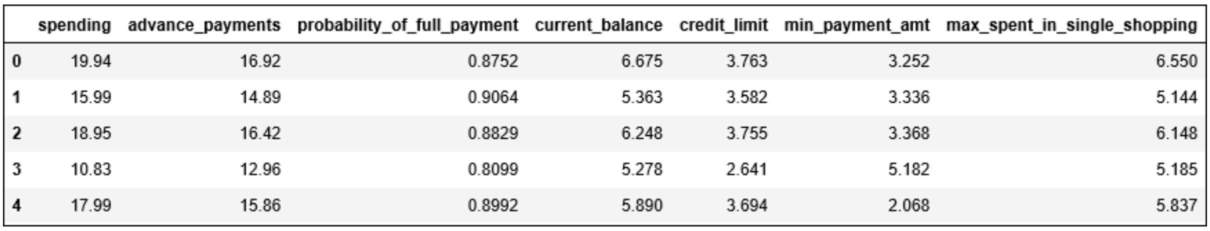
ANSWER REPORT

Data Mining

Problem Statement 1

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.

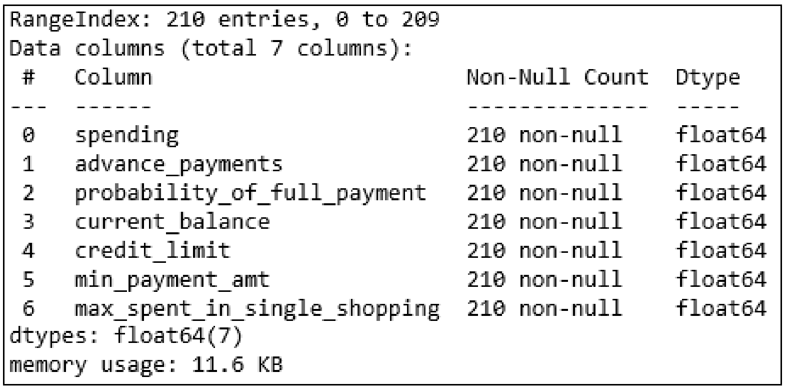
1.1 Read the data and do exploratory data analysis. Describe the data briefly.



	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

These are the top 5 rows of the data, with double digit values in Spending and advance_payments, single digit values in current_balance, credit_limit, min_payment_amt and max_spent_in_single_shopping, and point values in probability_of_full_payment.

The shape of the data is `(210, 7)`



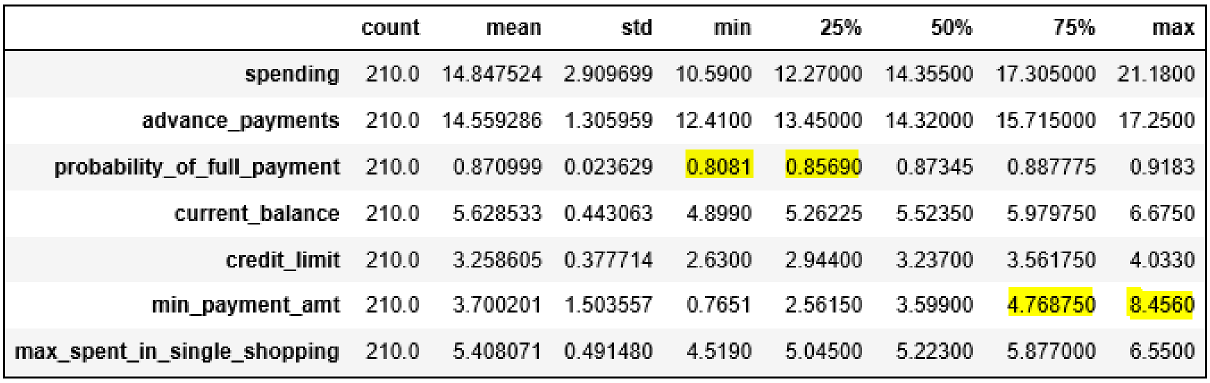
```
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   spending                      210 non-null    float64
 1   advance_payments              210 non-null    float64
 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 no null values in the dataset and all the values are of Float data type.

Now that we see the descriptive analysis of the data, we notice in the below table that, the difference between 25th percentile and min value of Probability_of_full_payment is large, we can determine that there must be some outliers in this column.

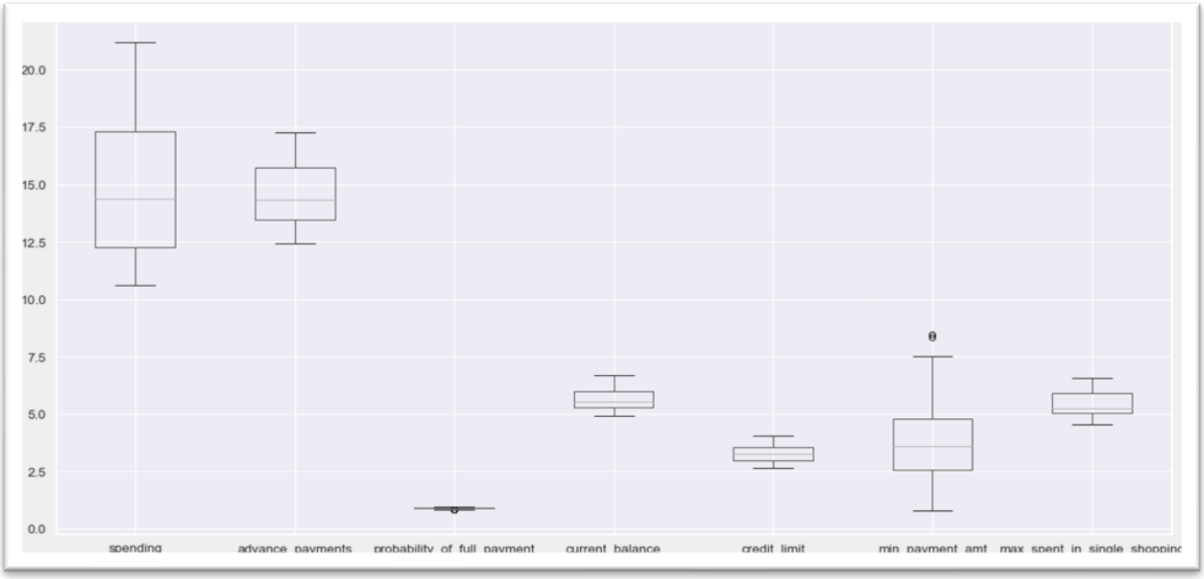
Also, the box plot will be right skewed as the range of Probability_of_full_payment data has more weightage from its 25th percentile to median is more than the median to 75th percentile

Similarly, for min_payment_amt since the difference is quite large, we can determine that there will be outliers in this data column.



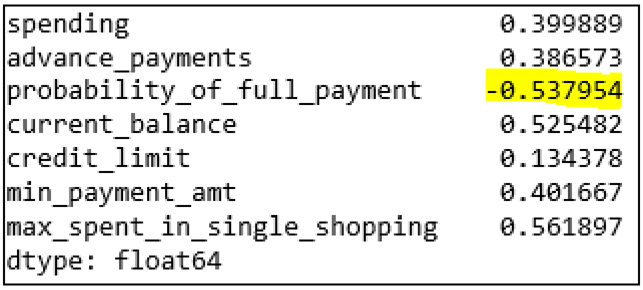
	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.847524	2.909699	10.5900	12.27000	14.35500	17.305000	21.1800
advance_payments	210.0	14.559286	1.305959	12.4100	13.45000	14.32000	15.715000	17.2500
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.85690	0.87345	0.887775	0.9183
current_balance	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

Before looking at the outliers in the dataset, we confirmed if we have any duplicate values in the data. There are no Duplicate values in the dataset.



As analysed in the description table of the data, there are outliers in both probability_of_full_payment and min_payment_amt. However, the number of outliers is quite less, therefore, I don't see a point in treating the outliers as treating them may disturb the data.

Apart from this, we can see that in the above figure that almost all the box plots seem to be positively skewed. Let's have a look at variable skewness in the next code.



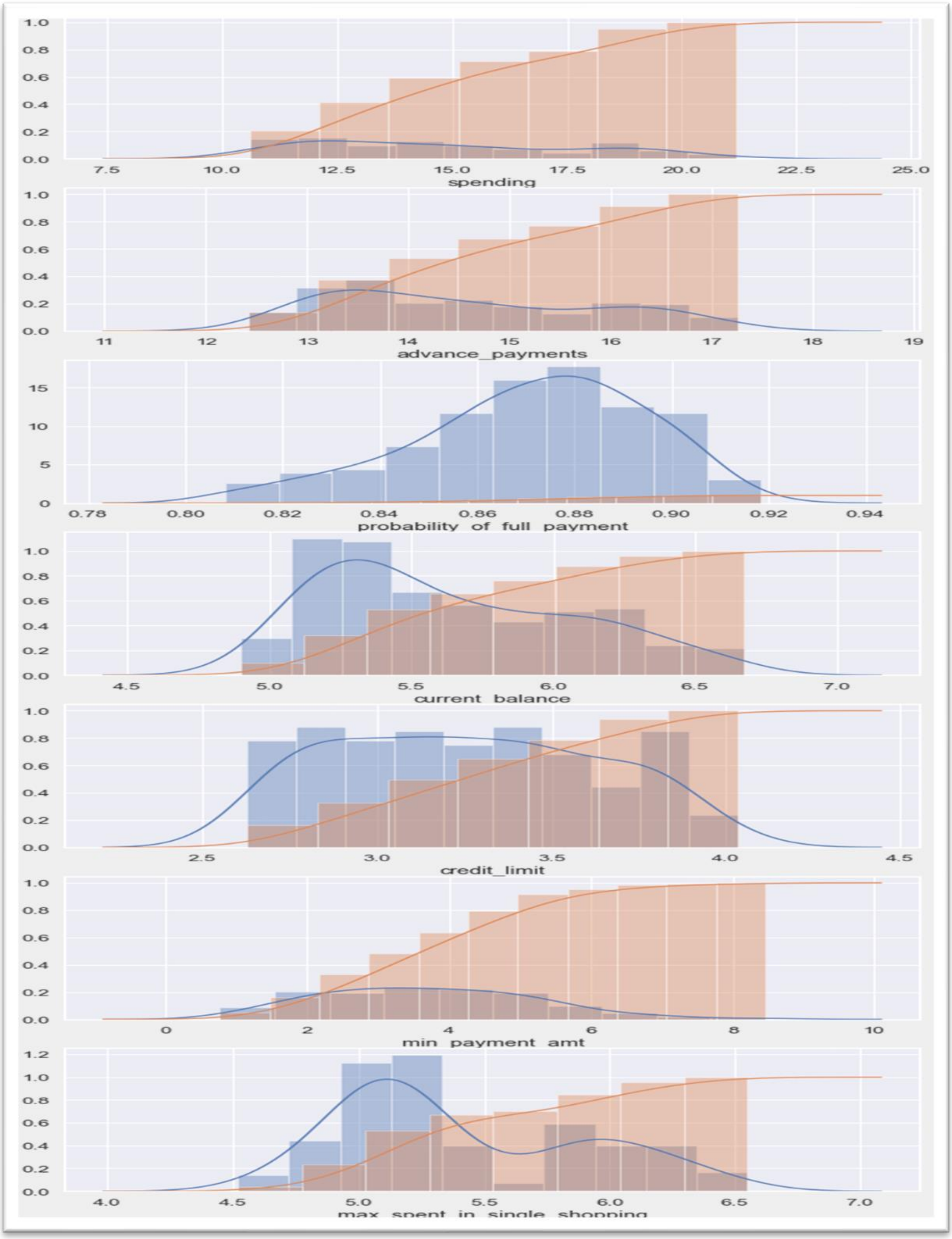
```
spending                        0.399889
advance_payments                0.386573
probability_of_full_payment    -0.537954
current_balance                 0.525482
credit_limit                    0.134378
min_payment_amt                 0.401667
max_spent_in_single_shopping    0.561897
dtype: float64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

In the above data, we can see that apart from probability_of_full_payment, all the other variables are positively skewed. Also, we can see that the range of Skewness values lies between -0.5 and 0.5, which conveys that the distribution is approximately symmetric.

Now, let's look at the data in a more detailed manner with Univariate and Multivariate analysis:

UNIVARIATE ANALYSIS:

The analysis of univariate data is thus the simplest form of analysis since the information deals with only one quantity that changes. It does not deal with causes or relationships and the main purpose of the analysis is to describe the data and find patterns that exist within it.

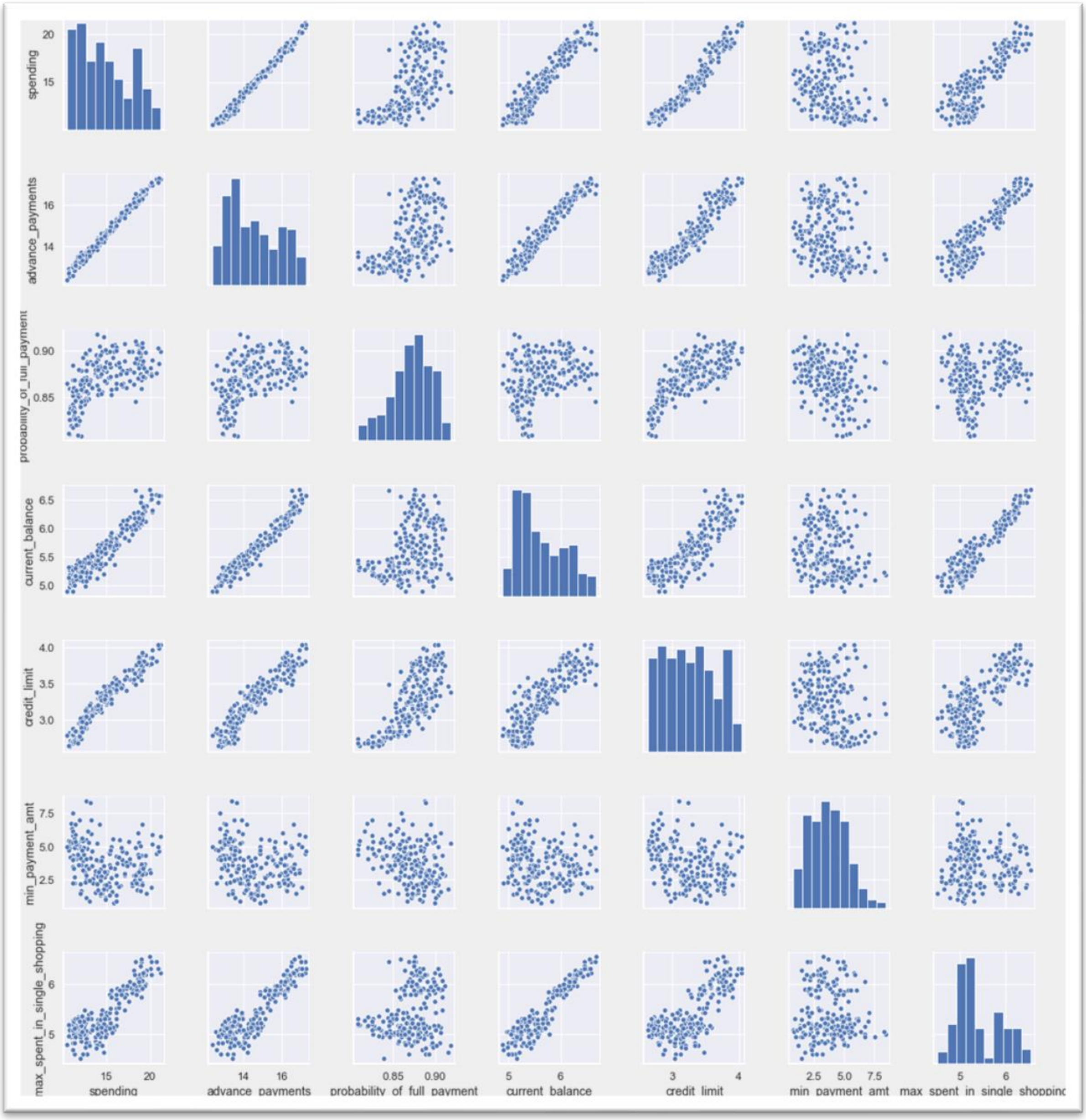


In the above images, we have combined the histogram with its cumulative graph, to have an inference of the data at individual level.

Apart from this, we can see a constant growth in each of the cumulative frequencies, apart from max_spent_in_single_shopping, where there is slight dip in the data for the interval 5.5 - 5.75.

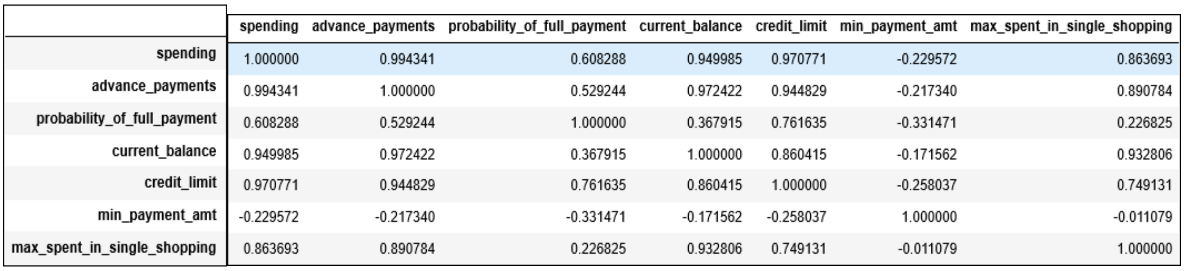
With the lines of both cumulative and histogram, we can see how constant the data is.

MULTIVARIATE ANALYSIS:

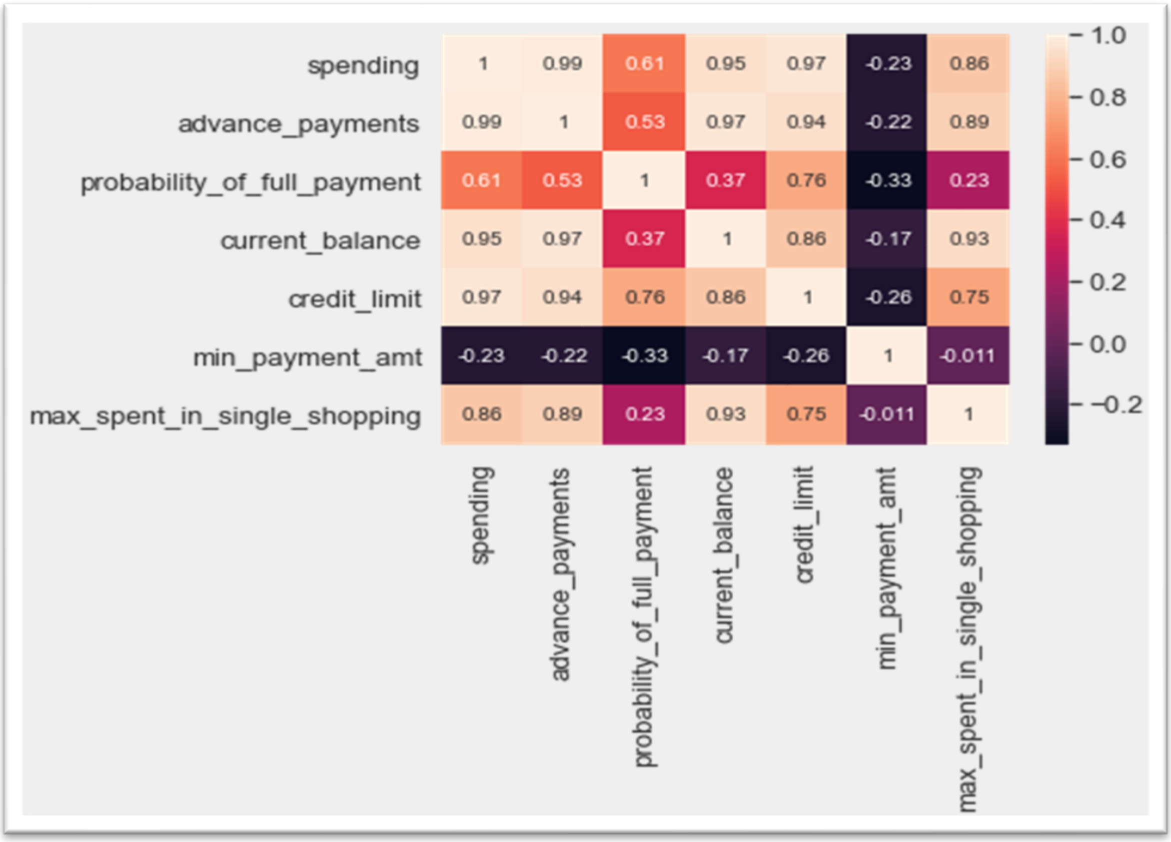


Upon performing Multivariate analysis on Variables present in the dataset, we get to infer using scatterplot for all the variables.

We can see high correlations in a lot of variables here, let’s have a look at the correlation table and the heatmap for more clarity.



	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



Observing the Correlations in the dataset , we infer that high multicollinearity exists within any two Independent variables. However, we will not be treating multi-collinearity, as it does not impact the clustering process.

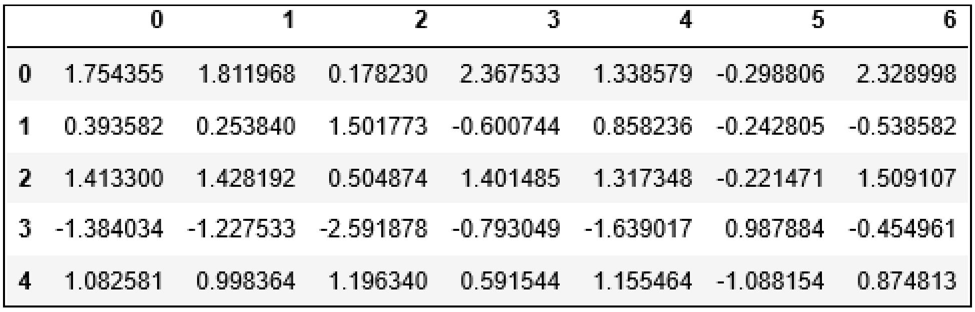
1.2Do you think scaling is necessary for clustering in this case? Justify

Yes, I think that Scaling is necessary for clustering in this case as standardising the data prevents variables with larger scales from dominating the clustering process. We scaled the data using Standard scaler.

I tried both Standardisation using Z score and normalisation sing Max-Min logic and concluded using the dataset with Standardisation over Min-Max scaling, since we are interested in the components that maximize the variance.

From the datasets, we can clearly notice that applying Max-Min Nominalisation in our dataset has generated smaller standard deviations than using Standardisation method. It implies the data are more concentrated around the mean if we scale data using Max-Min Nominalisation.

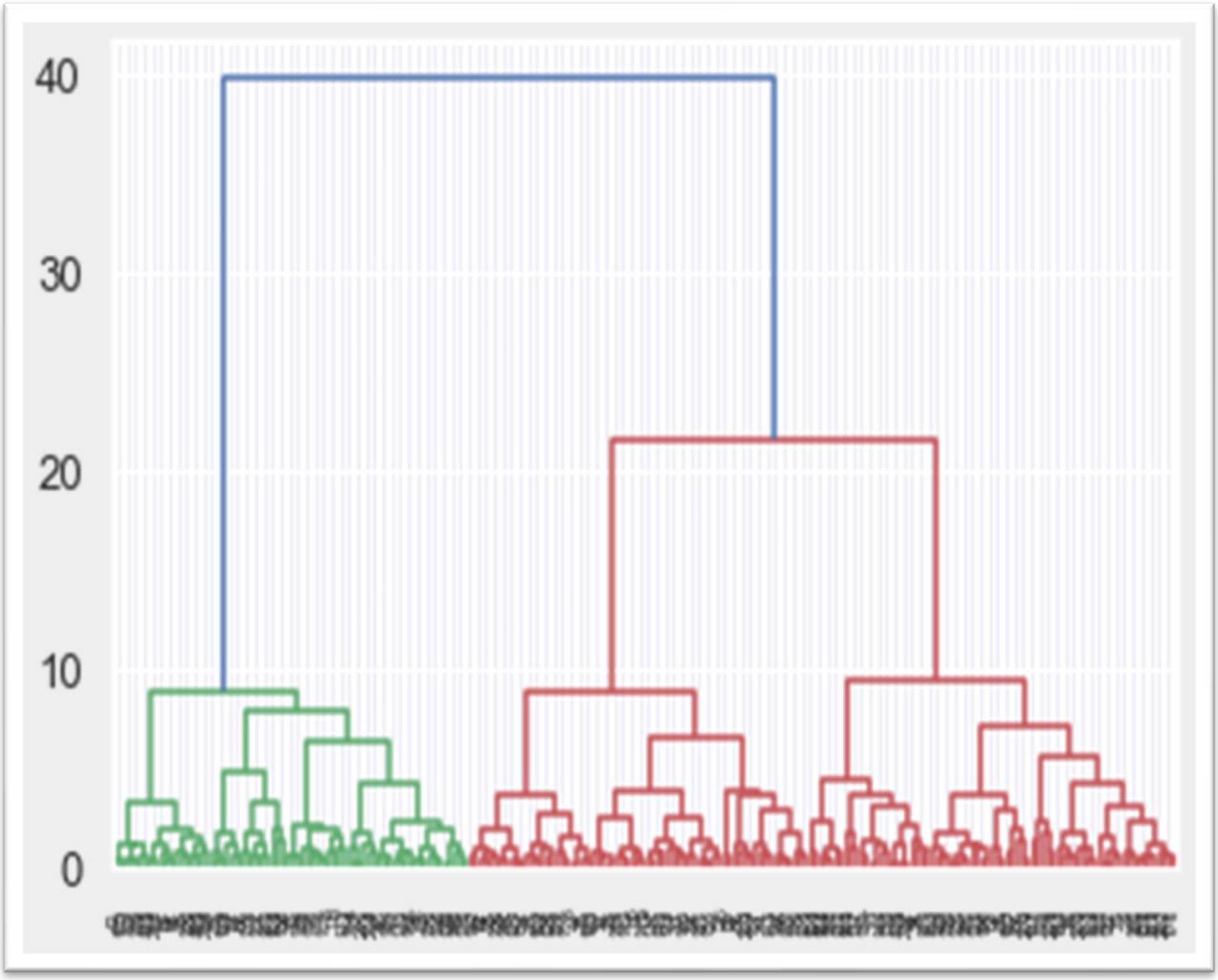
The head of the scaled data is:



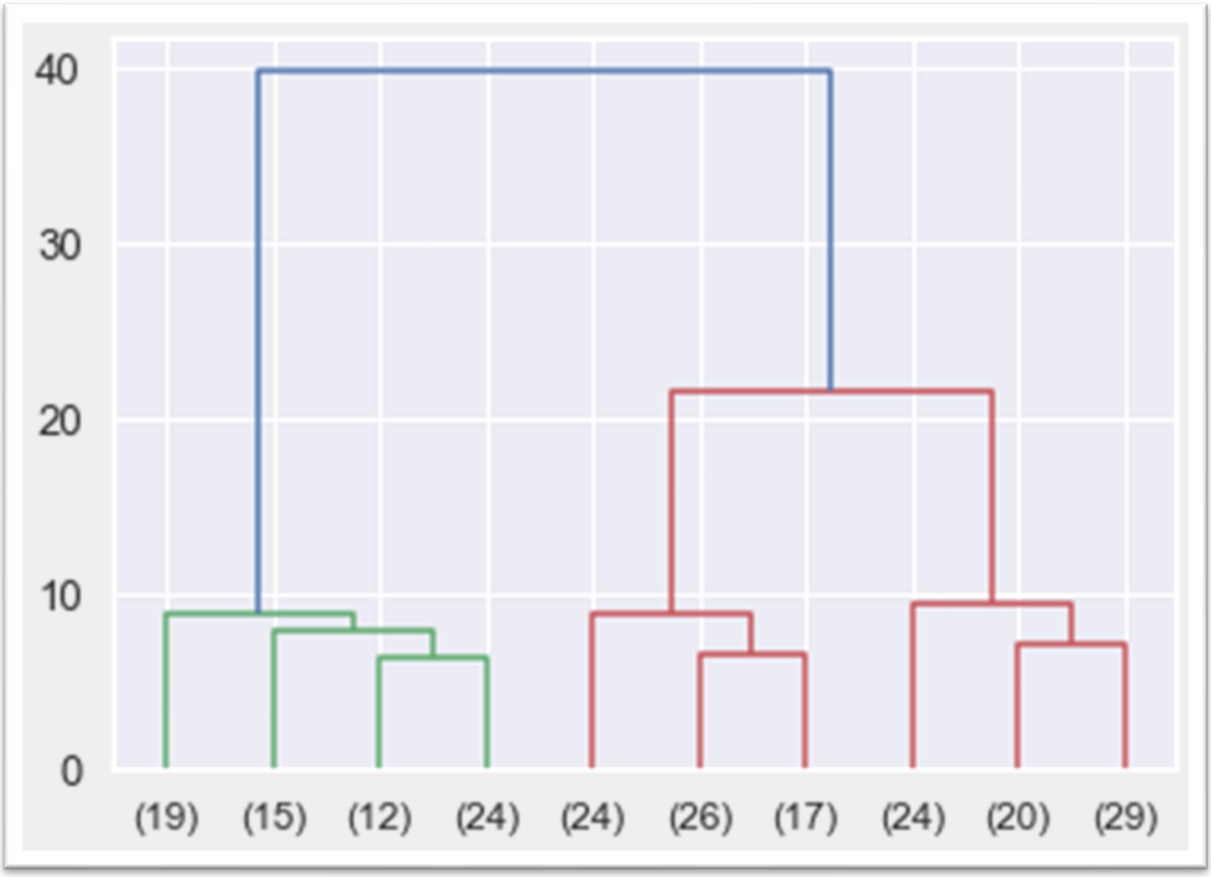
	0	1	2	3	4	5	6
0	1.754355	1.811968	0.178230	2.367533	1.338579	-0.298806	2.328998
1	0.393582	0.253840	1.501773	-0.600744	0.858236	-0.242805	-0.538582
2	1.413300	1.428192	0.504874	1.401485	1.317348	-0.221471	1.509107
3	-1.384034	-1.227533	-2.591878	-0.793049	-1.639017	0.987884	-0.454961
4	1.082581	0.998364	1.196340	0.591544	1.155464	-1.088154	0.874813

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

We construct a Dendrogram for the scaled data and we obtain different cluster patterns using different linkages and distance criterions. Ward method was chosen after analysing all the dendrograms:

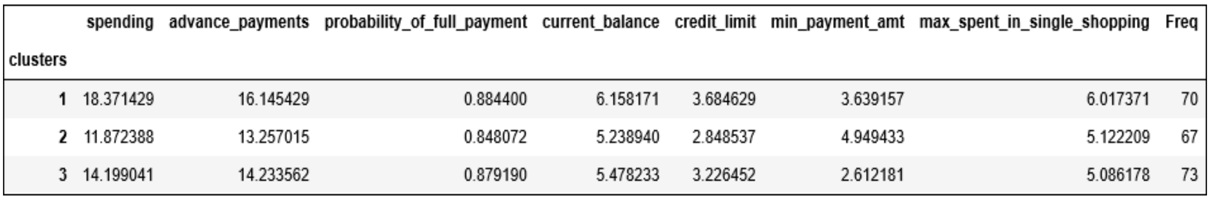


Let’s have a closer look:



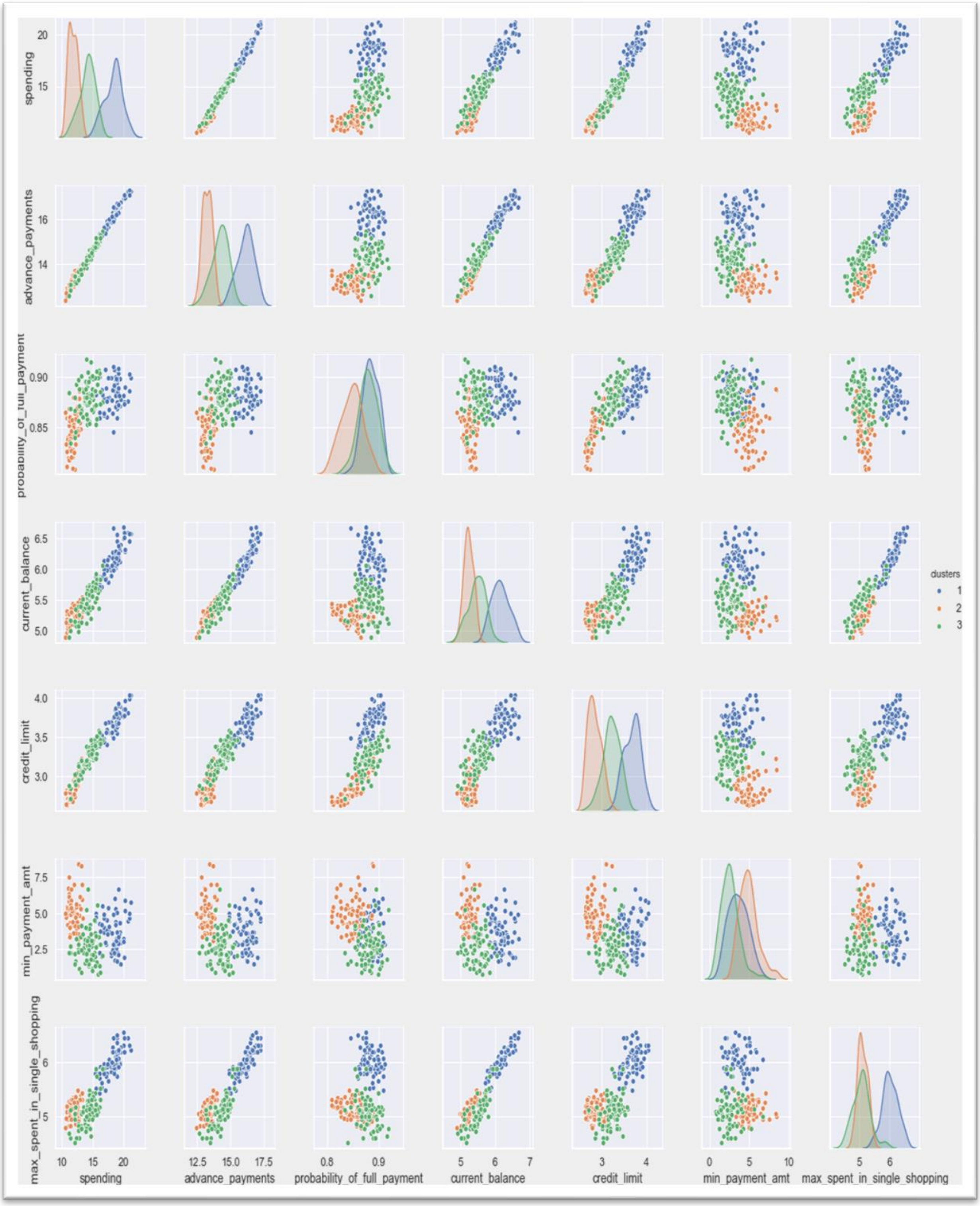
Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly like each other.

Upon performing hierarchical clustering on scaled dataset, we obtain mean values within 3 cluster formations as follows:

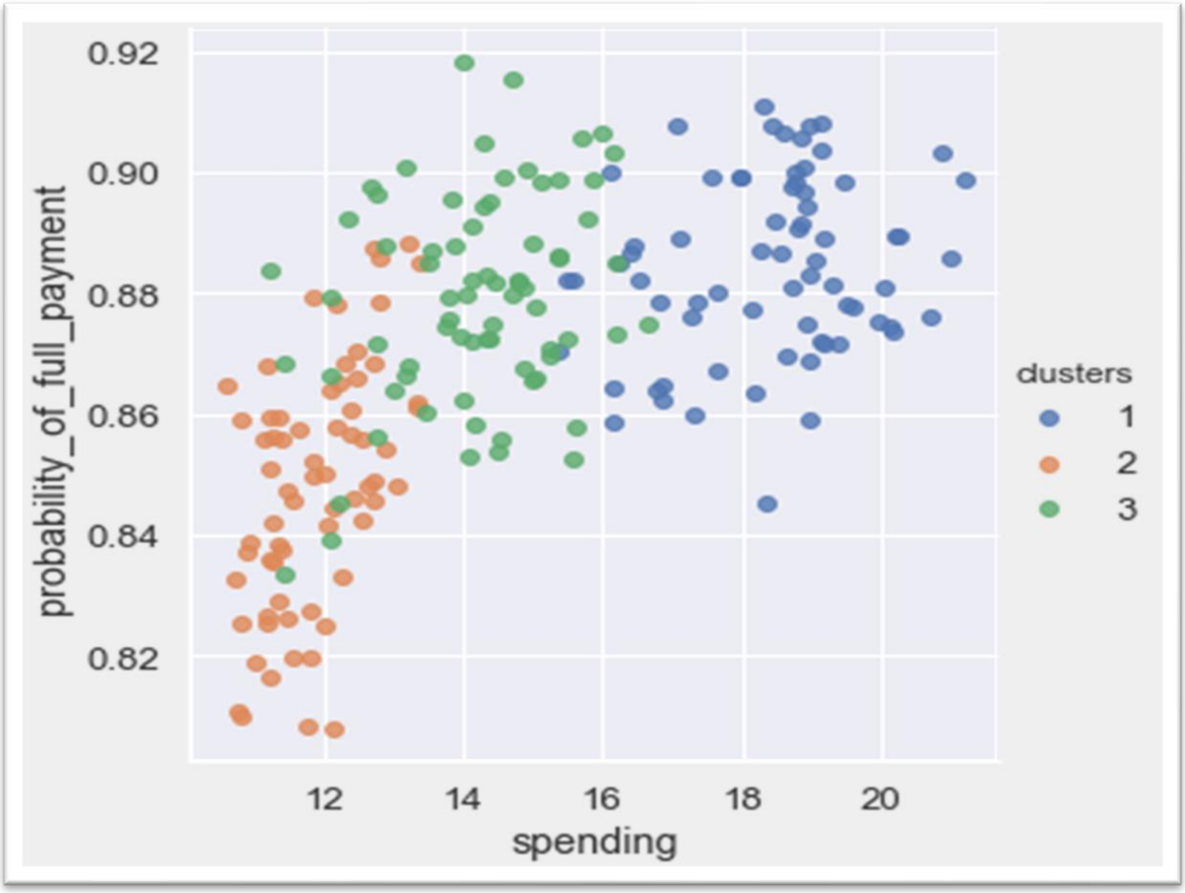


	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
clusters								
1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371	70
2	11.872388	13.257015	0.848072	5.238940	2.848537	4.949433	5.122209	67
3	14.199041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

Let’s have a visual representation of the clustering applied on the dataset:

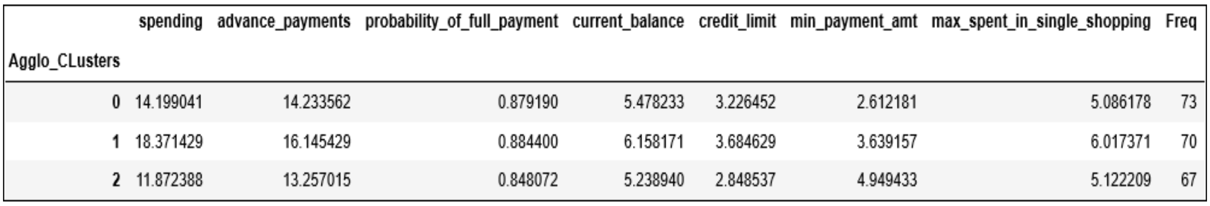


For a more focused view of Probability_of_full_payment cluster:



Now, let’s perform **Agglomerative Clustering** on the scaled data:

Below is the Frequency distribution as per the agglomerative clustering on the mean values of the dataset:



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
Agglo_CLusters								
0	14.199041	14.233562	0.879190	5.478233	3.226452	2.612181	5.086178	73
1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371	70
2	11.872388	13.257015	0.848072	5.238940	2.848537	4.949433	5.122209	67

From the above two tables: we can see that frequencies of both the tables are the same for same clusters.

As we can observe from the 3 cluster segmentations, the customers under the high spenders cluster have higher valuations and probabilities across the various criteria mentioned except the min_payment_amt where the customers of the low spenders cluster have a higher bill value amount as their minimum amount that would have to be remitted.

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

K-Means is a non-hierarchical approach to forming good clusters is to prespecify a desired number of clusters, k. The ‘means’ in the K-means refers to averaging of the data; that is, finding the centroid.

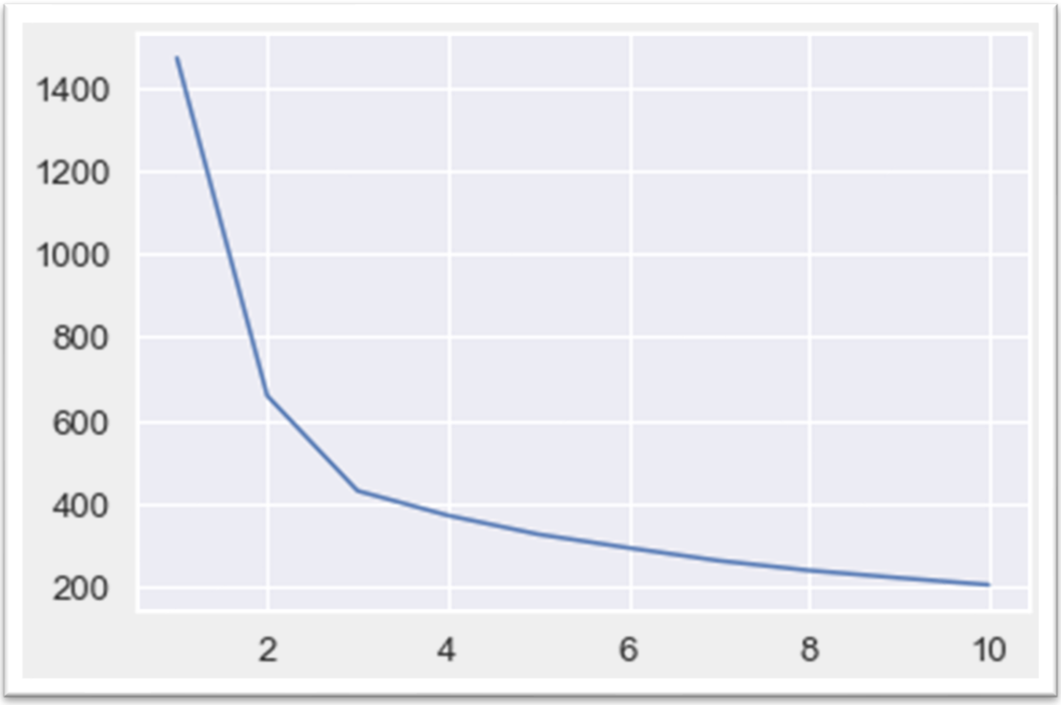
In this method the partitions are made such that non-overlapping groups having no hierarchical relationships between themselves.

K-means clustering is widely used in large dataset applications.

Using SKlearn’s K-Means package, we fit the scaled data, calculating the inertia and then total within-cluster sum of squares. (WSS)

Now with the help of **Elbow Curve**, for a given number of clusters, the total within-cluster sum of squares (WCSS) is computed.

That value of k is chosen to be optimum, where addition of one more cluster does not lower the value of total WSS appreciably.



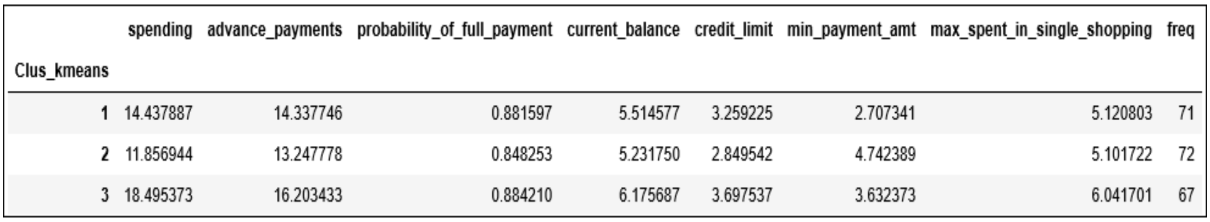
We go with 3 cluster segmentation as per our business recommendation as we see an elbow at cluster number = 3 after which the scree plot seems redundant.

Silhouette Score

This method measures how tightly the observations are clustered and the average distance between clusters. For each observation a silhouette score is constructed which is a function of the average distance between the point and all other points in the cluster to which it belongs, and the distance between the point and all other points in all other clusters, that it does not belong to. The maximum value of the statistic indicates the optimum value of k.

However, the Silhouette Score of 2 clusters was more appropriate, however, objective of this clustering effort is to devise a suitable recommendation system. It may not be practical to manage a very low number of tailor-made recommendations. Therefore, Cluster number = 3 serves the purpose of our requirement to produce valuable insights.

Upon performing Non-Hierarchical clustering on scaled dataset, we obtain mean values within 3 cluster formations as follows:



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	freq
Clus_kmeans								
1	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803	71
2	11.856944	13.247778	0.848253	5.231750	2.849542	4.742389	5.101722	72
3	18.495373	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701	67

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

Business Recommendations opting Number of clusters as **Three**.

Group 1 : High Spending Group

- The offerings such as, higher reward points on a higher probability can increase their spending capacity.
- Adding an option of no cost EMI as a promotional scheme with bank's tied up brands, can be a great motivator for this group.
- The segmentation of maximum max_spent_in_single_shopping is the highest of this group, hence, the discounts offered or attractive offers on the next transactions with full payments upfront.
- Periodic assessment and increase of credit limits
- The preferential customer treatment which might lead to higher spending habits
- Since there is a clear indication that the customers of this category are financially stable, interesting loan schemes exclusively for them could be planned.
- Collaborations with high end luxury brands and accessories would lead to higher one-time maximum spending.

Group 2 : Low Spending Group

- We can spend some time analysing the brands and utilities this segment spends its most amount on and provide discounts and offers on the credit card usage accordingly.
- Customers of this segment will have to be given timely reminders on payments so that the due dates of the billing cycles are not missed.
- Small-scale campaigns could be run providing the customers of this segment attractive offers for early payments which would improve the rate of payment received and result in lesser default rates.

Group 3 : Medium Spending Group

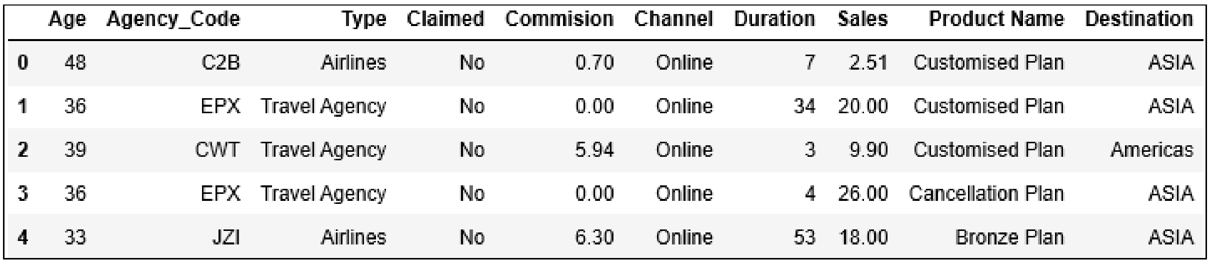
- The customers of this segmentation cluster are suggested to be the target customers with highest potential as there is consistent maintenance of a higher credit score which results in timely payments of their bills.
- The customers of this category can have an increased credit limit raised and monitored periodically and have significantly marginalised interest rates keeping RBI guidelines in mind.
- The advertisement and promotion of premium cards or loyalty cards of specific brand collaborated partnerships would lead to increase in the transactional values over an extended period.
- Once the above-mentioned credit limits are enhanced, the result would be an automatic increase in spending habits across the premium partners in e-commerce, travel portals, airlines & hotels.

Problem Statement 2

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?

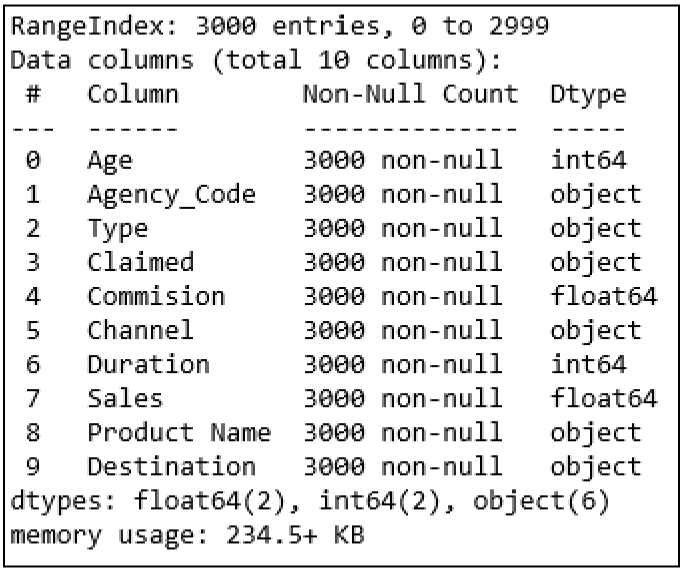
After importing the necessary libraries and data in the python notebook, below is the top 5 rows of the data.



	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

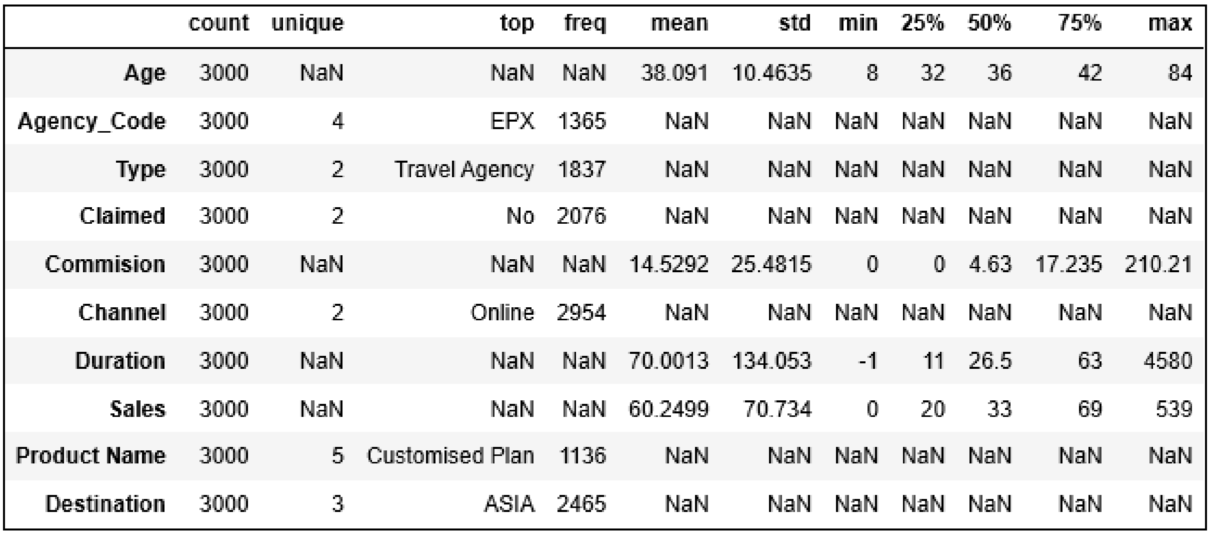
In the below image, we can see that there are no null values in the data. 2 of the 10 variables are of Data type Float, 2 variables are of Data type Interger and the remaining 6 are of Object Data type.

The shape of the data is (3000, 10)



```
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
```

Descriptive Analysis:



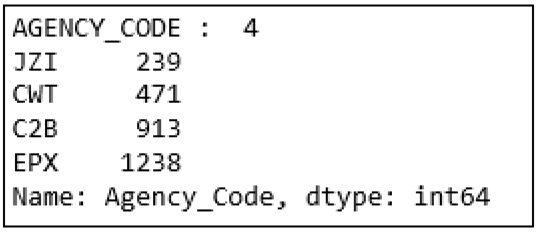
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	3000	NaN	NaN	NaN	38.091	10.4635	8	32	36	42	84
Agency_Code	3000	4	EPX	1365	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Type	3000	2	Travel Agency	1837	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Claimed	3000	2	No	2076	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Commision	3000	NaN	NaN	NaN	14.5292	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

- For Object data type variables like, Agency_code, Type, Claimed, Channel, Product Name, and Destination, there are very less unique values.
- The topmost frequent value of:
    - Agency_code is **EPX** with a frequency of 1365
    - Type is **Travel Agency** with a frequency of 1837
    - Claimed is **No** with a frequency of 2076
    - Channel is **Online** with a frequency of 2954
    - Product Name is **Customised Plan** with a frequency of 1136
    - Destination is **ASIA** with a frequency of 2465
- For the float and integers data type values like: Age, Commision, Duration and Sales the difference between its 75th percentile and Max value is very large, indicating there will be large number of outliers in the data.

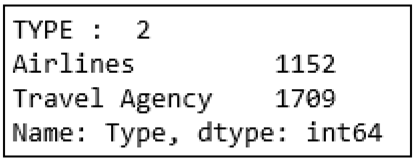
After this, let’s check if we have any duplicates in the data; We have 139 duplicate rows in the data, but since the count of our dataset is 3000 and 139 is even less than 5% of the data we can remove these duplicate values to optimise our results.

Now the shape of the data is (2861, 10)

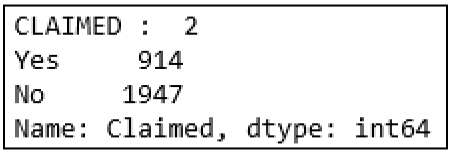
Then we analysed the Object data type variables for their count of unique values:



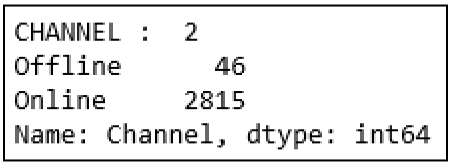
```
AGENCY_CODE :  4
JZI      239
CWT      471
C2B      913
EPX     1238
Name: Agency_Code, dtype: int64
```



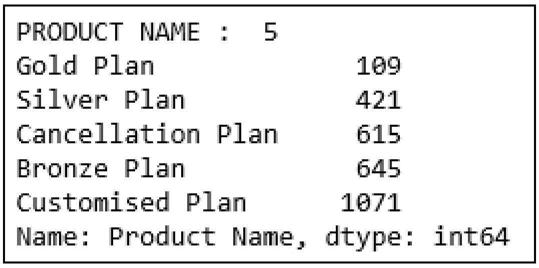
```
TYPE :  2
Airlines         1152
Travel Agency    1709
Name: Type, dtype: int64
```



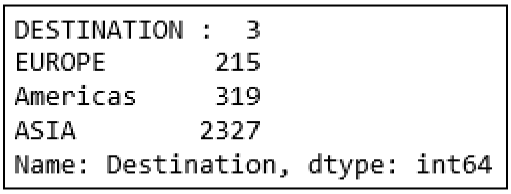
```
CLAIMED :  2
Yes     914
No     1947
Name: Claimed, dtype: int64
```



```
CHANNEL :  2
Offline      46
Online     2815
Name: Channel, dtype: int64
```

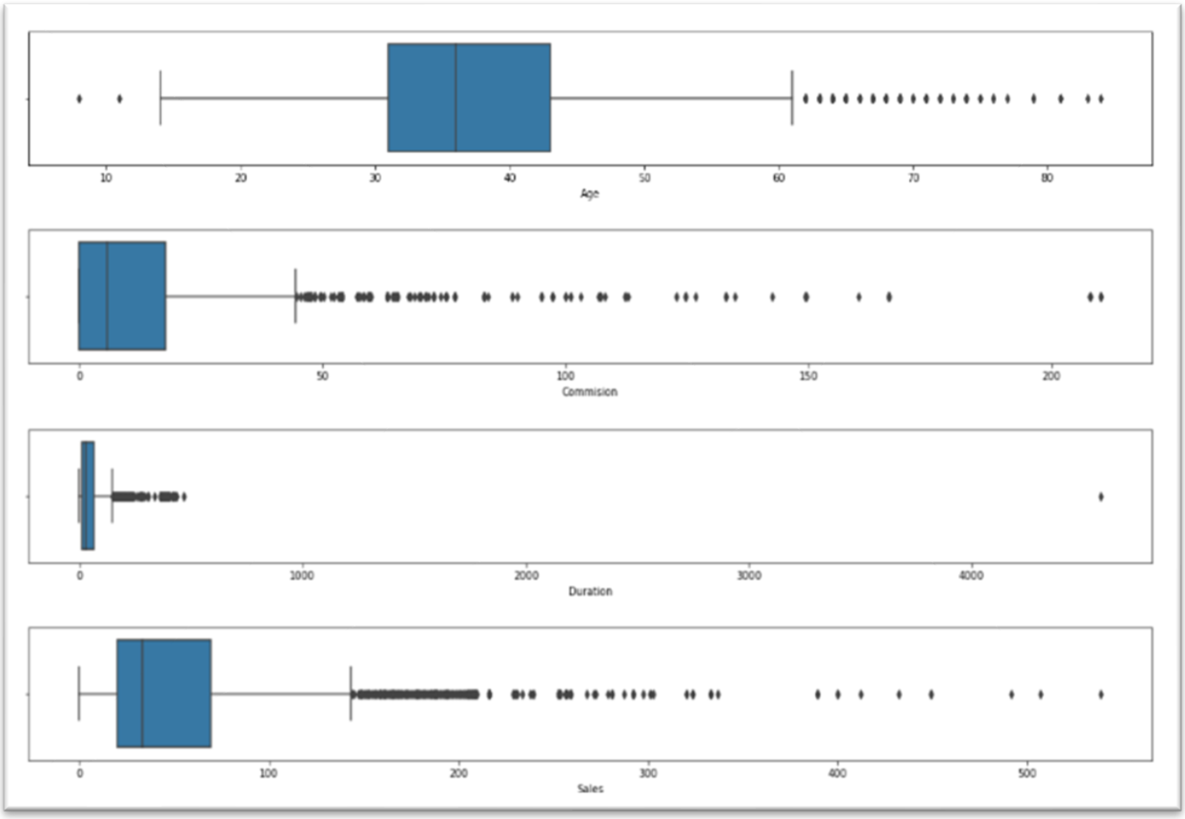


```
PRODUCT NAME :  5
Gold Plan              109
Silver Plan            421
Cancellation Plan      615
Bronze Plan            645
Customised Plan       1071
Name: Product Name, dtype: int64
```



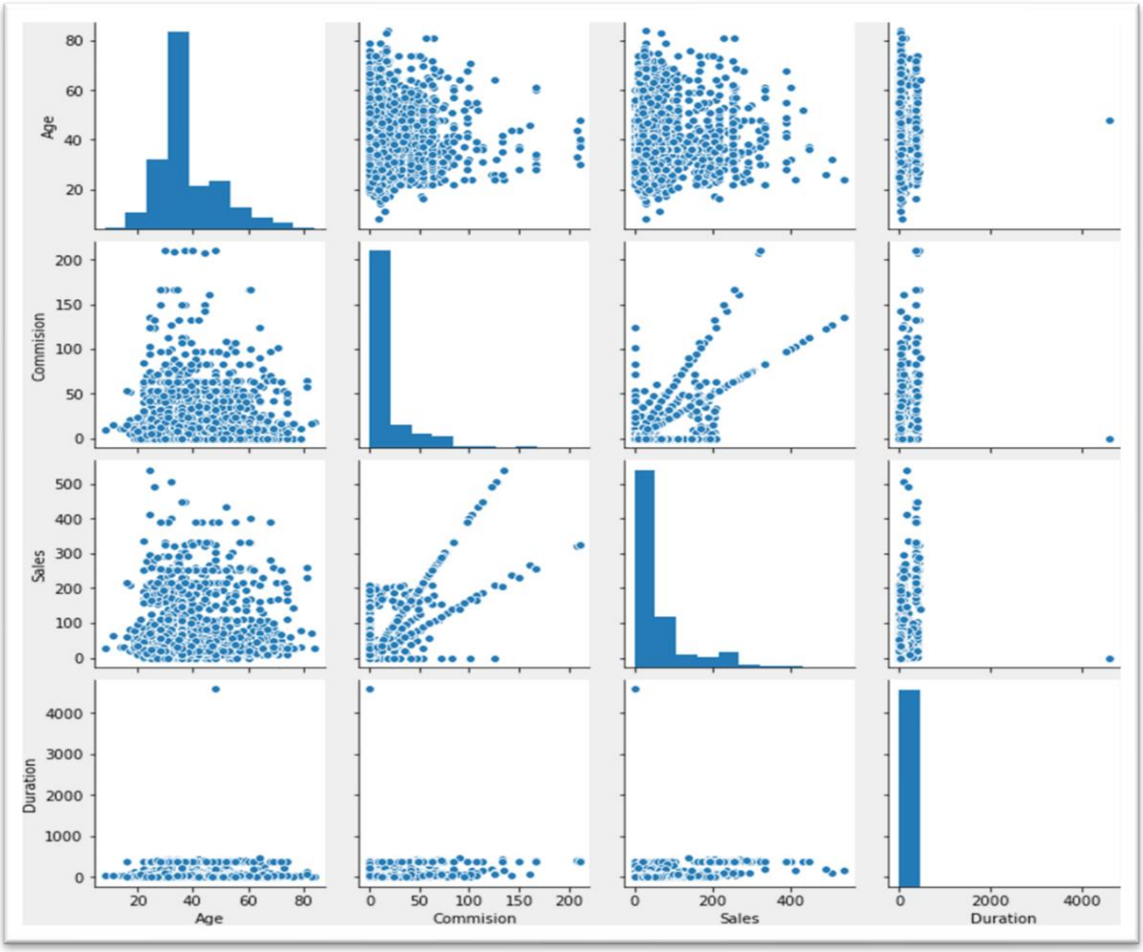
```
DESTINATION :  3
EUROPE       215
Americas     319
ASIA        2327
Name: Destination, dtype: int64
```

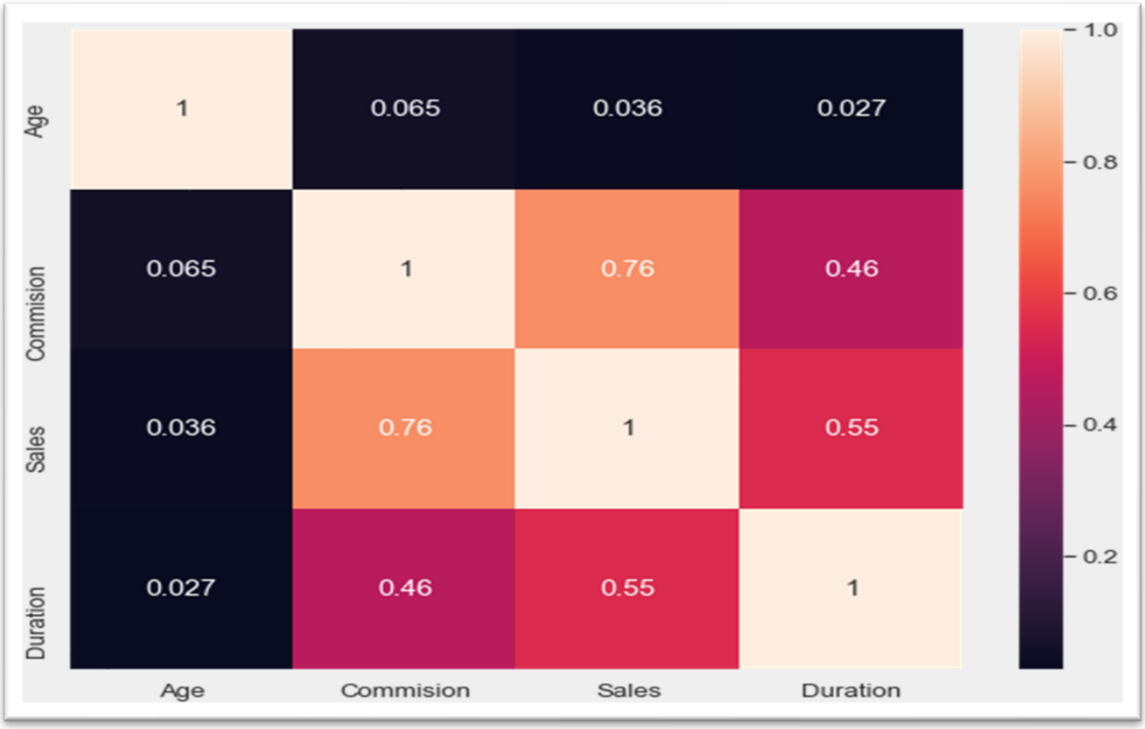
Let’s analyse the outliers of the numeric data type variables.



There are multiple outliers in the data. However, since the outliers do not directly impact any of the three models, treating the outliers is not necessary at this stage.

We will have a look at the variable correlation next:

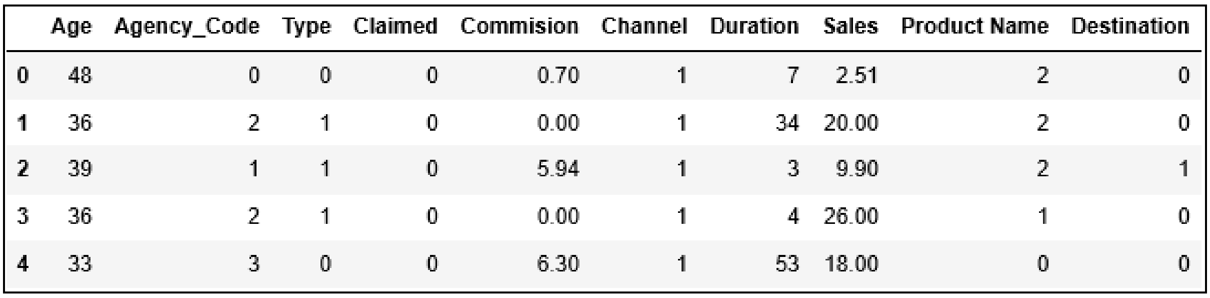




There is no major correlation in any of the two variables but in comparison, Sales and Commision has a correlation of 0.76 which is high in comparison with other variables.

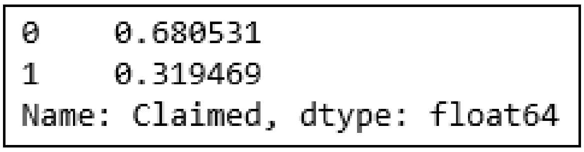
On the next step, we are changing the data type of Object variables into Categorical data. After which, the all the data types of the data are either Integer or Float.

This is how the data looks after Conversion:



	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	0	0	0	0.70	1	7	2.51	2	0
1	36	2	1	0	0.00	1	34	20.00	2	0
2	39	1	1	0	5.94	1	3	9.90	2	1
3	36	2	1	0	0.00	1	4	26.00	1	0
4	33	3	0	0	6.30	1	53	18.00	0	0

Now let’s see the Proportions of 0s and 1st of our target variable:



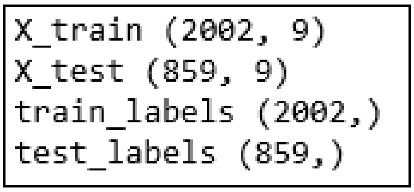
```
0    0.680531
1    0.319469
Name: Claimed, dtype: float64
```

There is no issue of class imbalance here as we have reasonable proportions in both the classes.

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

Firstly, splitting the data into Train and Test data.

Below is the data shape:



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

CART MODEL:

CART is a Binary Decision Tree model. I have used Gini Index as its Criteria. It is an attribute that Maximizes the reduction in impurity is chosen as the Splitting Attribute.

Using the Decision Tree Classifier and the Grid search method, I have identified the best grid:

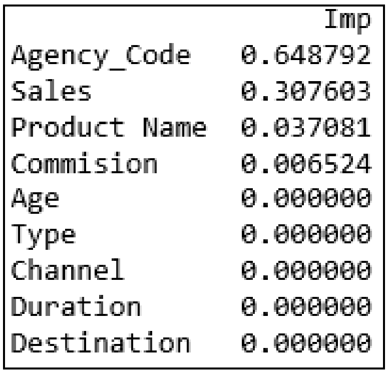
'criterion': 'gini',

'max_depth': 4,

'min_samples_leaf': 25,

'min_samples_split': 300

After looking at the decision tree, we extracted the variable importance shown below:



```
                   Imp
Agency_Code   0.648792
Sales         0.307603
Product Name  0.037081
Commision     0.006524
Age           0.000000
Type          0.000000
Channel       0.000000
Duration      0.000000
Destination   0.000000
```

As per the above extract, Agency_code is the most important variable in the dataset, followed by Sales and Product Name.

Commision has comparatively very less importance, however Age, Type, Channel, Duration and Destination have no importance in the model building.

RANDOM FOREST:

Random Forest Consists of many individual decision trees that operate as an ensemble. Each tree in the random forest spits out a class prediction. Class with most votes becomes model's prediction.

Using the random forest classifier and grid search function, we identified the best grid parameters:

'max_depth': 15,

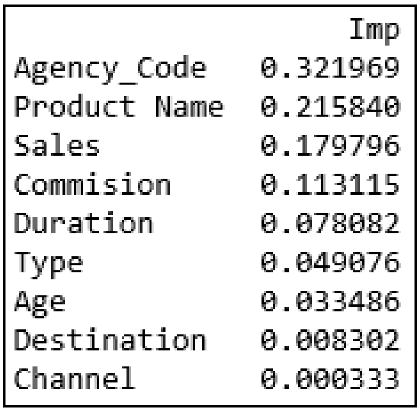
'max_features': 4,

'min_samples_leaf': 25,

'min_samples_split': 25,

'n_estimators': 200

We extracted the variable importance as per RF:



```
                    Imp
Agency_Code    0.321969
Product Name   0.215840
Sales          0.179796
Commision      0.113115
Duration       0.078082
Type           0.049076
Age            0.033486
Destination    0.008302
Channel        0.000333
```

Like CART, for RF as well Agency_code has the most importance in the model, however Sales and Product Name exchanged places. In this model, each of the variable pays a role in model building at some importance level but Channel Variable has the lowest importance of them all.

NEURAL NETWORK CLASSIFIER:

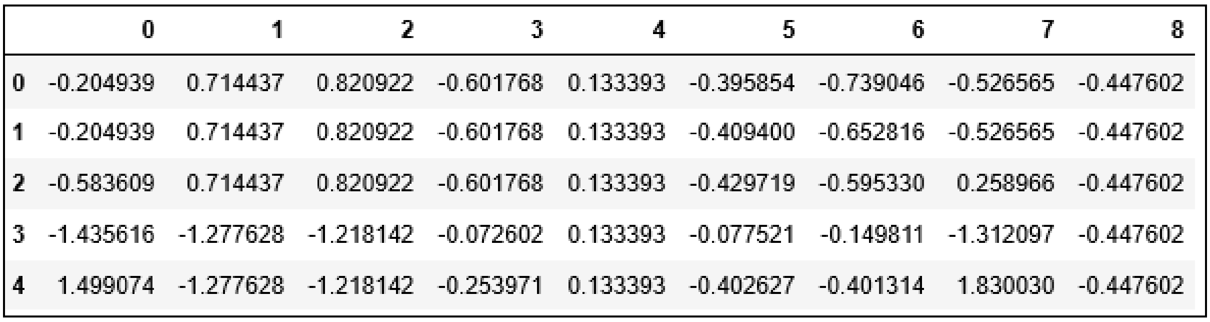
NN is made of layers with many interconnected nodes(neurons). There are three main layers specifically.

- Input Layer
- Hidden Layer
- Output Layer

Hidden Layer can be one or more.

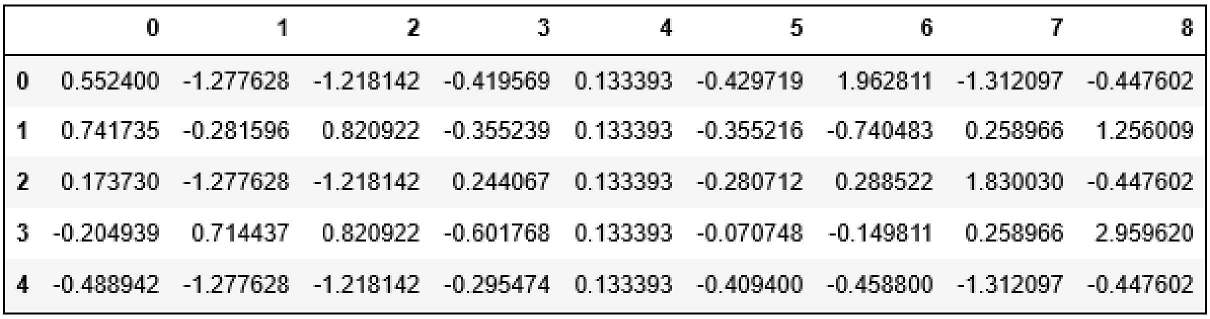
Using the Standard scaler, the raw data was first scaled.

Below is the head of the Train scaled data:



	0	1	2	3	4	5	6	7	8
0	-0.204939	0.714437	0.820922	-0.601768	0.133393	-0.395854	-0.739046	-0.526565	-0.447602
1	-0.204939	0.714437	0.820922	-0.601768	0.133393	-0.409400	-0.652816	-0.526565	-0.447602
2	-0.583609	0.714437	0.820922	-0.601768	0.133393	-0.429719	-0.595330	0.258966	-0.447602
3	-1.435616	-1.277628	-1.218142	-0.072602	0.133393	-0.077521	-0.149811	-1.312097	-0.447602
4	1.499074	-1.277628	-1.218142	-0.253971	0.133393	-0.402627	-0.401314	1.830030	-0.447602

Head of Test Scaled data:



	0	1	2	3	4	5	6	7	8
0	0.552400	-1.277628	-1.218142	-0.419569	0.133393	-0.429719	1.962811	-1.312097	-0.447602
1	0.741735	-0.281596	0.820922	-0.355239	0.133393	-0.355216	-0.740483	0.258966	1.256009
2	0.173730	-1.277628	-1.218142	0.244067	0.133393	-0.280712	0.288522	1.830030	-0.447602
3	-0.204939	0.714437	0.820922	-0.601768	0.133393	-0.070748	-0.149811	0.258966	2.959620
4	-0.488942	-1.277628	-1.218142	-0.295474	0.133393	-0.409400	-0.458800	-1.312097	-0.447602

Using the MLP Classifier and grid search, I identified the best grid:

'hidden_layer_sizes': 8,

'max_iter': 2500,

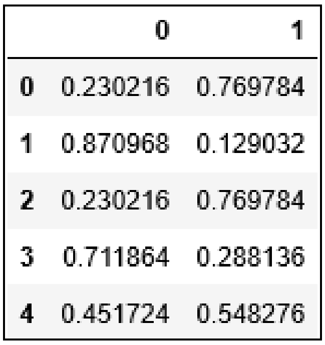
'solver': 'adam',

'tol': 0.001

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 Performance Matrix:

After predicting the test and train data, below is the head of ytest_predict_prob:

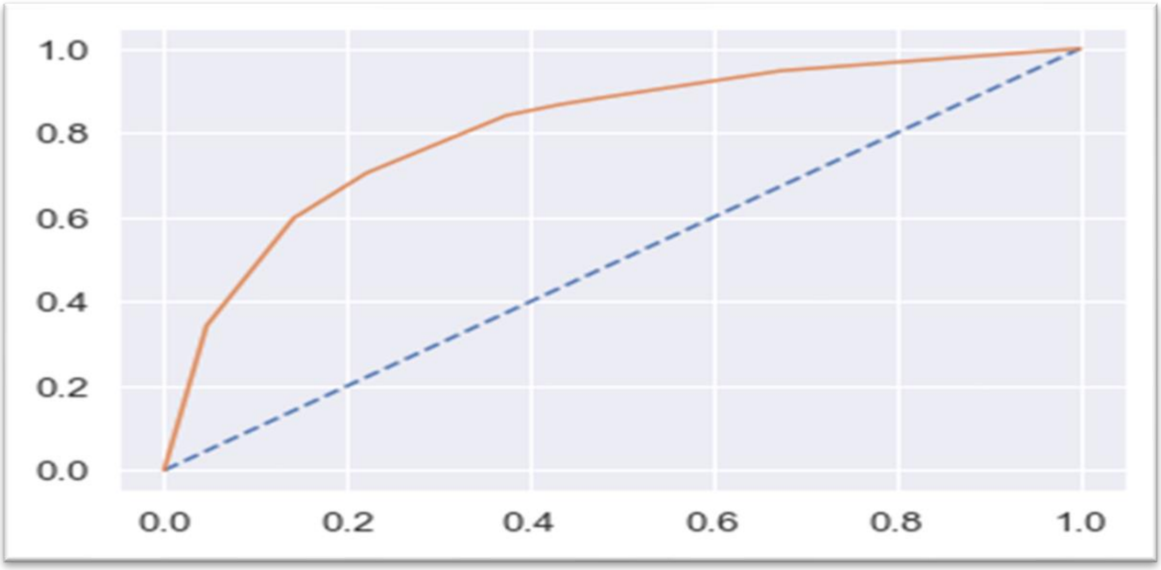


	0	1
0	0.230216	0.769784
1	0.870968	0.129032
2	0.230216	0.769784
3	0.711864	0.288136
4	0.451724	0.548276

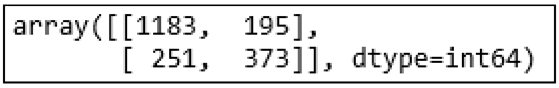
Training data

AUC and ROC curve of CART:

AUC: **0.809**



Confusion Matrix on CART

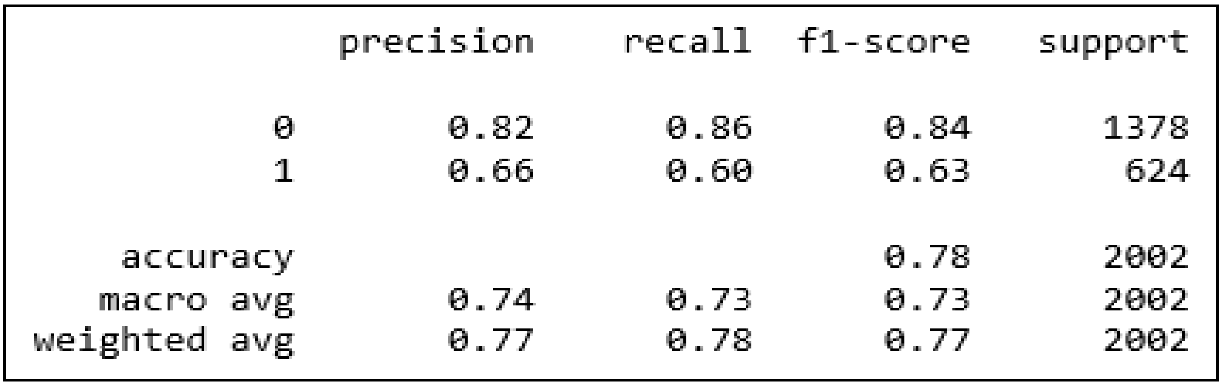


```
array([[1183,  195],
       [ 251,  373]], dtype=int64)
```

Data Accuracy

0.7772227772227772

Classification Report

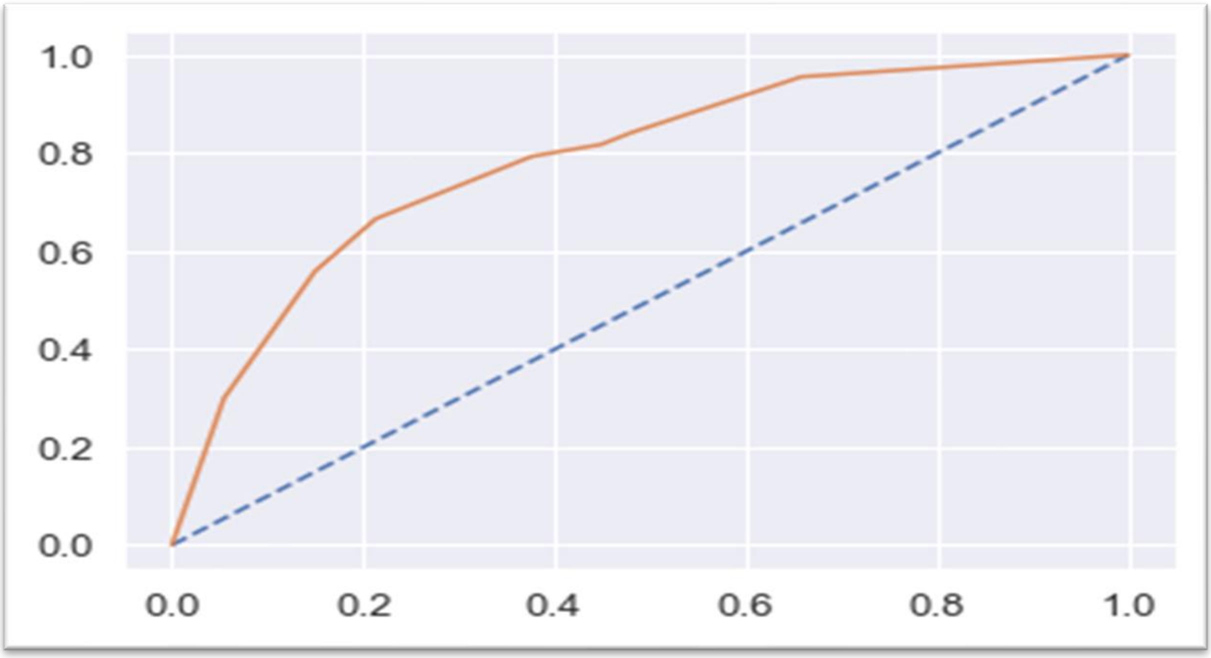


```
              precision    recall  f1-score   support

           0       0.82      0.86      0.84      1378
           1       0.66      0.60      0.63       624

    accuracy                           0.78      2002
   macro avg       0.74      0.73      0.73      2002
weighted avg       0.77      0.78      0.77      2002
```

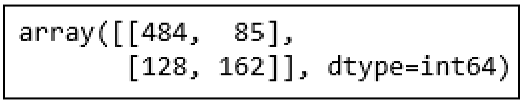
Testing data

AUC and ROC curve of CART:

AUC: **0.786**



Confusion Matrix on CART

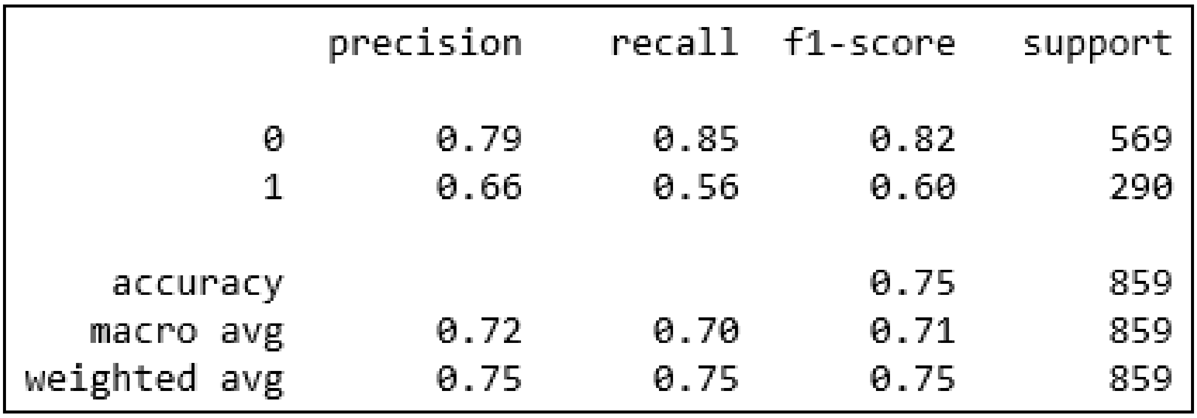


```
array([[484,  85],
       [128, 162]], dtype=int64)
```

Data Accuracy

0.7520372526193247

Classification Report



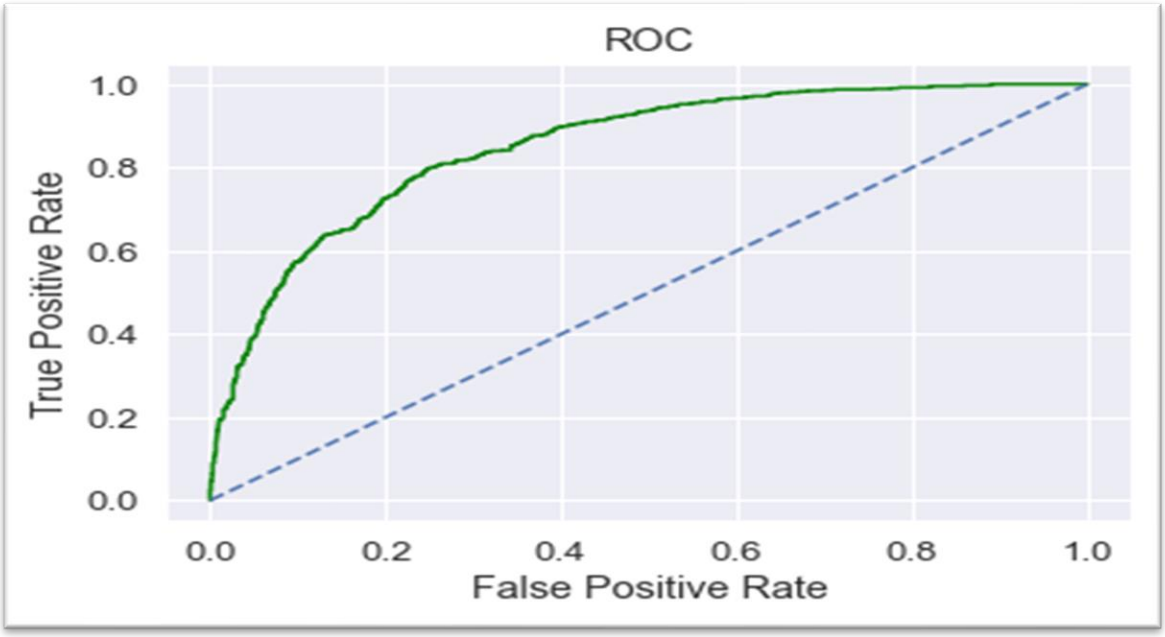
```
              precision    recall  f1-score   support

           0       0.79      0.85      0.82       569
           1       0.66      0.56      0.60       290

    accuracy                           0.75       859
   macro avg       0.72      0.70      0.71       859
weighted avg       0.75      0.75      0.75       859
```

RANDOM FOREST Performance Matrix:

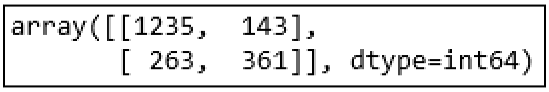
Training data

AUC and ROC curve of RF:

AUC: **0.8500044192624019**



Confusion Matrix on RF

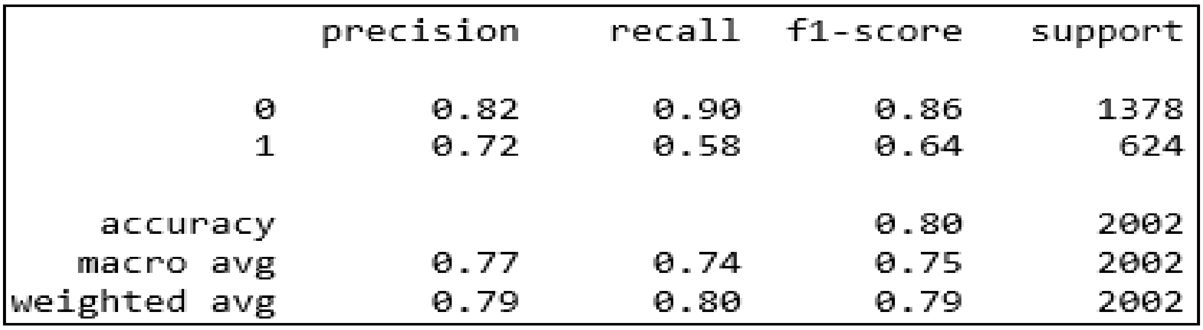


```
array([[1235,  143],
       [ 263,  361]], dtype=int64)
```

Data Accuracy

0.7972027972027972

Classification Report

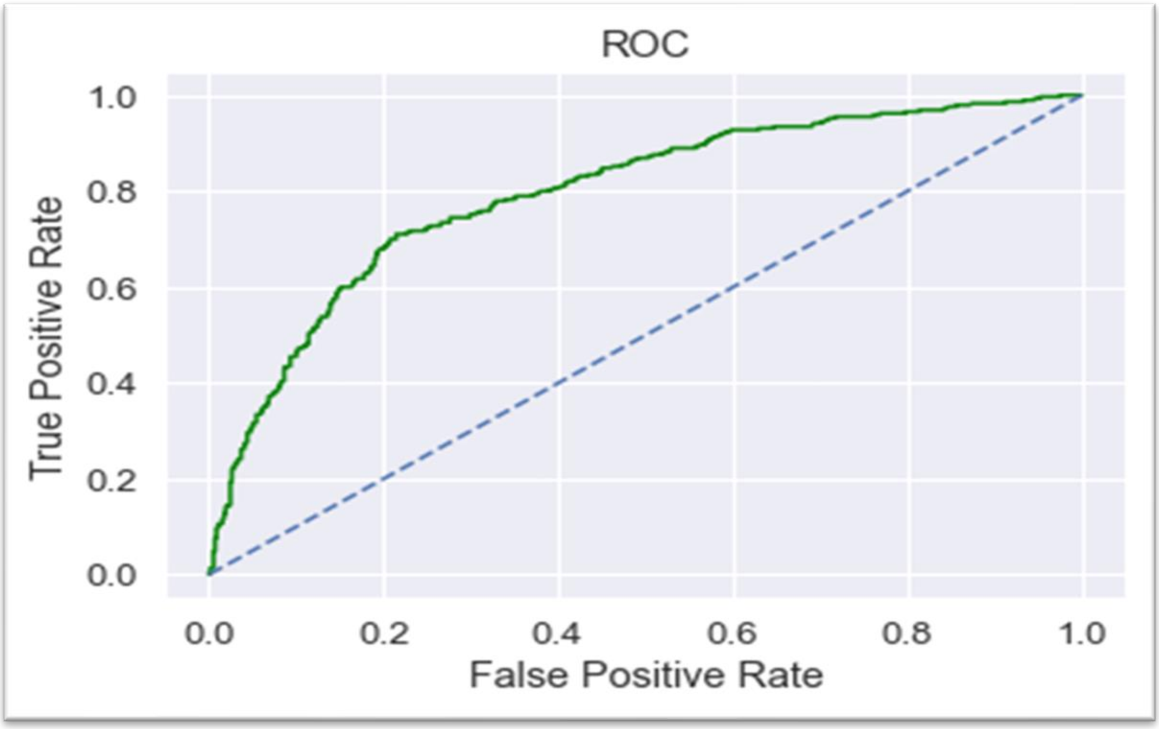


```
              precision    recall  f1-score   support

           0       0.82      0.90      0.86      1378
           1       0.72      0.58      0.64       624

    accuracy                           0.80      2002
   macro avg       0.77      0.74      0.75      2002
weighted avg       0.79      0.80      0.79      2002
```

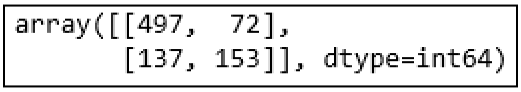
Testing data

AUC and ROC curve of RF:

AUC: **0.7952336222047149**



Confusion Matrix on RF

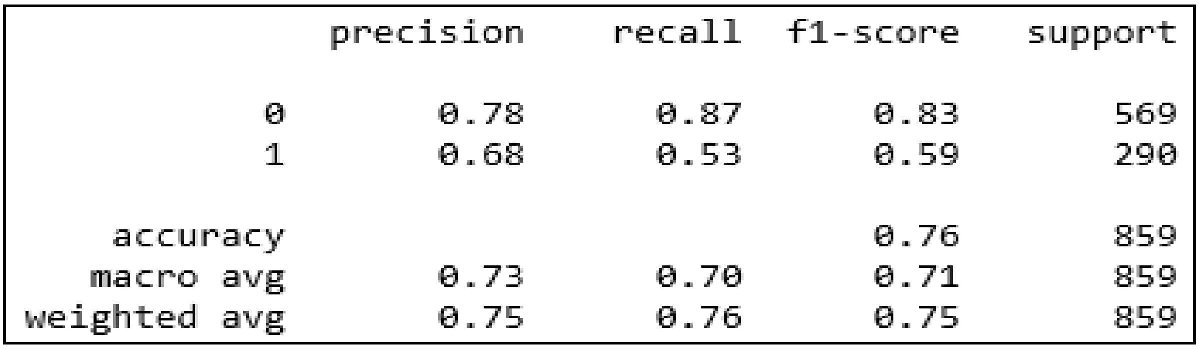


```
array([[497,  72],
       [137, 153]], dtype=int64)
```

Data Accuracy

0.7566938300349243

Classification Report on Test Data



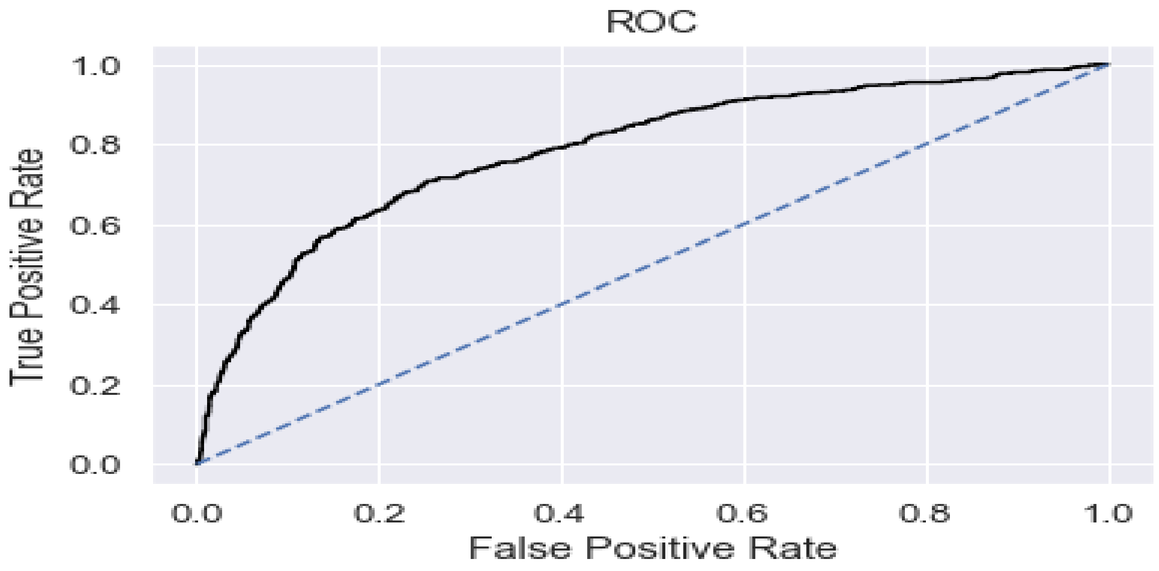
```
              precision    recall  f1-score   support

           0       0.78      0.87      0.83       569
           1       0.68      0.53      0.59       290

    accuracy                           0.76       859
   macro avg       0.73      0.70      0.71       859
weighted avg       0.75      0.76      0.75       859
```

NEURAL NETWORK CLASSIFIER Performance Matrix:

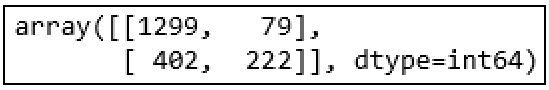
<u>Training data</u>

AUC and ROC curve of NN:

AUC: **0.7859838441070299**



Confusion Matrix on NN

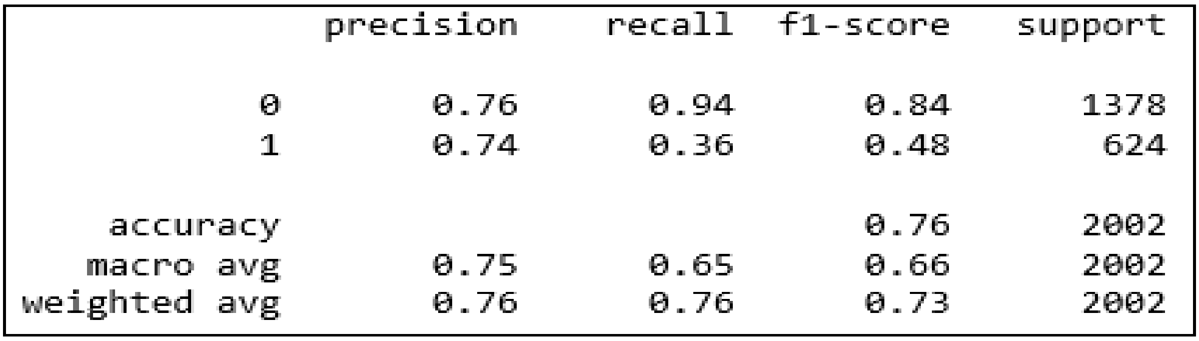


```
array([[1299,   79],
       [ 402,  222]], dtype=int64)
```

Data Accuracy

0.7597402597402597

Classification Report

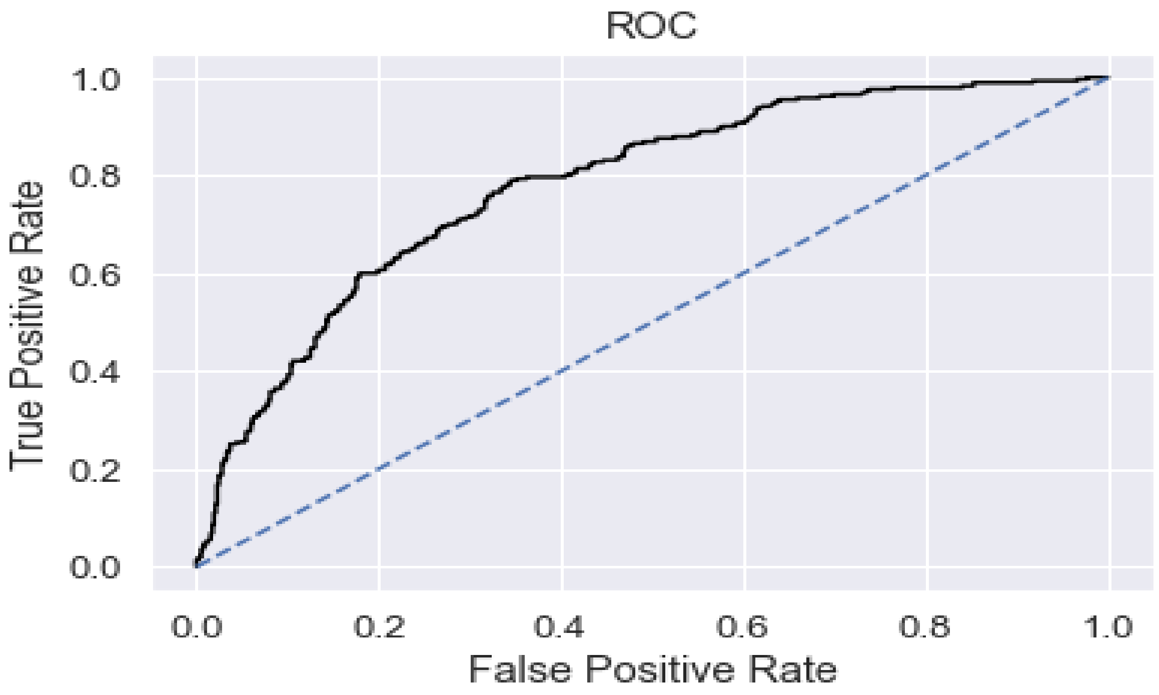


```
              precision    recall  f1-score   support

           0       0.76      0.94      0.84      1378
           1       0.74      0.36      0.48       624

    accuracy                           0.76      2002
   macro avg       0.75      0.65      0.66      2002
weighted avg       0.76      0.76      0.73      2002
```

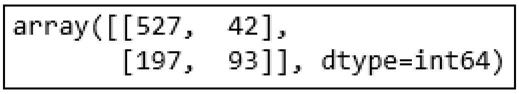
<u>Testing data</u>

AUC and ROC curve of NN:

AUC: **0.7810556935943276**



Confusion Matrix on NN

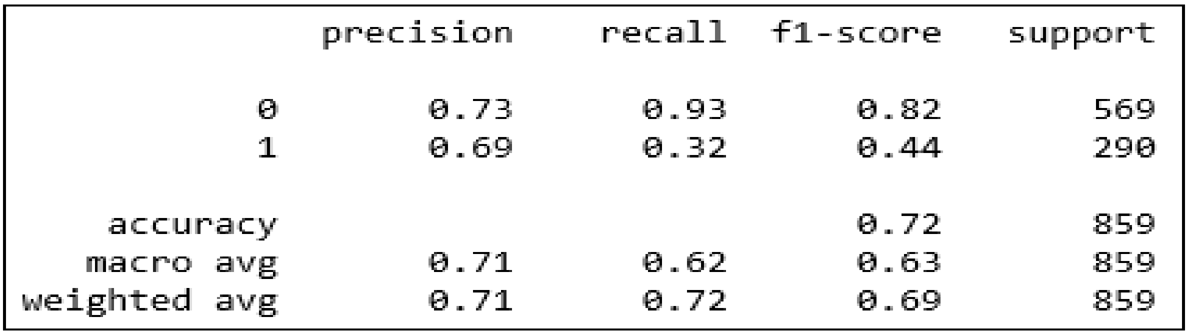


```
array([[527,  42],
       [197,  93]], dtype=int64)
```

Data Accuracy

0.7217694994179278

Classification Report

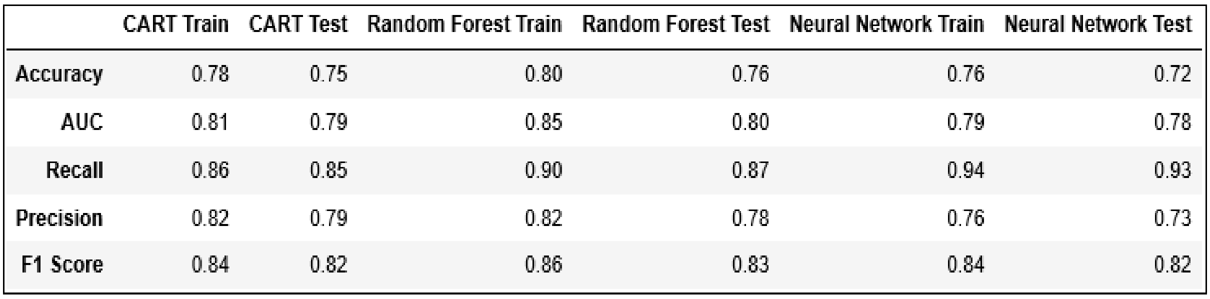


```
              precision    recall  f1-score   support

           0       0.73      0.93      0.82       569
           1       0.69      0.32      0.44       290

    accuracy                           0.72       859
   macro avg       0.71      0.62      0.63       859
weighted avg       0.71      0.72      0.69       859
```

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

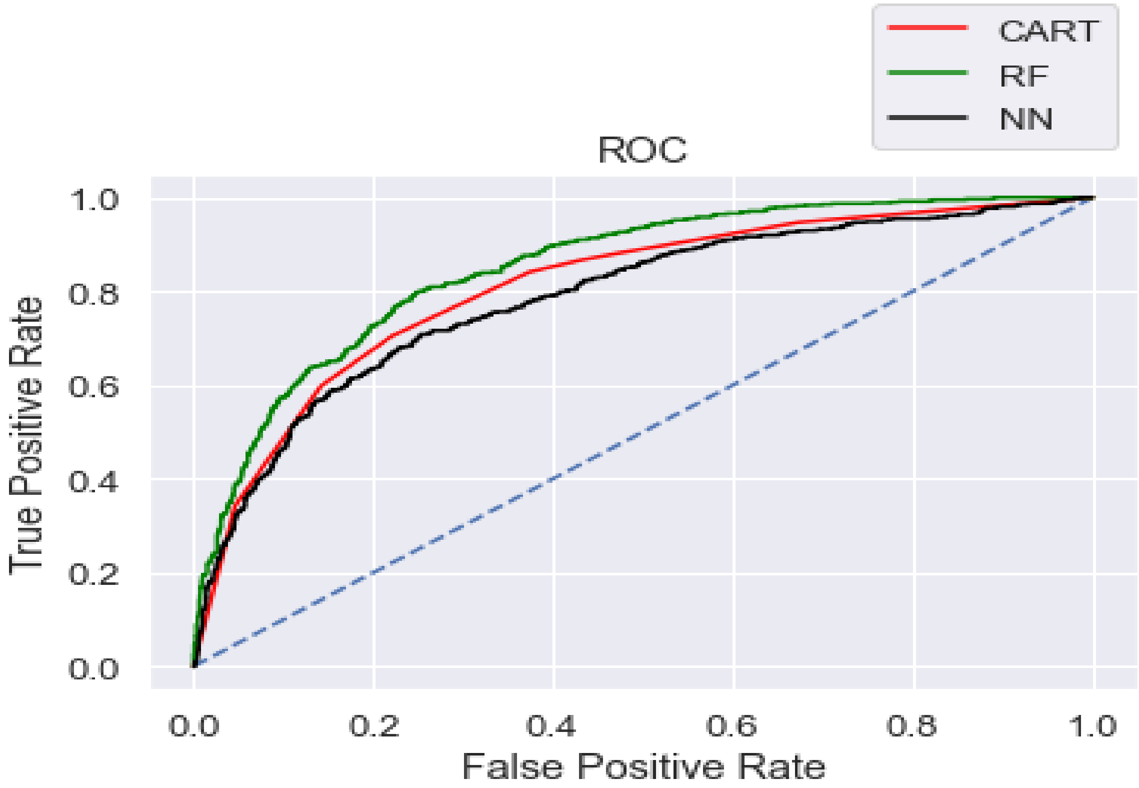
Below we are comparing Accuracy, AUC, Recall, Precision and F1 score of all the models, where Target is 0, i.e. the claimed as NO.

The logic to choose Claimed as NO is that the model is calculating Claimed as No more accurately than Claimed as Yes. Also, this way we will be able to identify using the attributes that which policy will not be claimed with more than approx. 75% accuracy.

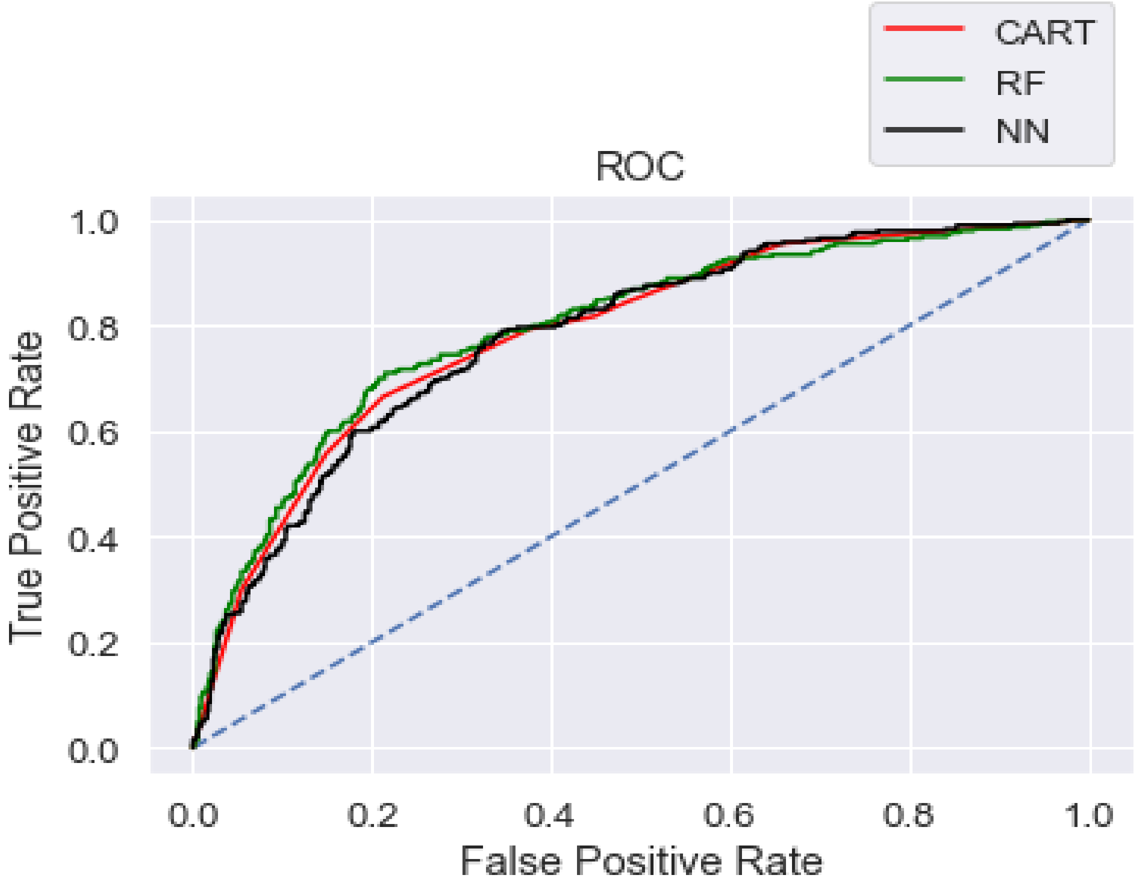


	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.78	0.75	0.80	0.76	0.76	0.72
AUC	0.81	0.79	0.85	0.80	0.79	0.78
Recall	0.86	0.85	0.90	0.87	0.94	0.93
Precision	0.82	0.79	0.82	0.78	0.76	0.73
F1 Score	0.84	0.82	0.86	0.83	0.84	0.82

ROC curve of the **Train data** of all the three models:



ROC curve of the **Test data** of all the three models:



Out of the 3 models, **Random Forest** has slightly better performance than the Cart and Neural network model.

Overall, all the 3 models are reasonably stable enough to be used for making any future predictions. From Random Forest Model, the variable change is found to be the most useful feature amongst all other features for predicting if a person will claim or not. If change is NO, then those policies have more chances of getting claimed.

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

- It seems that all the models show high accuracy in predicting the customers who will not claim for tour insurance after performing the supervised learning algorithm.
- As we know this Insurance firm was facing higher claim frequency and this model would certainly help in reducing the ratio.
- Since the variable **Agency code** seem to be the most important factor in deriving the model, therefore I recommend that the insurance company tie up with more Agencies to expand its business.
- To attain the less frequency of claims, they should add certain steps to their policy's terms and conditions that would benefit both customers and company.
- Using this model and customer data this insurance firm can easily pick their profitable customers.
- Team can easily target the customers who will not claim for tour insurance. Once Team receives customer data who falls under NO claim status as per the model, then team needs to build strong relationship with those customers because you only get profit when repeated customer sees loyalty and trust in an organization.
- I believe that the tour insurance company should also increase its varieties for Product Name. For now, they are having Bronze, Cancellation, Customized, Gold and Silver plans, but adding few more to the list will encourage customers to choose the optimum plan which proves to be the successful for them and in return, would lead to less frequency of claims for the company. The same would also result in more sales for the tour insurance company.
- Product plan which has higher commission rate can be recommended to the set customers who will fall under NO claim status.