 25] class = No
gini = 0.338 samples = 65 value = [51, 14] class = No
gini = 0.12 samples = 140 value = [131, 9] class = No
gini = 0.492 samples = 124 value = [70, 54] class = No
Duration <= 29.5 gini = 0.403 samples = 318 value = [229, 89] class = No
gini = 0.49 samples = 56 value = [32, 24] class = No
Age <= 40.5 gini = 0.481 samples = 196 value = [79, 117] class = Yes
gini = 0.359 samples = 115 value = [27, 88] class = Yes
gini = 0.229 samples = 76 value = [10, 66] class = Yes
gini = 0.033 samples = 60 value = [59, 1] class = No
Product Name <= 0.5 gini = 0.134 samples = 290 value = [269, 21] class = No
gini = 0.06 samples = 130 value = [126, 4] class = No
gini = 0.27 samples = 112 value = [94, 18] class = No
gini = 0.359 samples = 149 value = [114, 35] class = No
Duration <= 48.5 gini = 0.435 samples = 169 value = [115, 54] class = No
gini = 0.46 samples = 145 value = [52, 93] class = Yes
gini = 0.498 samples = 51 value = [27, 24] class = No
gini = 0.205 samples = 69 value = [61, 8] class = No
Duration <= 21.5 gini = 0.111 samples = 221 value = [208, 13] class = No
gini = 0.48 samples = 50 value = [30, 20] class = No
gini = 0.408 samples = 119 value = [85, 34] class = No
gini = 0.065 samples = 89 value = [86, 3] class = No
gini = 0.14 samples = 132 value = [122, 10] class = No

RANDOM FOREST: We fit random forest model into train and test data and used Grid Search Cross Validation function from python to find out the best parameters.

ARTIFICIAL NEURAL NETWORK: To create ANN MODEL, we first scaled the data using StandardScaler function. Then we fitted training and test data into MLP classifier. We also used GridSearchCV to find out the best parameters.

We will now create classification report confusion matrix and accuracy store after finding out the best parameters using cross validation.

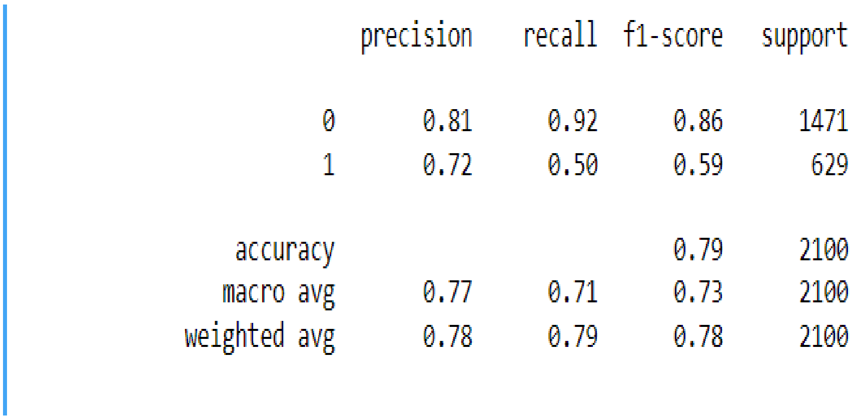
2.3. Performance Metrics: Comment and Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score, classification reports for each model.

Solution:

Performance matrices (confusion matrix, AUC SCORE, classification report ROC CURVE)

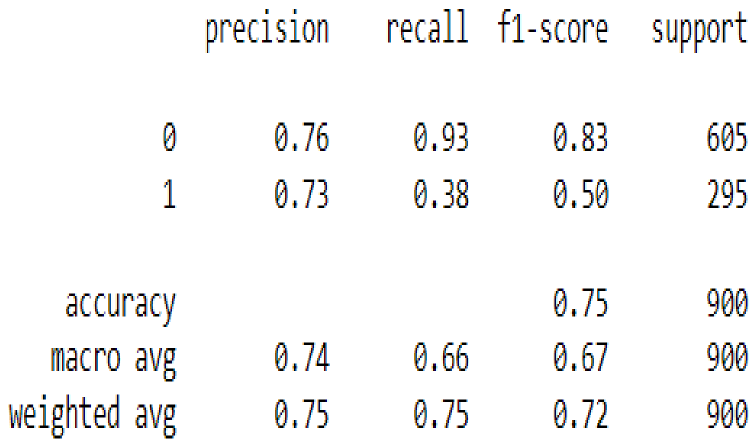
<u>DECISION TREE (confusion matrix, AUC SCORE, classification report ROC CURVE) OF TRAINING AND TEST DATA:</u>

DECISION tree classification report for train dataset:



```
              precision    recall  f1-score   support

           0       0.81      0.92      0.86      1471
           1       0.72      0.50      0.59       629

    accuracy                           0.79      2100
   macro avg       0.77      0.71      0.73      2100
weighted avg       0.78      0.79      0.78      2100
```

DECISION tree classification report for test dataset:

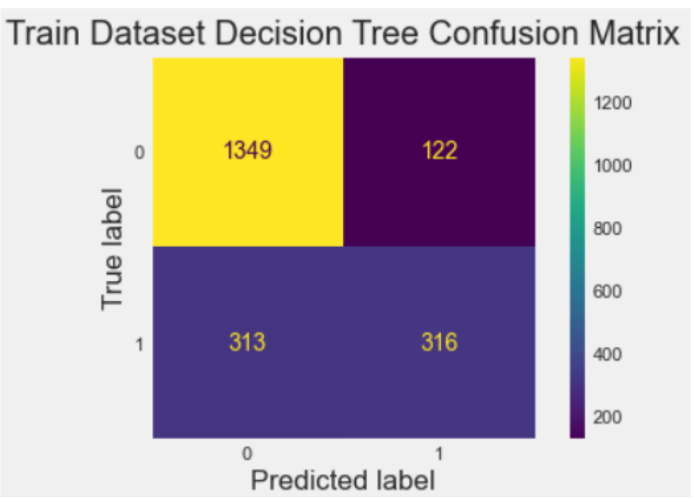


```
              precision    recall  f1-score   support

           0       0.76      0.93      0.83       605
           1       0.73      0.38      0.50       295

    accuracy                           0.75       900
   macro avg       0.74      0.66      0.67       900
weighted avg       0.75      0.75      0.72       900
```

-

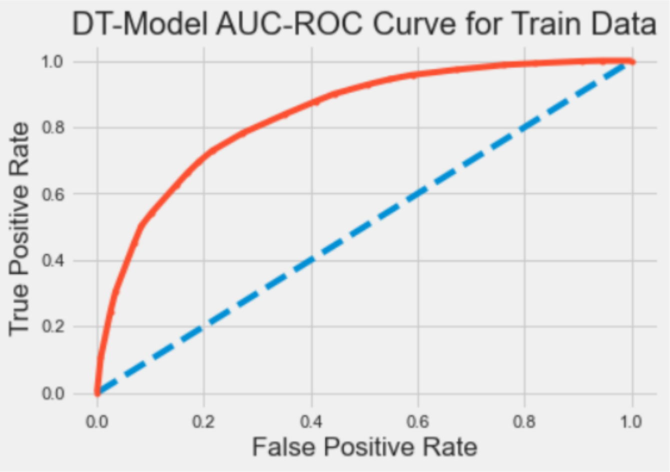
There is an accuracy of 79% in training dataset and 75% in test dataset.

Decision tree dataset confusion matrix:



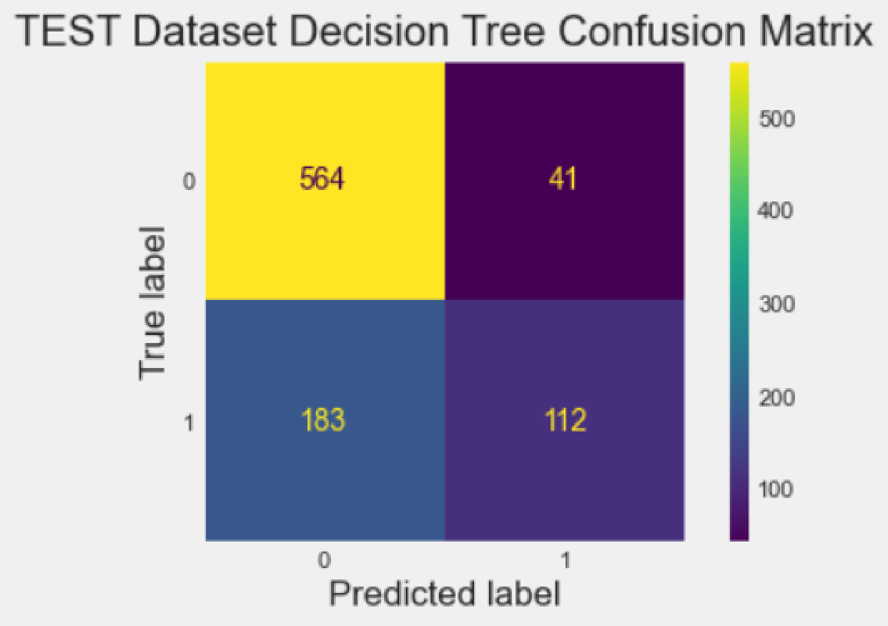
Decision Tree train data AUC:

83.6%

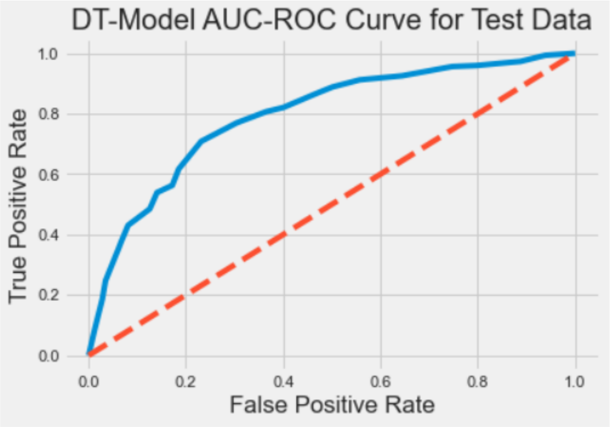


DECISION TREE (confusion matrix, AUC SCORE, classification report ROC CURVE) OF TEST DATA:

DECISION tree confusion matrix for test dataset:



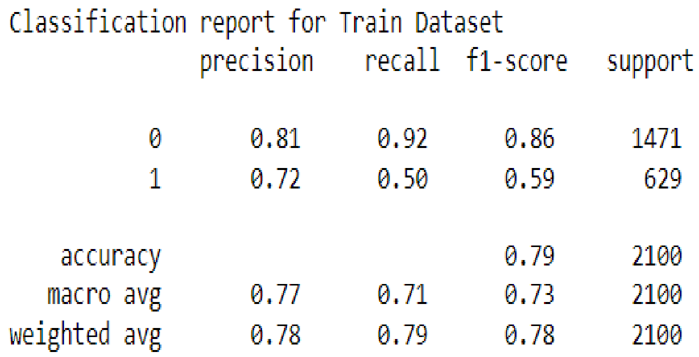
DECISION TREE AUC of test data: 79.28 %



RANDOM FOREST(confusion matrix, AUC SCORE, classification report ROC CURVE) OF TRAINING AND TEST DATA:

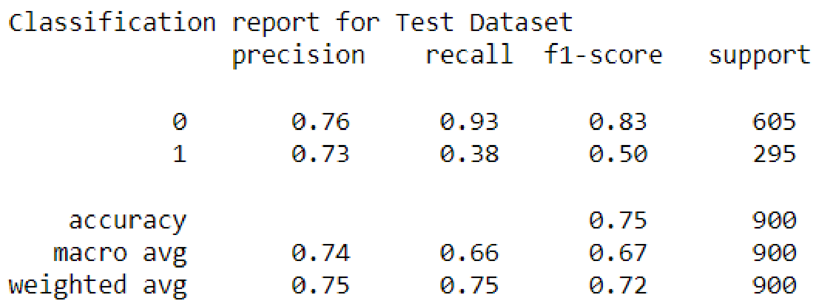
PERFORMANCE EVALUATION OF RANDOM FOREST:

Random forest classification report for train dataset:



```
Classification report for Train Dataset
              precision    recall  f1-score   support

           0       0.81      0.92      0.86      1471
           1       0.72      0.50      0.59       629

    accuracy                           0.79      2100
   macro avg       0.77      0.71      0.73      2100
weighted avg       0.78      0.79      0.78      2100
```

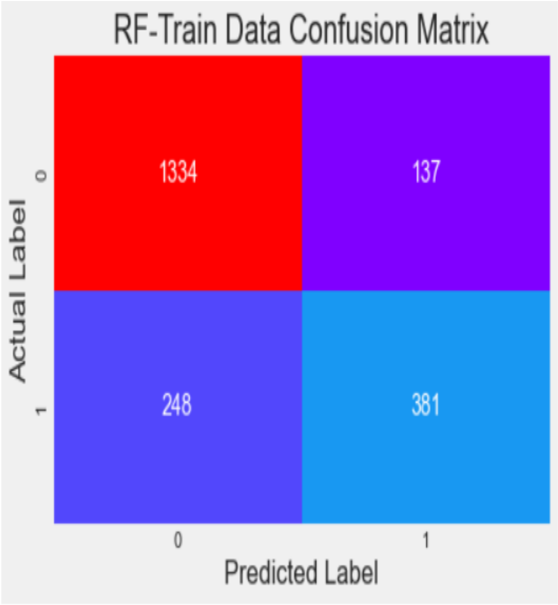
Random forest classification report for test dataset



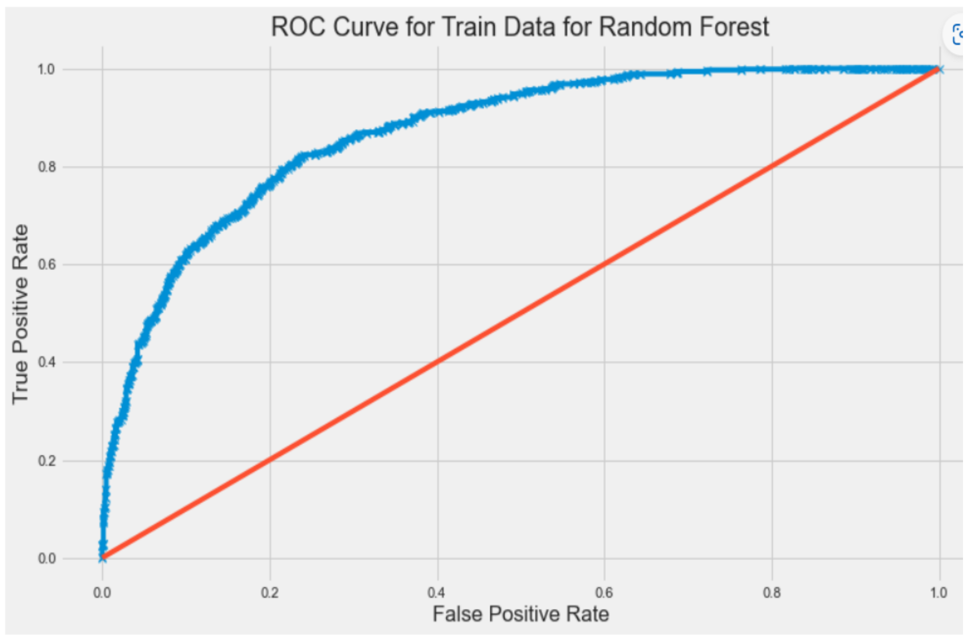
```
Classification report for Test Dataset
              precision    recall  f1-score   support

           0       0.76      0.93      0.83       605
           1       0.73      0.38      0.50       295

    accuracy                           0.75       900
   macro avg       0.74      0.66      0.67       900
weighted avg       0.75      0.75      0.72       900
```

Random forest dataset confusion matrix:

Train data confusion matrix

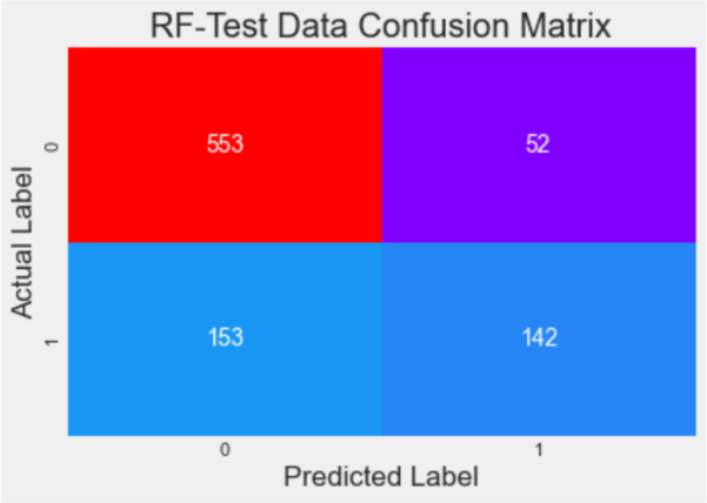


RANDOM FOREST :AUC SCORE FOR TRAIN DATA: 87%

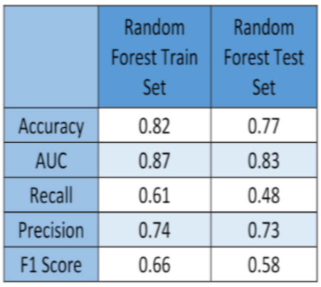


```
Area under Curve is 0.869469521506951
```

RANDOM FOREST-TEST DATA EVALUATION



Random forest AUC SCORE for test data: 83%

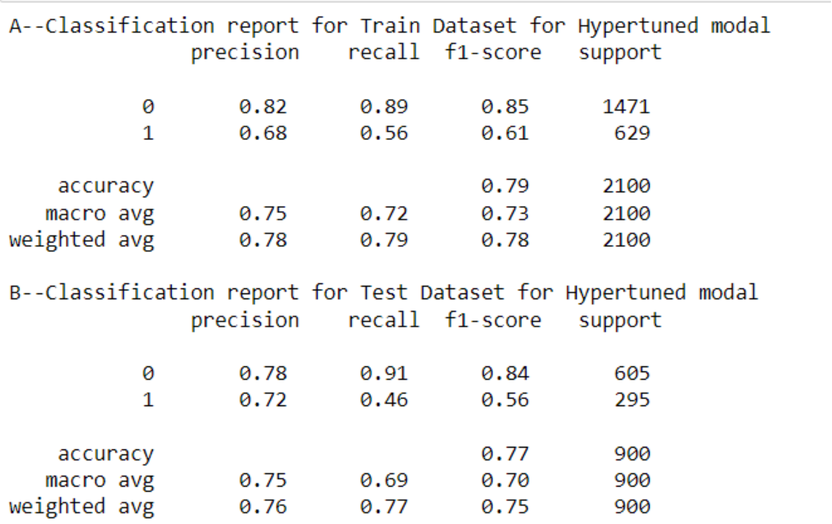


	Random Forest Train Set	Random Forest Test Set
Accuracy	0.82	0.77
AUC	0.87	0.83
Recall	0.61	0.48
Precision	0.74	0.73
F1 Score	0.66	0.58

By analysing the random forest model, we got the above results. Since the results of train and test datasets are almost equal, this random model is performing well.

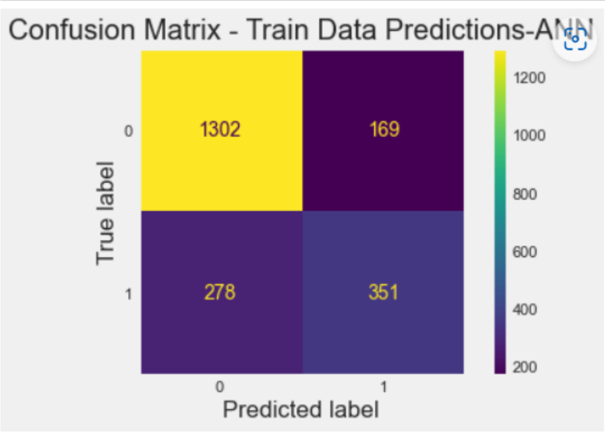
ARTIFICIAL NEURAL NETWORK MODEL:

Classification report for train and test dataset for hyper tuned ANN Model:

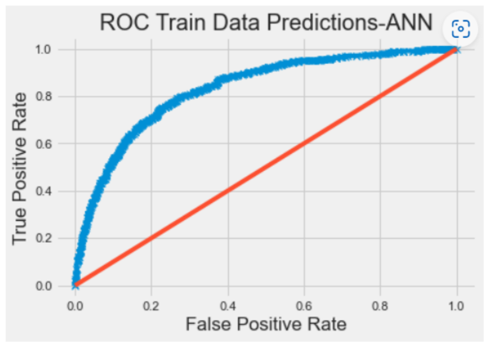


```
A--Classification report for Train Dataset for Hypertuned modal
               precision    recall  f1-score   support

           0       0.82      0.89      0.85      1471
           1       0.68      0.56      0.61       629

    accuracy                           0.79      2100
   macro avg       0.75      0.72      0.73      2100
weighted avg       0.78      0.79      0.78      2100

B--Classification report for Test Dataset for Hypertuned modal
               precision    recall  f1-score   support

           0       0.78      0.91      0.84       605
           1       0.72      0.46      0.56       295

    accuracy                           0.77       900
   macro avg       0.75      0.69      0.70       900
weighted avg       0.76      0.77      0.75       900
```

Confusion matrix Train dataset:

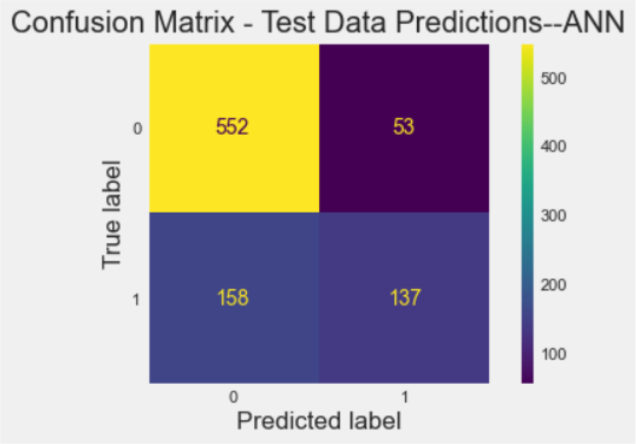


ANN AUC score of train data: 84%

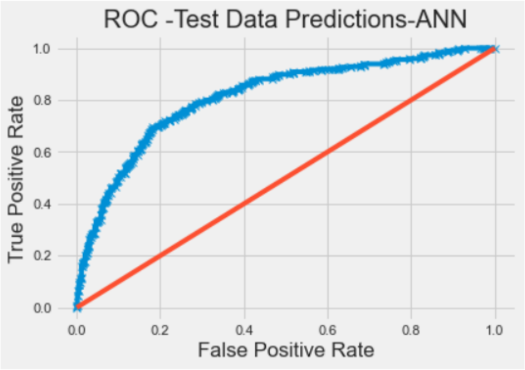


```
Area under Curve is 0.8322431881235415
```

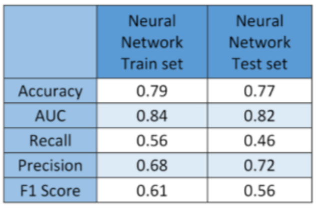
ANN TEST DATA PREDICTION:



ANN AUC score of train data: 82%



```
Area under Curve is 0.8134220479058691
```

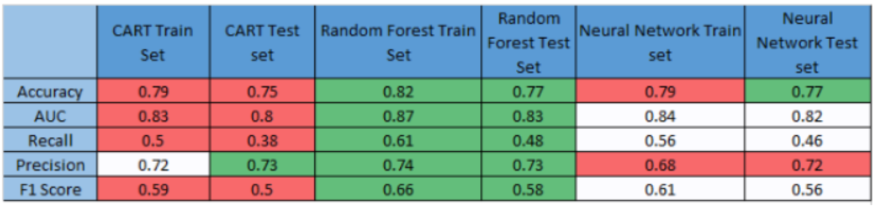


	Neural Network Train set	Neural Network Test set
Accuracy	0.79	0.77
AUC	0.84	0.82
Recall	0.56	0.46
Precision	0.68	0.72
F1 Score	0.61	0.56

By analysing the random forest model, we got the above results. Since the results of train and test datasets are almost equal, this ANN model is also performing well.

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

Solution:



	CART Train Set	CART Test set	Random Forest Train Set	Random Forest Test Set	Neural Network Train set	Neural Network Test set
Accuracy	0.79	0.75	0.82	0.77	0.79	0.77
AUC	0.83	0.8	0.87	0.83	0.84	0.82
Recall	0.5	0.38	0.61	0.48	0.56	0.46
Precision	0.72	0.73	0.74	0.73	0.68	0.72
F1 Score	0.59	0.5	0.66	0.58	0.61	0.56

In all the models, Train and Test results are very much same, but we have to compare all the Models and find out the best model among all.

we can notice here on the table that in Random Forest Model, the Train and test set results are highest among all the models. So, we can conclude that Random Forest model is the best/optimized Model. Neural network Model has given optimum results, but their results are lesser optimized in comparison to Random Forest Model. Thus, this model is performed model after random forest model. In the CART test model, both Train and Test set results are the lowest compared to all .so this is the least Optimized Model.

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

Solution:

Following are my Suggestions to Management

• We can see 90% of insurance is done by online mode, so we can increase their online experiences and also show the reviews about their insurance for benefitted customers. This will increase the Confidence of customers which will increase the insurance sales and profit too.

• More sales happen via Agency than Airlines and the trend shows the claim are processed more at Airline (See Box plot below. So, have to find out why this is happening?

• We need to talk to JZI & CWT agency to pick up sales as they Have lower sales, We can also tell them to use Digital marketing tools to increase their reach to the customers .

• Insurance Company can also increase their Insurance Portfolio by adding more insurance plans like health insurance, Corporate accidental Insurance etc to increase their revenue.

Below are some Key performance indicators (KPI) of insurance claims are:

- Average Cost Per Claim

- Claim Frequency

- Average Time to Settle a Claim

- Claims Ratio

- Increase customer satisfaction

Based on the above Insurance Company can make automated reporting process to senior officials of the company if a claim is pending from more than 1 week so that genuine claims can’t be neglected; this will Build the reputation of the company in the customer’s eye.