DATA MINING PROJECT

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

```
<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)
```

All the variables are float.

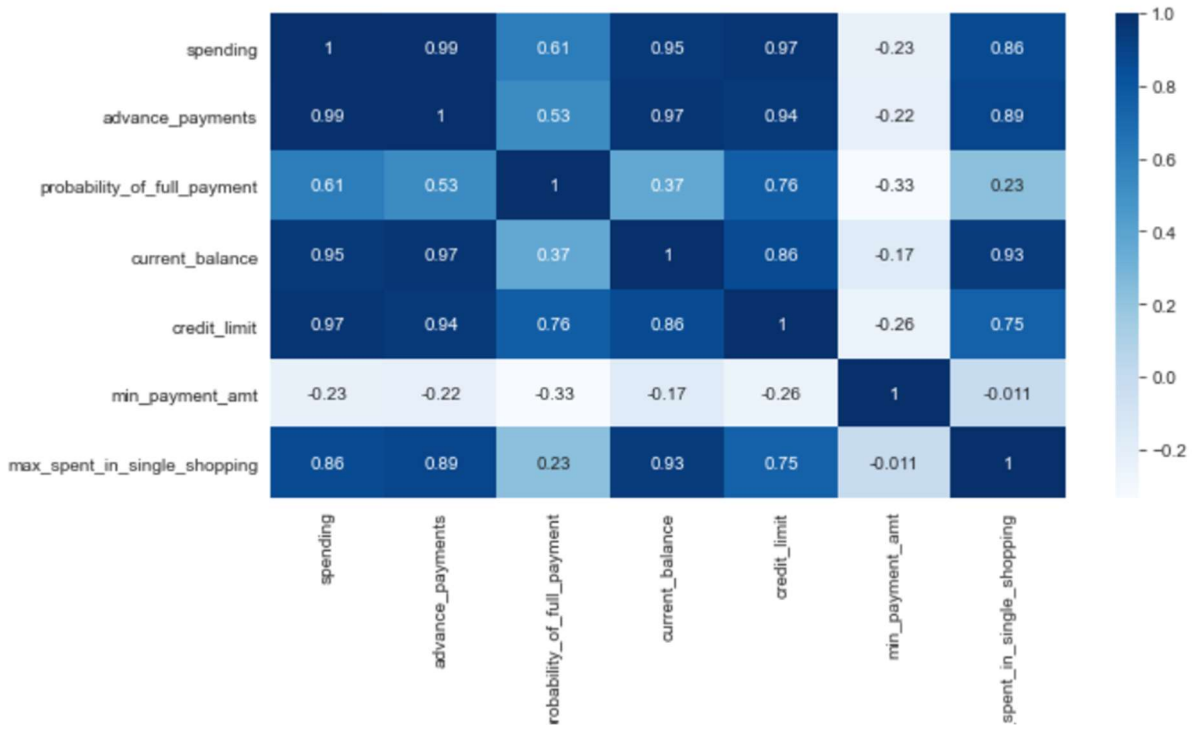
```
spending                      0
advance_payments              0
probability_of_full_payment   0
current_balance               0
credit_limit                  0
min_payment_amt               0
max_spent_in_single_shopping  0
dtype: int64
```

There are no null values in the dataset.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000
mean	14.847524	14.559286	0.870999	5.628533	3.258605	3.700201	5.408071
std	2.909699	1.305959	0.023629	0.443063	0.377714	1.503557	0.491480
min	10.590000	12.410000	0.808100	4.899000	2.630000	0.765100	4.519000
25%	12.270000	13.450000	0.856900	5.262250	2.944000	2.561500	5.045000
50%	14.355000	14.320000	0.873450	5.523500	3.237000	3.599000	5.223000
75%	17.305000	15.715000	0.887775	5.979750	3.561750	4.768750	5.877000
max	21.180000	17.250000	0.918300	6.675000	4.033000	8.456000	6.550000

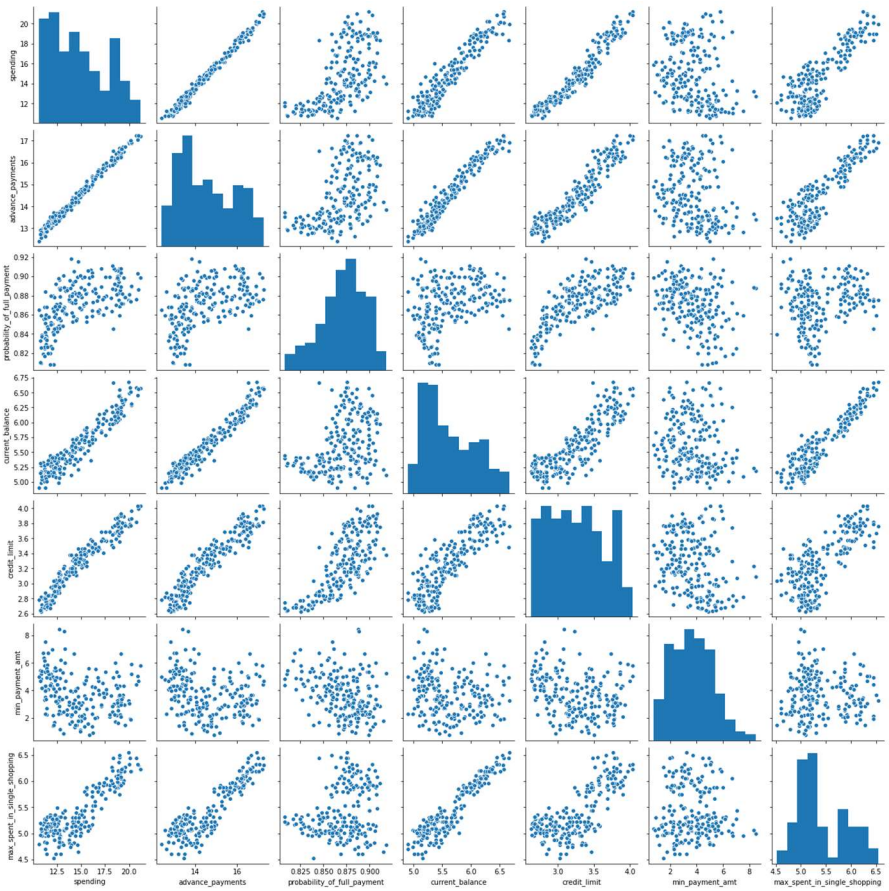
The dataset is not scaled. The above describe function gives us the details of the dataset. We can see spending and advance payment are not in same magnitude as other variable so the data is not scaled.

Correlation of dataset:



Most of the variable are very highly co related. Spending and advance payment is very highly co related. Spending and credit limit is also highly correlated. Spending and current balance is corelated. Advance payment and current balance is correlated.

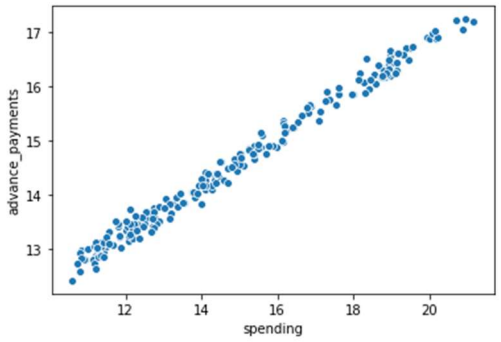
EDA:

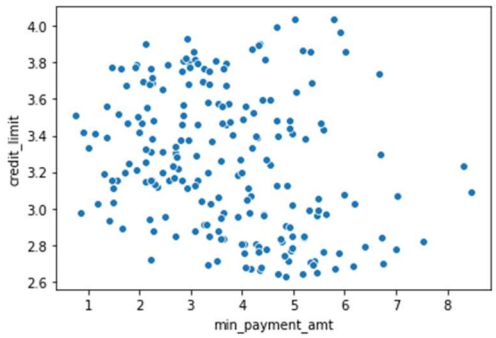


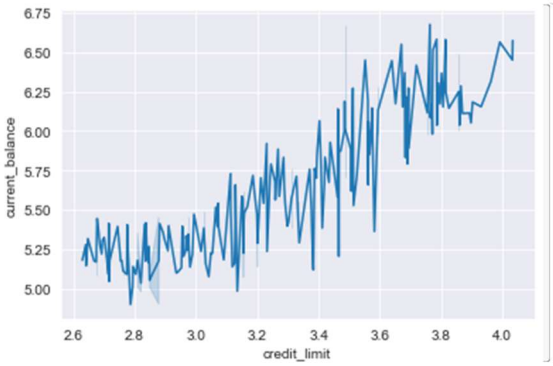
Above we can see the pair plot which shows us the relationship of every variable through a scatterplot and histogram.

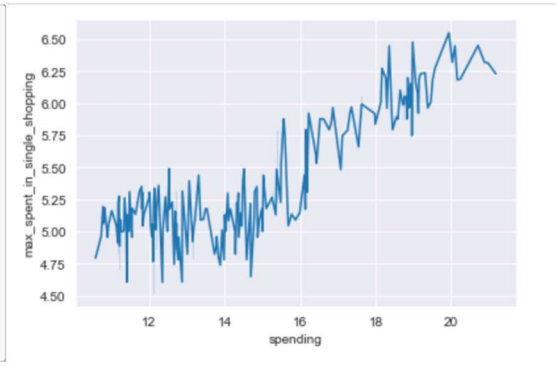
Below is the few analysis that can be assumed from the above pair plot.

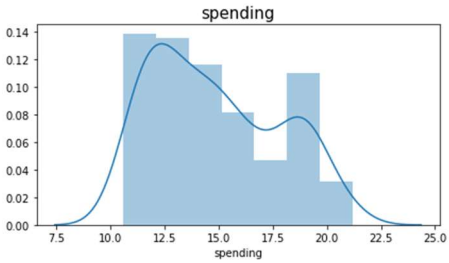
- We can see that spending and advance payment shows an increasing trend. Customers who make more advance payment tend to spend more. (Scatterplot).
- We can also infer that customers who are spending more and mostly spending in single shopping event. These might be the customers who have high credit limit since their one-time spending is high value spending's. (Image4)
- It is also noticed that people who have high credit limit spend more. Most of the customers are given a credit limit between 2.6 and 3
- The average credit limit for customer are 3.25.
- High credit limit is given to customer who make higher advance payments and customer who have highest current balance in the account. (image1)
- We can also see that people who have high credit limit make lesser minimum payment. (image3). Few customers who have less credit limit are making the highest minimum payment.

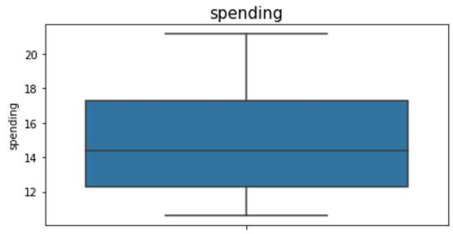


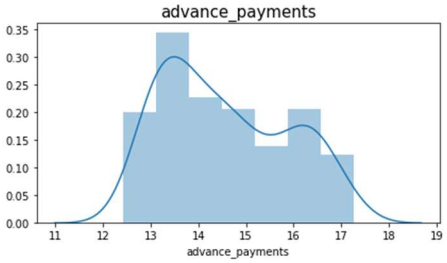




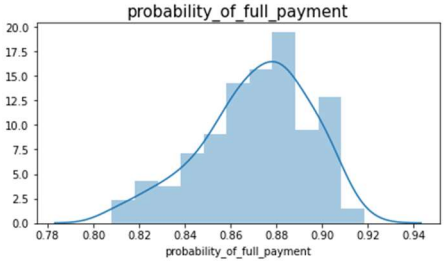


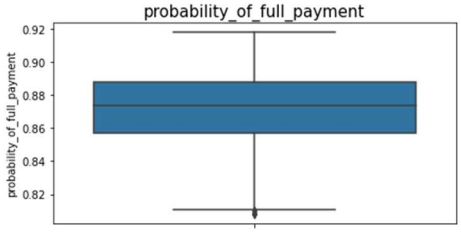

spending
spending

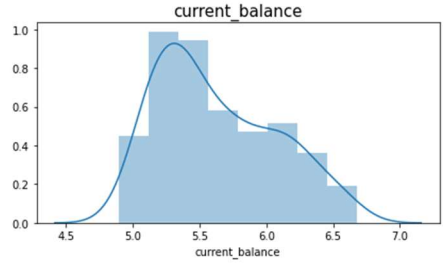

spending
spending

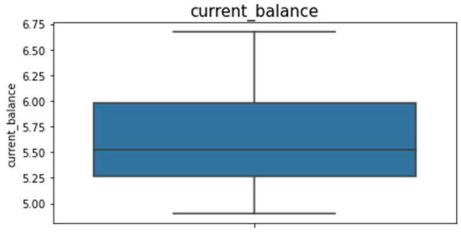

advance_payments
advance_payments

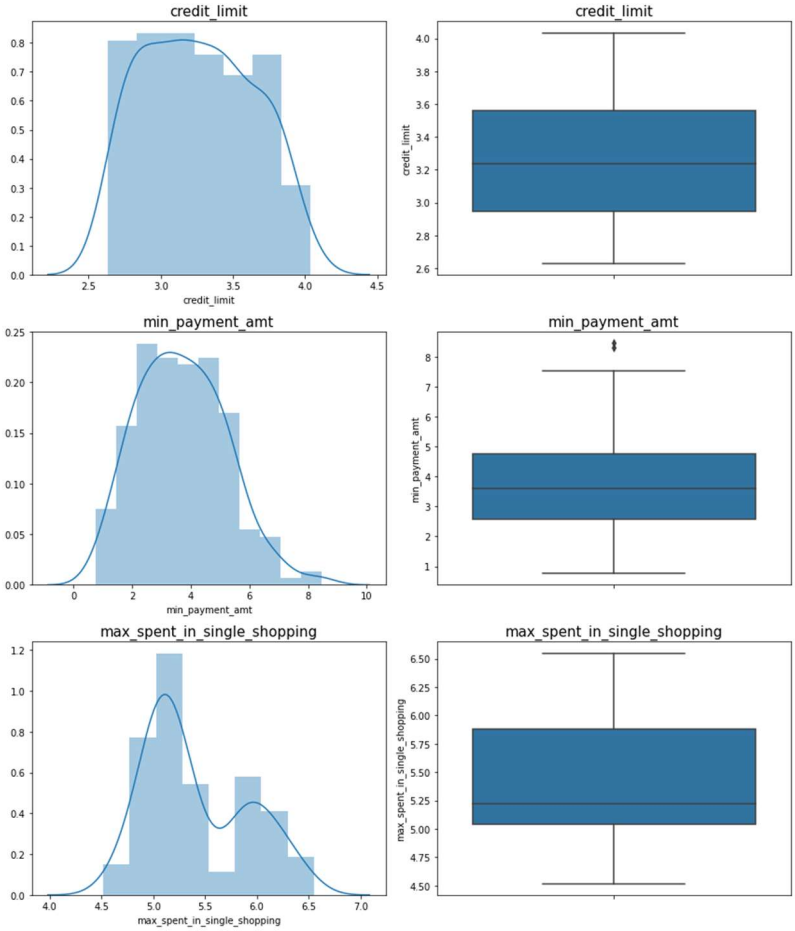

advance_payments
advance_payments


probability_of_full_payment
probability_of_full_payment


probability_of_full_payment
probability_of_full_payment

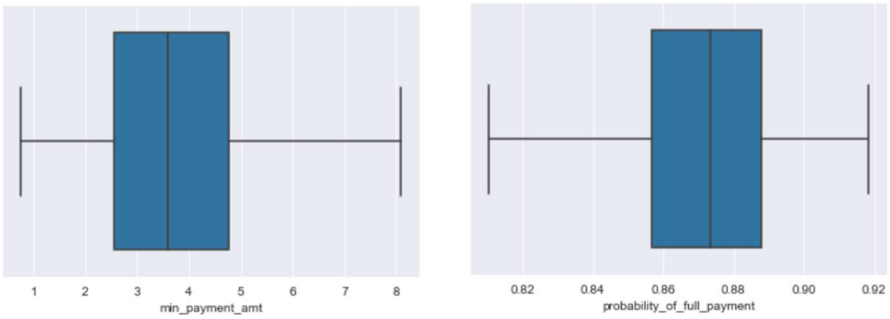

current_balance
current_balance

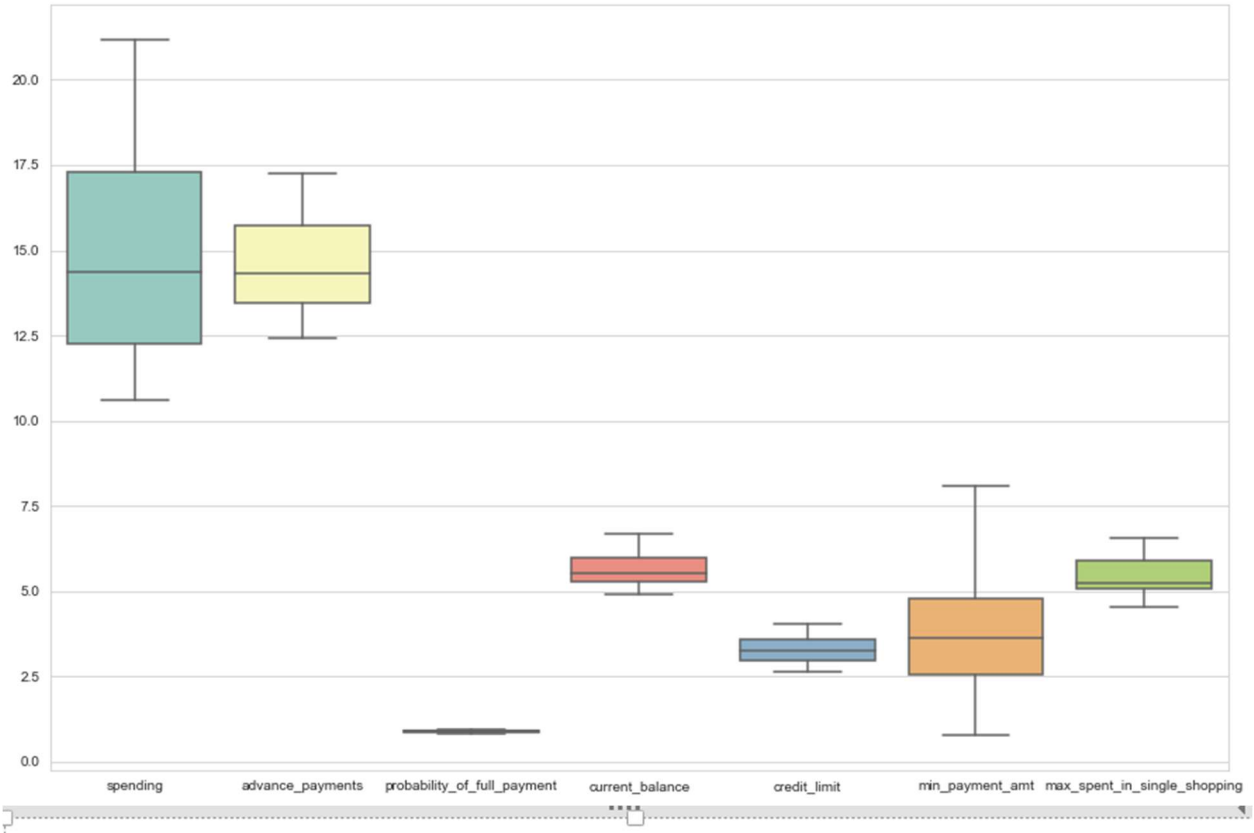

current_balance
current_balance



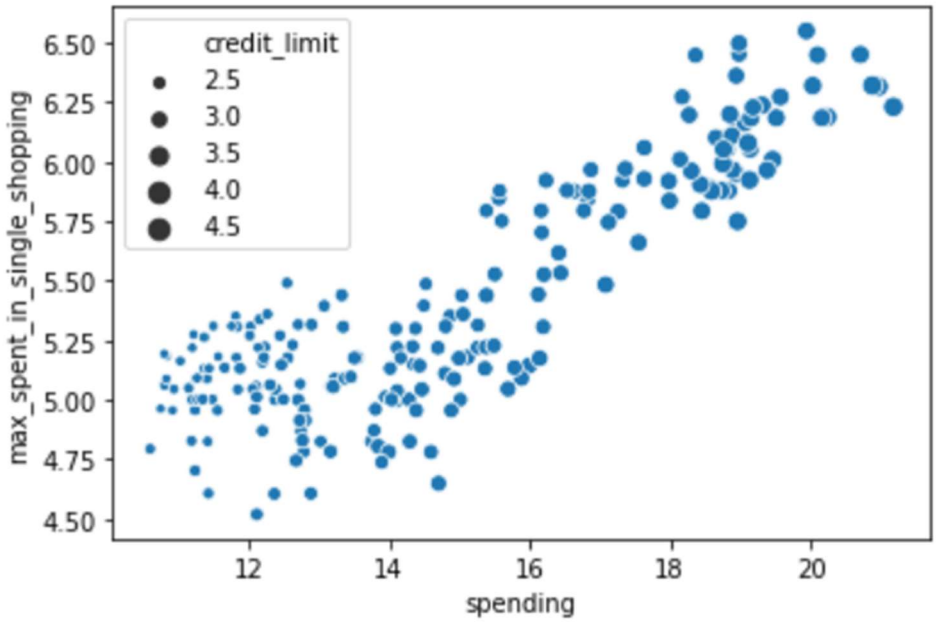
From the above distribution and box plot we can see that most of the variables are not normally distributed. We can see min payment and credit limit is normally distributed. Even from the box plot we can see the distribution of the data in q1 and q3 are almost equal.
Maximum spending is right tailed since most of the data lies above 5.25.

Outliers: From the box plot below, we can see that there are not many outliers in the data. Only in minimum payment and prob of full amount there are few outliers. The outlier has been treated using the IQR and the dataset has no outliers.

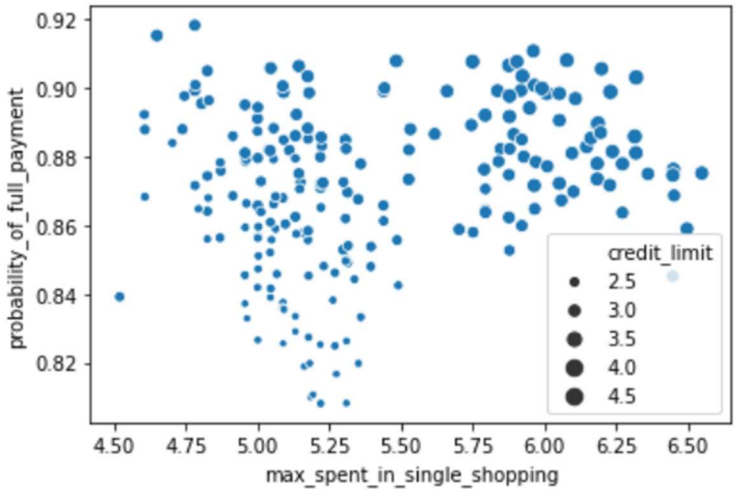




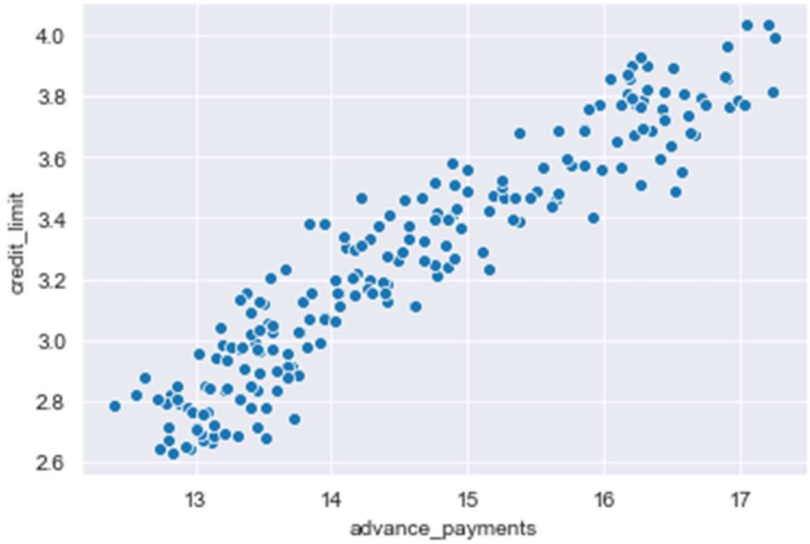
We can see above that the probability of customers making full payment of credit card bill is widely spread in the customer who have high credit limit the dark blue are (black circled). Their spending's are also high as seen earlier. People who have lesser credit limit spend less and also the probability of making full payment is much lesser(yellow circled) when compared to higher credit limit customers.



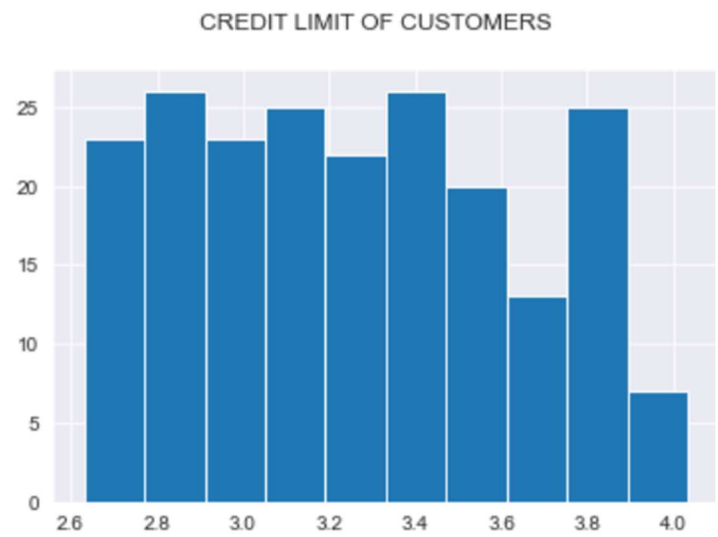
As discussed, earlier customer who are spending more are mostly spending in single shopping. From the above plot we can also infer that the customer who are given higher credit limit is spending more in single shopping when compared to lower credit limit customers. Customer with credit limit 4.0, 4.5 are the ones that are spending above 18 and 20 thousand in a month and in single shopping we can see that they are spending approximately around 6000 to 7000 per single shopping.



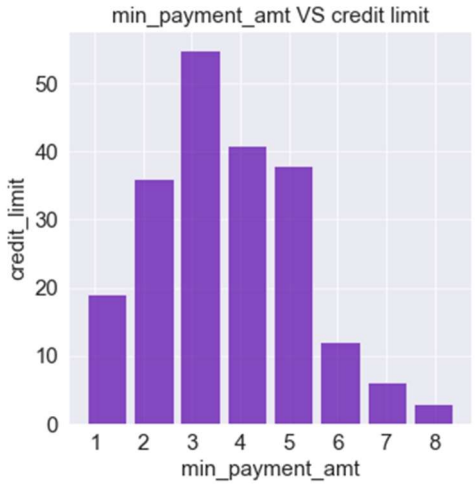
From the above we can confirm that higher credit limit customers are the ones who spend maximum on single shopping and their probability of making full payment is between 86% to 90%. We can see that the probability of making full payment is less in people who spend less on single shopping.



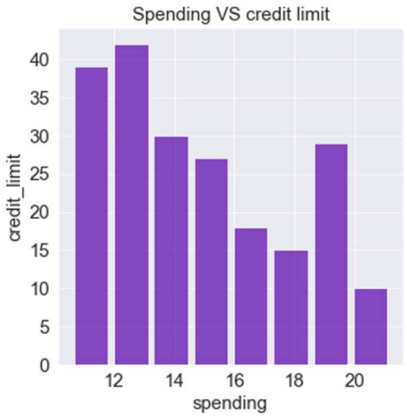
From the above scatter plot, we can infer that advance payments are done between 1300 and 1400 majorly and these are done by customers who have been allotted lesser credit limit.

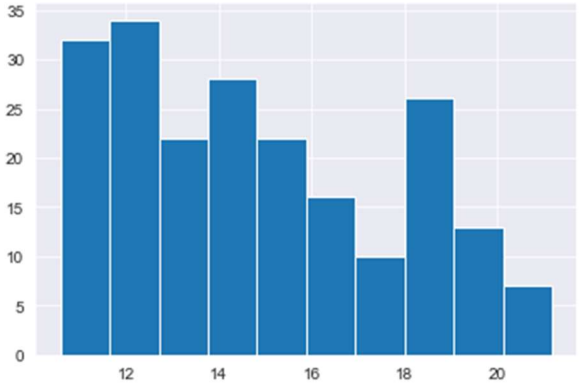


This is the spread of data for the number of customers in each credit limit. We can see that the customers are high in credit limit between 2.8 to 3 with 25 customers in that bin. And in 3.4 credit limit bin also there are around 25 customers.

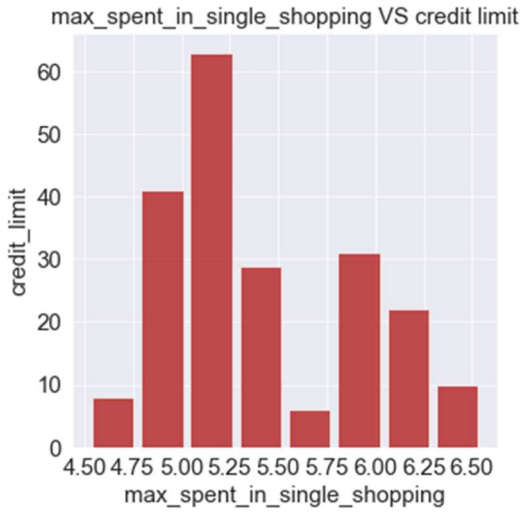


From the above data we can infer that most of the customers are making minimum payment around 3 range there are around 40 customer who have made payment between 3 and 4. And these customers put together contribute a credit limit of 50. This shows us that most of the customer make less minimum payment and only few make more minimum payment. The banks' exposure to higher credit limit is more in the minimum payment amount customer because around 100plus customer are there between 1.5 to 4.5 minimum payment range.

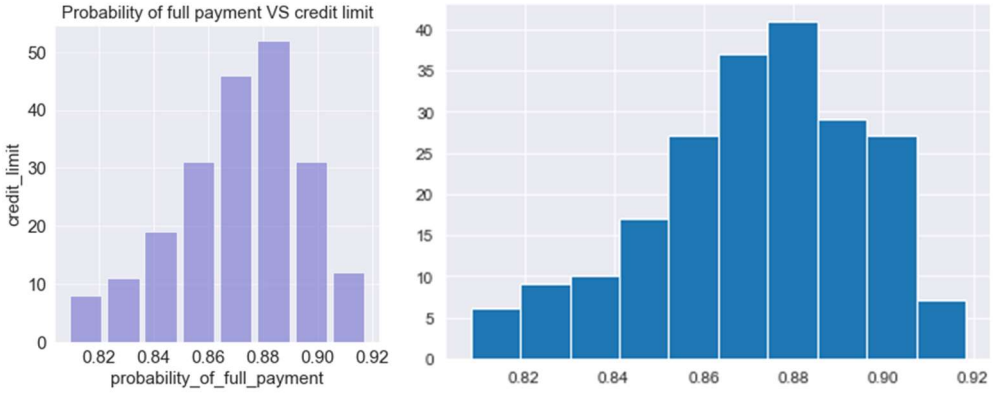




We can see here that most of the customer are there in spending power of 12 to 13 around 33 customers are there in that bin. And these customers are exposed to a total credit limit of 40 the highest exposure are to these customers. As we saw in the earlier scatter plot there were high spending between 18 to 20 among customer who had high credit limit. And above histogram also we can see around 35 customers are there in between 18 and 20th bin who are exposed to a credit limit of 32.



From the above histogram we can infer that number of customers who spend in single shopping are those who spend 5 to 5.25. And these customers put together the bank has given credit limit of 60.



We can see above that the probability of full payment is good for the bank prospective. Only a few credit limits are exposed to lesser probability paying customer. Around 70 customers are having a probability of 85% for making full payment which is good for the bank. Around 50% of the customer of total 210 customer are good paying customers. These customers put together the bank is exposed to 95 credit limits. 0,86 to 0.88 bins have a credit limit of 45 and 50, respectively.

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

	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

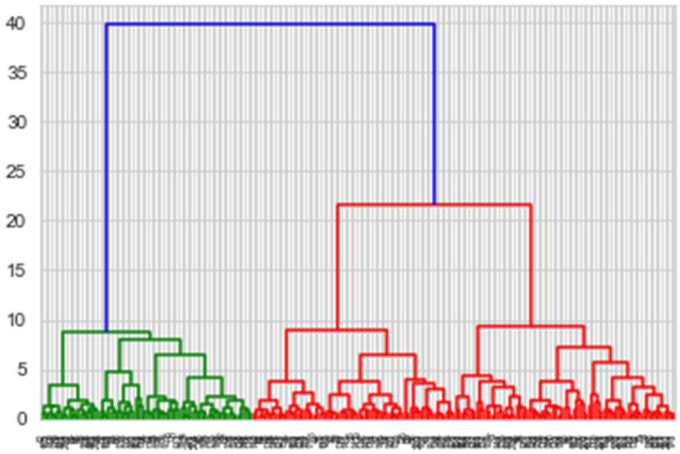
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000
mean	14.847524	14.559286	0.870999	5.628533	3.258605	3.700201	5.408071
std	2.909699	1.305959	0.023629	0.443063	0.377714	1.503557	0.491480
min	10.590000	12.410000	0.808100	4.899000	2.630000	0.765100	4.519000
25%	12.270000	13.450000	0.856900	5.262250	2.944000	2.561500	5.045000
50%	14.355000	14.320000	0.873450	5.523500	3.237000	3.599000	5.223000
75%	17.305000	15.715000	0.887775	5.979750	3.561750	4.768750	5.877000
max	21.180000	17.250000	0.918300	6.675000	4.033000	8.456000	6.550000

	0	1	2	3	4	5	6
0	1.754355	1.811968	0.177628	2.367533	1.338579	-0.298625	2.328998
1	0.393582	0.253840	1.505071	-0.600744	0.858236	-0.242292	-0.538582
2	1.413300	1.428192	0.505234	1.401485	1.317348	-0.220832	1.509107
3	-1.384034	-1.227533	-2.571391	-0.793049	-1.639017	0.995699	-0.454961
4	1.082581	0.998364	1.198738	0.591544	1.155464	-1.092656	0.874813
...	...	...	...	...	...	...	...
205	-0.329866	-0.413929	0.722220	-0.428801	-0.158181	0.193620	-1.366631
206	0.662292	0.814152	-0.307399	0.675253	0.476084	0.819993	0.789153
207	-0.281636	-0.306472	0.364831	-0.431064	-0.152873	-1.328049	-0.830235
208	0.438367	0.338271	1.232775	0.182048	0.600814	-0.957188	0.071238
209	0.248893	0.453403	-0.779662	0.659416	-0.073258	-0.709053	0.960473

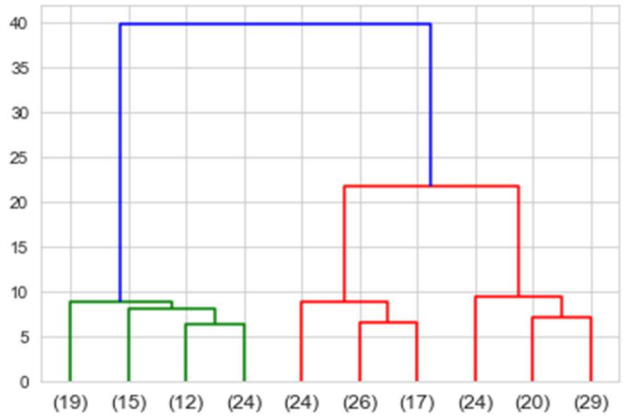
Yes, scaling is required for this dataset for clustering. It is mandatory to follow scaling in clustering. In the above info and we can see that spending and advance payment are not in same magnitude as the rest of the variables. Since clustering is a calculated on the distance-based methods if the magnitude of one variable is higher than the calculation will not be accurate since they are in different magnitude. In the describe function we can see that the mean of spending and advance payment is far away from other variables. When we do standardization, the mean will be close to 0 and standard deviation will be close to 1. Therefore, we will have to perform scaling for this dataset.

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

Data has been scaled. And the dendrogram is built on the scaled df.



LAST 10 CLUSTERS OF THE FULL DENDROGRAM IS BELOW:

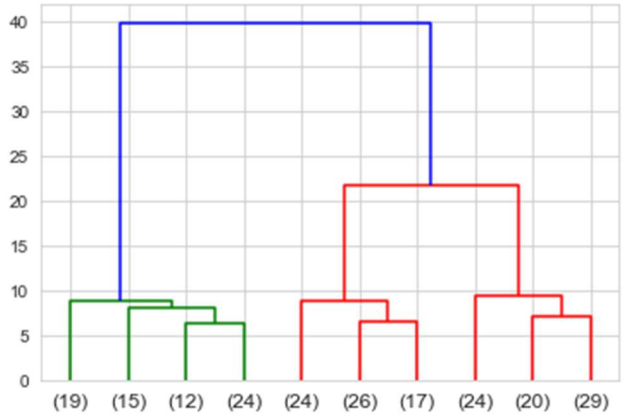


We will be using 3 clusters as this looks optimal for the dataset where more information is gathered based on 3 clusters.

clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Frequency
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

Agglo_CLusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
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

These are the 3 different clusters that we filter out from the dendrogram. We have used the ward link method here to finalize the clusters. The first cluster has around 70 rows of data and 2 cluster has 67 rows of data and 3rd has 73 rows of data. We will discuss about each cluster in detail below.



Each number mentioned below in the X axis is the number of rows of data present in each cluster and Y axis the distance. The more we move up the distance each cluster get merged and finally we can have our desired clusters where the rows of data are split based on the behaviors or each data and are clustered together.

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

```
wss
```

```
[1470.0,
 659.1717544870407,
 430.65897315130053,
 371.2935481943965,
 326.30618276116064,
 289.46717056412876,
 261.3787173805603,
 240.17374495889607,
 221.917050386605,
 204.92024405136195]
```

We can see above the within sum of square value for each cluster.

```
k_means = KMeans(n_clusters = 1,random_state=1)
k_means.fit(scaled_df1)
k_means.inertia_
```

```
1470.0
```

```
k_means = KMeans(n_clusters = 2,random_state=1)
k_means.fit(scaled_df1)
k_means.inertia_
```

```
659.1717544870407
```

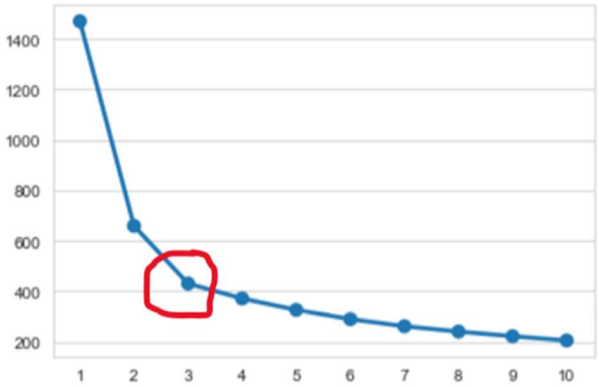
```
k_means = KMeans(n_clusters = 3,random_state=1)
k_means.fit(scaled_df1)
k_means.inertia_
```

```
430.65897315130053
```

```
k_means = KMeans(n_clusters = 4,random_state=1)
k_means.fit(scaled_df1)
k_means.inertia_
```

```
371.38509060801096
```

We can finalize on the number of clusters based on the reduction of inertia value(within sum of squares) if we see above the reduction between 1 and 2 and 2 and 3 is extremely high. However, the difference of wss reduction from cluster3 to 4 there is no huge difference. This is one indication of cluster 3 being the optimum.



We can also judge the optimum cluster looking at the above scree plot. We can see that the marked red circled area after 3 is where there is a steep dropping in other clusters and at this point, we can finalize the cluster. This is also called the elbow curve.

```
k_means = KMeans(n_clusters = 3,random_state=1)
k_means.fit(scaled_df1)
labels = k_means.labels_
silhouette_score(scaled_df1,labels,random_state=1)
```

```
0.4007270552751299
```

```
k_means = KMeans(n_clusters = 2,random_state=1)
k_means.fit(scaled_df1)
labels= k_means.labels_
silhouette_score(scaled_df1,labels,random_state=1)
```

```
0.46577247686580914
```

```
k_means = KMeans(n_clusters = 4,random_state=1)
k_means.fit(scaled_df1)
labels = k_means.labels_
silhouette_score(scaled_df1,labels,random_state=1)
```

```
0.3276547677266193
```

If we see the silhouette score is higher for cluster 2. when the silhoutte score is high we select that as the optimum cluster however in this scenario we choose 3 because that is the appropriate cluster who the dataset and also we can see that the score difference between 2 and 3 are not much but when we see the difference between 3 and 4 above we can see a huge gap in the scores.

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

Hierarchy clusters

clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Frequency
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

Cluster1: there are around 70 customers in this cluster. This is the best performing or top cluster of customers who are spending more make more advance payments and full payments. They maintain higher current balance than the rest of the cluster customers. They have the highest credit limit among the other clusters. They spend the max on single shopping.

Promotion: since these customers are spending more the bank can run a promotion for these customer in encouraging to increase their minimum payment amount on their card and when they do so and meet a target of payment made to the bank then the bank can offer shopping, dining, and health care vouchers. This will make the customer to spend more and make the minimum payment also more on this particular card so that he can earn these vouchers.

Another promotion what the bank can offer is to organize a competition that the highest top 10 spender in the country will be enjoying a fully paid trip to some holiday destination. It can also mention that the customer has to spend the highest and the payments also has to be made in full to qualify.

Cluster2: There are 67 customers in this cluster, and this is the cluster which falls behind in all the aspects compared to the other two clusters. Apart from making the

most minimum payment and spending better in max spend in single shopping in other criteria they are the last performing cluster.

Promotion: There can be some reward point promotion offer or free supplementary card given to their close family members to make these customers spent more. Based on their shopping activity since they do spend a good amount on single shopping more credit limit can be offered since their credit limits are less comparatively. we can see their current balance left is lesser so what ever limit is given to them they are utilizing it with little left to spend.

Cluster3: There are around 73 customers in this cluster and this cluster has maximum customers. This is an average performing cluster in all criteria apart from min payment amount and max spent in single shopping.

Promotion: These customers has to be concentrated in offering promotion for making more minimum payment and in spending for single shopping. Promotion such as highest spender along with maximum minimum payment can be offered shopping, food vouchers so that they spend more and make the payment.

KMeans:

Clus_kmeans3	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	freq
0	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803	71
1	11.856944	13.247778	0.848253	5.231750	2.849542	4.742389	5.101722	72
2	18.495373	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701	67

We can see above the three cluster of data. Cluster 2 customers seems to be best out of the other 2.

Cluster0: There are 71 customers in this particular cluster, and we can see that they spend the 2nd most of the three clusters. In advance payment also they are the 2nd best cluster of customers. In all the segments they are the 2nd best segment of customer that the bank has. The only issue for this cluster of customers is that they are making lesser minimum payment compared to other cluster of customers in spite of spending higher than the cluster 1.

Promotion: Bank has to focus on improving the minimum payment amount value. A promotion offer for this cluster of customers are that, the more the minimum payment they make the more cash back they can receive for the value of payment that was done. In this way we can make the customer to make more minimum payment amount.

Cluster1: This cluster has around 72 customers. This is the worst performing cluster of all 3. Neither the customers are spending or making advance payment. We can observe that the credit limit given to the customer are only at 2.8. these might be the lower middle-class customers or young customers who joined work recently as the spending is less and also the max spend in single shopping is also the least among

the three. However, we can see that these customers have the highest minimum payment.

Promotion: Bank can offer higher credit limit to those customers who has been making regular payment and having a good credit score. May be once the credit limit is high their spending goes high. Also, since the credit limit is low their one-time max spending can be less since their limit assigned is less. Therefore, bank can run an promotional offer of increasing their credit limit for free and customer who opt for higher credit limit can be offered some shopping vouchers of Rs500 and when using the voucher the customer should use that particular bank card to redeem the voucher. Then we can see an increase in the spending of these customers. Another offer that can be done is spend a fixed value on the credit card and earn shopping voucher. For e.g.: spend 50000 and when ever customer spend 50000, they earn vouchers of 1000. This can prompt the customer to spend more to earn free vouchers. This cluster has the greatest number of customers also.

Cluster2: This is the best performing cluster among the 3. The banks should concentrate highly towards these customers because they are the best in most of the parameter. These might also be the highest revenue generating customers for the bank. There are around 67 customers in this cluster and the spend the most, have the highest credit limit, they spend the highest in max spend in single shopping. These set of clusters can be classified as HNI clusters since they are provided with high limit and the advance payment made by these customers are the highest. Bank earn when their customers spend a lot on their card, so these are the revenue generating customers for the bank and the important ones.

Promotion: since these customers are spending more the bank can run a promotion for these customer in encouraging to increase their minimum payment amount on their card and when they do so and meet a target of payment made to the bank then the bank can offer shopping, dining, and health care vouchers. This will make the customer to spend more and make the minimum payment also more on this particular card so that he can earn these vouchers.

Another promotion what the bank can offer is to organize a competition that the highest top 10 spender in the country will be enjoying a fully paid trip to some holiday destination. It can also mention that the customer has to spend the highest and the payments also has to be made in full to qualify.

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

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

	Age	Commision	Duration	Sales
count	3000.000000	3000.000000	3000.000000	3000.000000
mean	38.091000	14.529203	70.001333	60.249913
std	10.463518	25.481455	134.053313	70.733954
min	8.000000	0.000000	-1.000000	0.000000
25%	32.000000	0.000000	11.000000	20.000000
50%	36.000000	4.630000	26.500000	33.000000
75%	42.000000	17.235000	63.000000	69.000000
max	84.000000	210.210000	4580.000000	539.000000

	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

We can see from the dataset that there are total of 10 variables and out of them 4 are the numerical variables. In the dataset our target variable is claimed. We have to predict the claim status by using different methods. There are no null data in the dataset. The average age of the customer is 38. There are 4 types of agency code.

```
: df.Duration.value_counts()

:  8       83
   5       82
   6       81
   10      81
   11      81
           ..
   466      1
   421      1
  -1        1
   119      1
   4580     1
  Name: Duration, Length: 257, dtype: int64
```

The duration variable has few outliers however for cart, random and neural network treating outliers is not mandatory. But we can see a data which is totally wrong like -1 and 0's. A duration can never be -1 and 0 therefore I will be replacing these two with the median of Duration for the further process. Below we can see the describe fucntion after removing -1 from now the data set the duration variable is cleaned. We can also see 4580 which looks like an extreme duration, but I am not treating them since there is no significant difference in the accuracy after and before removing it.

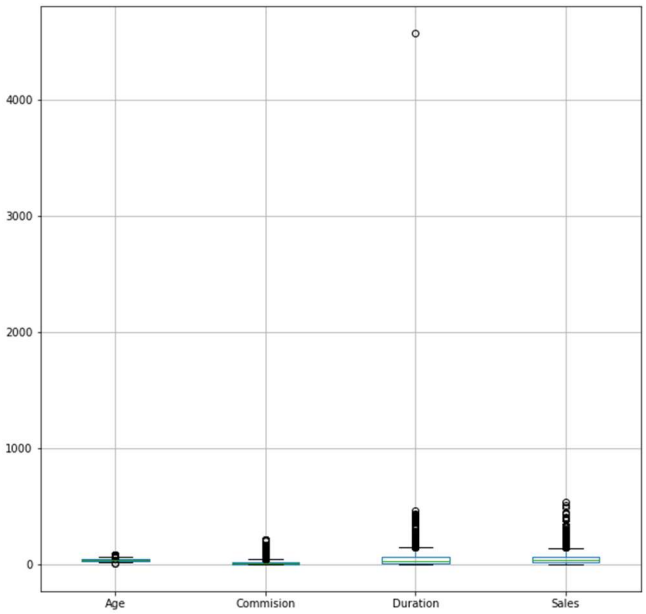
	Age	Commision	Duration	Sales
count	2861.000000	2861.000000	2861.000000	2861.000000
mean	38.204124	15.080996	72.149948	61.757878
std	10.678106	25.826834	135.964455	71.399740
min	8.000000	0.000000	1.000000	0.000000
25%	31.000000	0.000000	12.000000	20.000000
50%	36.000000	5.630000	28.000000	33.500000
75%	43.000000	17.820000	66.000000	69.300000
max	84.000000	210.210000	4580.000000	539.000000

```
<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
```

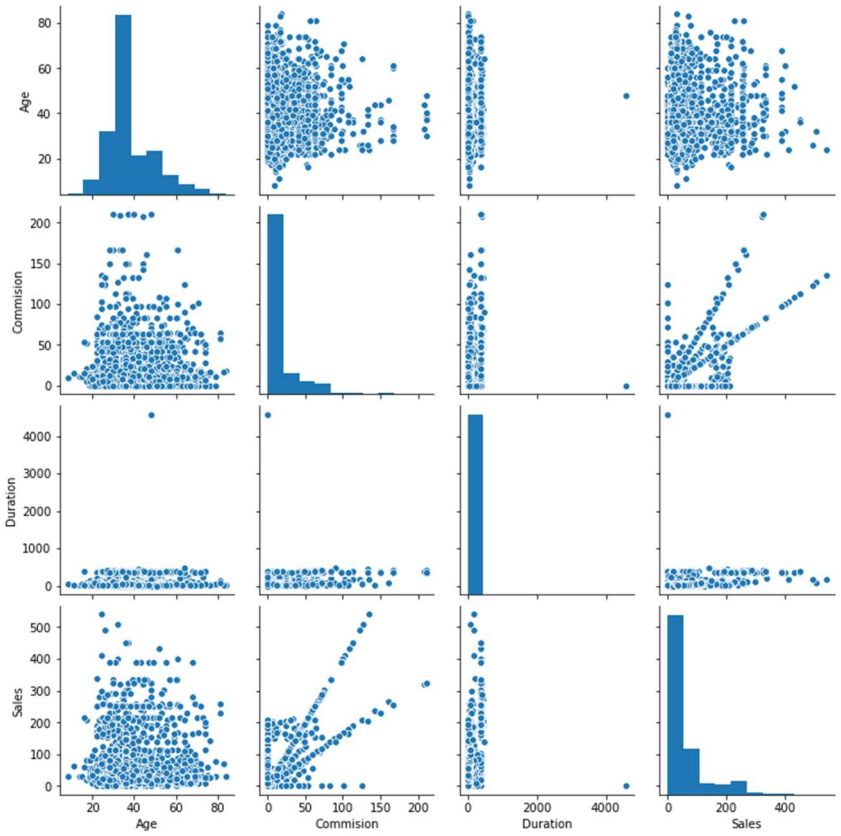
We can see that few variables are in object form and in order to perform the cart, random forest, and neural network these objects have to be converted to categorical form.

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

There are around 139 duplicated records of rows. These rows have to be removed before proceeding to converting them to integer.



There are outliers in few variables. There is not much difference in the final model prediction after treating the outliers and without treating the outliers. Moreover, few outliers in sales variables cannot be determined as outliers because there can be few tours where sales can be higher than the others. International tour insurance is higher than domestic so in order to classify this as a genuine outlier more details would be required.

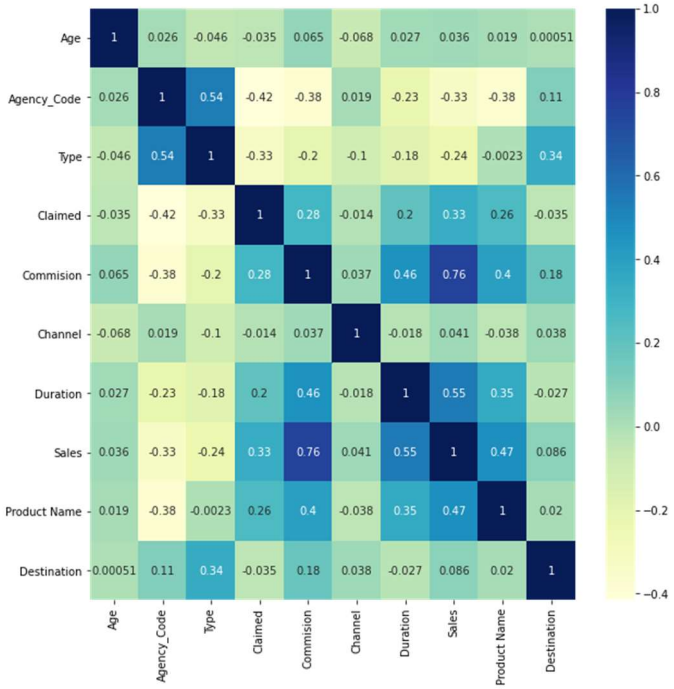


This is the spread of the dataset most of variables are not well distributed.

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
count	2861.000000	2861.000000	2861.000000	2861.000000	2861.000000	2861.000000	2861.000000	2861.000000	2861.000000	2861.000000
mean	38.204124	1.280671	0.597344	0.319469	15.080996	0.983922	72.149948	61.757878	1.666550	0.261797
std	10.678106	1.003773	0.490518	0.466352	25.826834	0.125799	135.964455	71.399740	1.277822	0.586239
min	8.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000
25%	31.000000	0.000000	0.000000	0.000000	0.000000	1.000000	12.000000	20.000000	1.000000	0.000000
50%	36.000000	2.000000	1.000000	0.000000	5.630000	1.000000	28.000000	33.500000	2.000000	0.000000
75%	43.000000	2.000000	1.000000	1.000000	17.820000	1.000000	66.000000	69.300000	2.000000	0.000000
max	84.000000	3.000000	1.000000	1.000000	210.210000	1.000000	4580.000000	539.000000	4.000000	2.000000

The variables which were object are converted to integer now. We can see below the info for the data types of the data.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2861 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Age            2861 non-null   int64
 1   Agency_Code    2861 non-null   int8
 2   Type           2861 non-null   int8
 3   Claimed        2861 non-null   int8
 4   Commision      2861 non-null   float64
 5   Channel        2861 non-null   int8
 6   Duration       2861 non-null   int64
 7   Sales          2861 non-null   float64
 8   Product Name   2861 non-null   int8
 9   Destination    2861 non-null   int8
dtypes: float64(2), int64(2), int8(6)
memory usage: 208.5 KB
```



From the above heatmap we can see that only a few variables are well correlated. Commission and sales are well correlated. Then sales and duration are correlated. Rest are not very well correlated to each other.

	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

This is the head of the data frame after converting the object variable to integer.

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

There are around 68% data where customers have not claimed and 31% of data are claimed. We can see below the details.

```
df.Claimed.value_counts()
```

```
0    1947
1     914
Name: Claimed, dtype: int64
```

```
df.Claimed.value_counts(normalize=True)*100
```

```
0    68.053128
1    31.946872
Name: Claimed, dtype: float64
```

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

y = df.pop("Claimed")

X.head()
```

To split the data for different model we have to split the data into testing and training. Above we can see that the data is split in train and test . X represent the training data which has only the independent variables and Y has only the dependent variable.

X.head()

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

```
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)
```

Above code is used to split the data into train and test. We have split the training data to 70% and test data to 30%. Below we can see the shape of the data frame after the splitting. This will be used further for all the models.

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

CART MODEL:

```
{'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 30, 'min_samples_split': 200}
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=5, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=30, min_samples_split=200,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=1, splitter='best')
```

```
grid_search.best_params_
```

```
{'criterion': 'gini',
 'max_depth': 5,
 'min_samples_leaf': 30,
 'min_samples_split': 200}
```

In the above model I have used the gini index as the criterion and the max depth that was found to be the best is 5. Max depth is maximum number of root nodes and it wont extend more than 5.

Min sample leaf is selected as 30 as the best. This means that the terminal node will have atleast 30 observation.

Min sample split is selected at 200 as the best. This means that the decision tree model is instructed to every node that splits into child nodes has to have 200 observations.

RANDOM FOREST:

```
GridSearchCV(cv=5, error_score=nan,
             estimator=RandomForestClassifier(bootstrap=True, ccp_alpha=0.0,
                                              class_weight=None,
                                              criterion='gini', max_depth=None,
                                              max_features='auto',
                                              max_leaf_nodes=None,
                                              max_samples=None,
                                              min_impurity_decrease=0.0,
                                              min_impurity_split=None,
                                              min_samples_leaf=1,
                                              min_samples_split=2,
                                              min_weight_fraction_leaf=0.0,
                                              n_estimators=100, n_jobs=None,
                                              oob_score=False, random_state=1,
                                              verbose=0, warm_start=False),
             iid='deprecated', n_jobs=None,
             param_grid={'max_depth': [5], 'max_features': [6],
                         'min_samples_leaf': [15], 'min_samples_split': [50],
                         'n_estimators': [300]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=0)
```

In random forest according to the best param function the best parameters that the model is selected are:

```
grid_search.best_params_
```

```
{'max_depth': 5,
 'max_features': 6,
 'min_samples_leaf': 15,
 'min_samples_split': 50,
 'n_estimators': 300}
```

```
best_grid = grid_search.best_estimator_
```

```
best_grid
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=5, max_features=6,
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=15, min_samples_split=50,
                       min_weight_fraction_leaf=0.0, n_estimators=300,
                       n_jobs=None, oob_score=False, random_state=1, verbose=0,
                       warm_start=False)
```

Max depth is the number of levels to which the tree has to grow. Each decision tress grows to 5 levels in this problem.
Max feature is the number of columns the model is show the prediction.
Min sample leaf: this is the 1/3rd of the min sample split
Min sample split is taken at max 3% of the size of the training data.

NEURAL NETWORK:

```
array([[ 2.88764239, -1.2626112 , -1.19813318, ..., -0.65375471,
        -1.31338076, -0.44775345],
       [-0.21666128,  0.71683095,  0.83463176, ..., -0.37032806,
         0.24339146, -0.44775345],
       [ 2.04101412, -0.27289013,  0.83463176, ...,  0.11574864,
         0.24339146,  1.24676906],
       ...,
       [-0.21666128,  0.71683095,  0.83463176, ..., -0.68209737,
        -0.53499465, -0.44775345],
       [-0.21666128,  0.71683095,  0.83463176, ...,  0.72086453,
         0.24339146, -0.44775345],
       [-0.21666128,  0.71683095,  0.83463176, ...,  0.72086453,
         0.24339146,  1.24676906]])
```

Scaling was done before doing the neural network. Scaling is not mandatory for any of the models however, neural network shows better result on scaled data.

```
{'hidden_layer_sizes': 100, 'max_iter': 150, 'solver': 'adam', 'tol': 0.01}
```

```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=100, learning_rate='constant',
              learning_rate_init=0.001, max_fun=15000, max_iter=150,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=1, shuffle=True, solver='adam',
              tol=0.01, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

Hidden layers are creating layers and here we have 100 neurons in one layer.

Max iter: this is set to run the model beyond 150 iteration for updating the synoptic weight that are randomly initiated with the MLP classifier.

Solver: this the method used for solving and adam is selected.

Activation: relu is the activation fucntion used here.

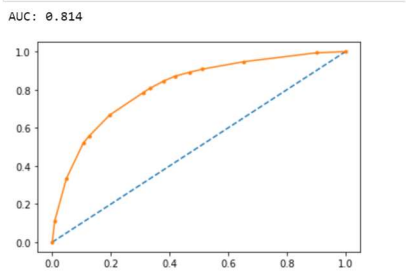
2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model.

CART MODEL:

	Imp
Agency_Code	0.579764
Sales	0.294460
Product Name	0.045725
Commision	0.043767
Duration	0.023026
Age	0.013258
Type	0.000000
Channel	0.000000
Destination	0.000000

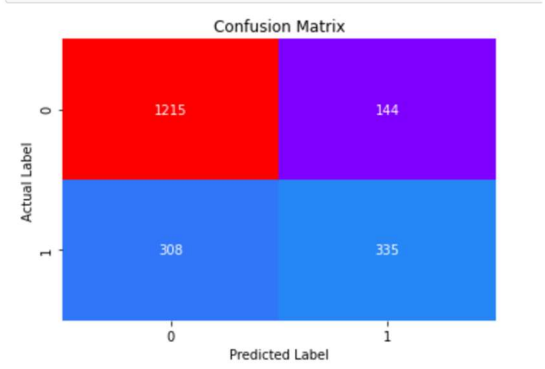
Agency code is the most important feature in the cart model. Then followed by sales and product name. the above variable contributes to the claim.

CART ROC and AUC:



The AUC score for the training data of the cart model is 81%.

Confusion matrix cart:




```
array([[1215,  144],
       [ 308,  335]], dtype=int64)
```

The above is the confusion matrix of the cart model.
0 is not claimed and 1 is claimed.

1215 is the number of people we had said that they wont claim and the model also predicted they wont claim.
308 we said will claim for insurance, but the model predicted they will not claim.
144 we said they will not claim, but the model said they will claim

335 is the number we said they will claim, and the model also predicted they will approach for claim.

ACCURACY SCORE CART TRAINING:

```
: 0.7742257742257742
```

The accuracy score for cart training is 77%

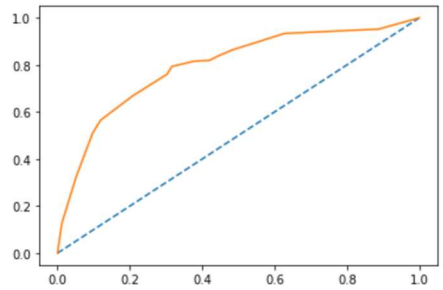
Classification Report for cart training:

	precision	recall	f1-score	support
0	0.80	0.89	0.84	1359
1	0.70	0.52	0.60	643
accuracy			0.77	2002
macro avg	0.75	0.71	0.72	2002
weighted avg	0.77	0.77	0.76	2002

CART TEST DATA:

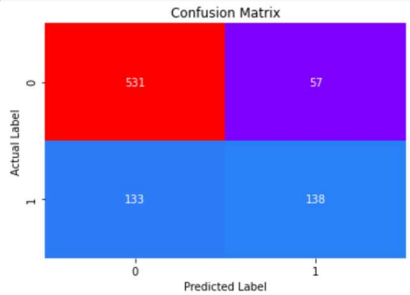
ROC AUC

```
AUC: 0.795
[<matplotlib.lines.Line2D at 0x160e914f748>]
```



CONFUSION MATRIX

```
: array([[531,  57],
         [133, 138]], dtype=int64)
```



ACCURACY FOR TEST DATA.

```
: 0.7788125727590222
```

CLASSIFICATION REPORT FOR CART TESTING DATA:

	precision	recall	f1-score	support
0	0.80	0.90	0.85	588
1	0.71	0.51	0.59	271
accuracy			0.78	859
macro avg	0.75	0.71	0.72	859
weighted avg	0.77	0.78	0.77	859

Cart Model conclusion:

Train Data:

AUC: 81%
Accuracy:77%
Precision:70%
F1 Score: 60%

Test Data:

AUC: 79%
Accuracy: 78%
Precision: 71%
f1-Score: 59%

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

Accuracy, AUC, Precision and Recall for test data is almost inline with training data.
This proves no overfitting or underfitting has happened, and overall the model is a good model for classification

Agency code is the most important variable for predicting the claim frequency.

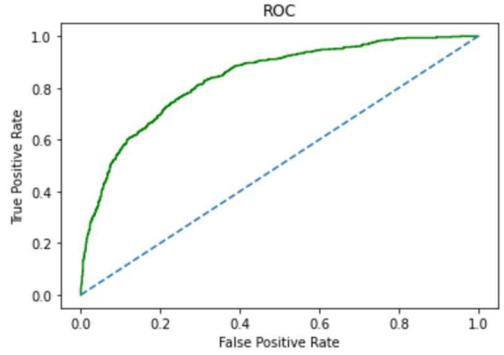
RANDOM FOREST:

TRAINIGN DATA:

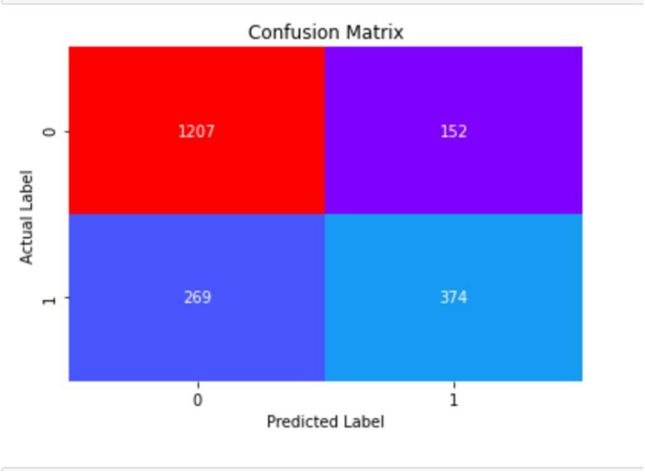
ROC AND AUC:

AUC IS 84% for random forest training data.

```
Area under Curve is 0.8375560888357897
```



CONFUSION MATRIX:




```
array([[1207,  152],
       [ 269,  374]], dtype=int64)
```

TRAINIGN ACCURACY FOR RANDOM FOREST:

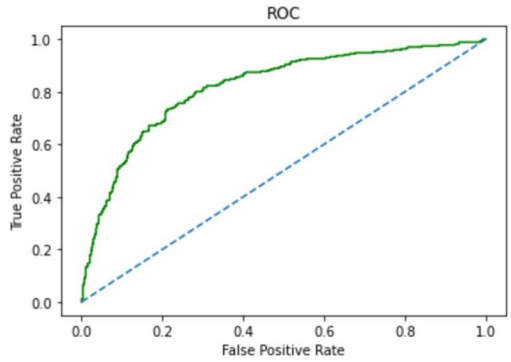
```
0.7897102897102897
```

CLASSIFICATION MODEL FOR TRAINIGN DATA:

	precision	recall	f1-score	support
0	0.82	0.89	0.85	1359
1	0.71	0.58	0.64	643
accuracy			0.79	2002
macro avg	0.76	0.73	0.75	2002
weighted avg	0.78	0.79	0.78	2002

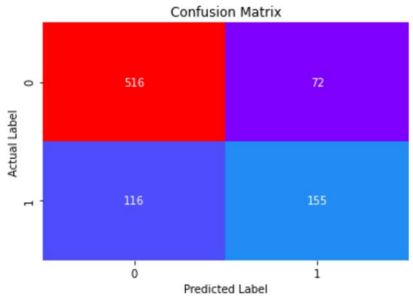
RANDOM FOREST TESTING DATA:

```
Area under Curve is 0.8186014258101765
```



CONFUSION MATRIC TEST DATA:

```
array([[516,  72],
       [116, 155]], dtype=int64)
```



TESTING DATA ACCURACY:

```
0.7811408614668219
```

CLASSIFICATION REPORT TRAINING DATA:

	precision	recall	f1-score	support
0	0.82	0.88	0.85	588
1	0.68	0.57	0.62	271
accuracy			0.78	859
macro avg	0.75	0.72	0.73	859
weighted avg	0.77	0.78	0.78	859

IMPORTANT FEATURES FOR RANDOM FOREST:

	Imp
Agency_Code	0.383236
Product Name	0.202148
Sales	0.199188
Commision	0.087202
Duration	0.059552
Age	0.038894
Type	0.019249
Destination	0.009839
Channel	0.000692

RANDOM FOREST CONCLUSION

TRAIN DATA

AUC: 84%
ACCURACY: 79%
PRECISION: 71%
F1 SCORE:64%

TEST DATA:

AUC: 82%
ACCURACY: 78%
PRECISION: 68%
F1 - SCORE: 62%

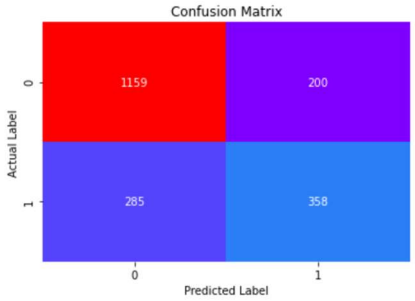
Training and Test set results are almost similar, and with the overall measures high, the model is a good model.

Agency code is again the important feature for prediction the frequency of claim. There is a difference of only 2% betwe
en the training and testing data AUC. There is no overfitting or underfitting here.

NEURAL NETWORK TRAINIGN DATA:

ConfusionMatrix:

```
array([[1159,  200],
       [ 285,  358]], dtype=int64)
```



Accuracy score for training data:

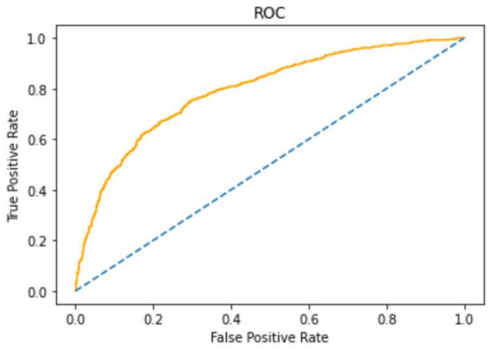
```
0.7577422577422578
```

Classification Report:

	precision	recall	f1-score	support
0	0.80	0.85	0.83	1359
1	0.64	0.56	0.60	643
accuracy			0.76	2002
macro avg	0.72	0.70	0.71	2002
weighted avg	0.75	0.76	0.75	2002

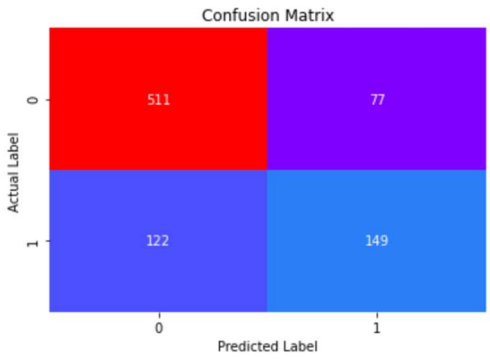
AUC AND ROC CURVE:
AUC SCORE IS 79%

```
Area under Curve is 0.7921265636497425
```



NEURAL NETWORK TESTING DATA:

Confusion Matrix



Accuracy Score:

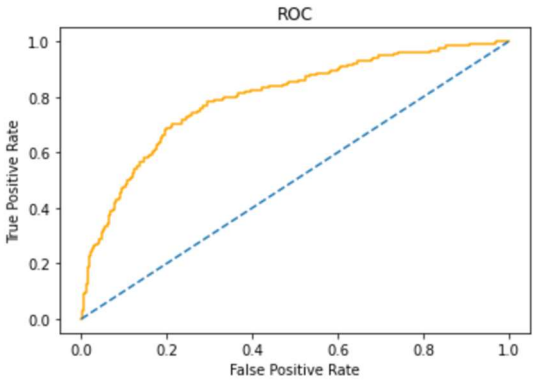
```
0.7683352735739232
```

Classification Report:

	precision	recall	f1-score	support
0	0.81	0.87	0.84	588
1	0.66	0.55	0.60	271
accuracy			0.77	859
macro avg	0.73	0.71	0.72	859
weighted avg	0.76	0.77	0.76	859

AUC AND ROC CURVE

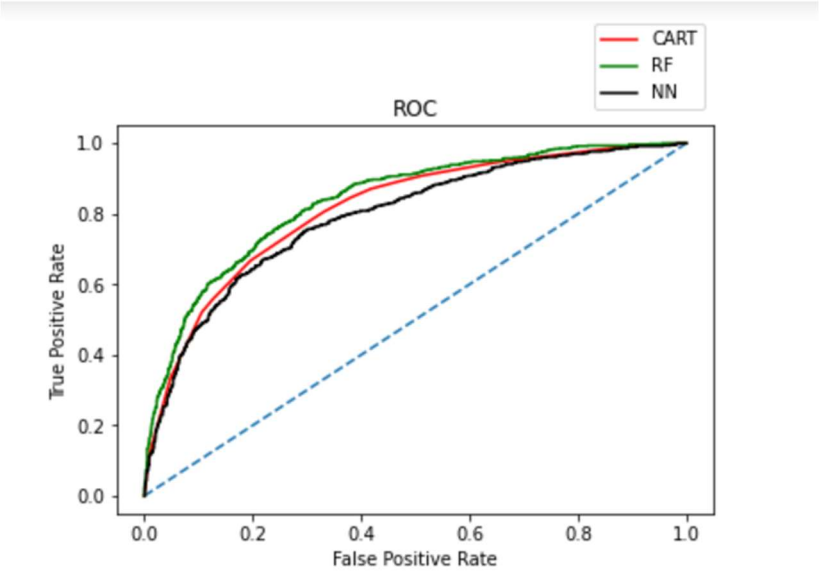
```
Area under Curve is 0.7977947636619223
```



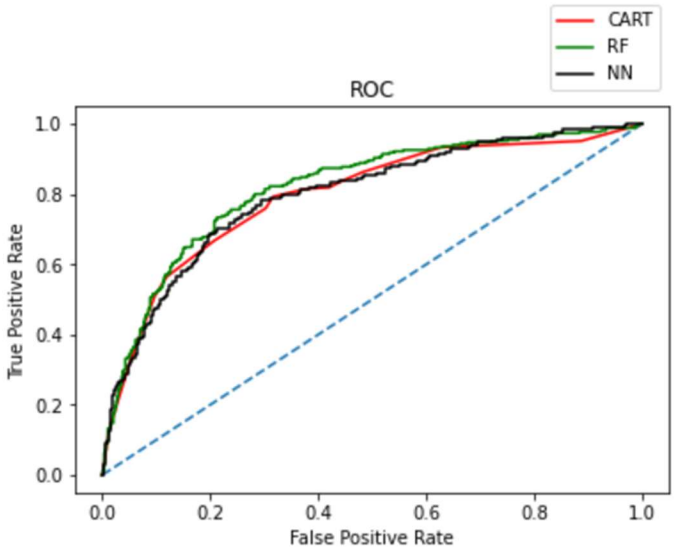

```
<bound method ClassifierMixin.score of MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=100, learning_rate='constant',
              learning_rate_init=0.001, max_fun=15000, max_iter=150,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=1, shuffle=True, solver='adam',
              tol=0.01, validation_fraction=0.1, verbose=False,
              warm_start=False)>
```

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

TRAINING DATA ROC CURVE OF ALL THE MODELS:



TESTING DATA ALL THE MODELS ROC CURVE:



We can confirm that all the 3 models are predicting almost very similarly. However, out of the 3 the random forest performs slightly better than the other two which can be seen from the above graph.

	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.77	0.78	0.79	0.78	0.76	0.77
AUC	0.81	0.80	0.84	0.82	0.79	0.80
Recall	0.52	0.51	0.58	0.57	0.56	0.55
Precision	0.70	0.71	0.71	0.68	0.64	0.66
F1 Score	0.60	0.59	0.64	0.62	0.60	0.60

From the table we can also see that the accuracy is best for random forest and cart at same level of 78%. The AUC score is best in random forest at 82%. Recall is best in random forest at 57%. Therefore, random forest is the best among the 3 models, just slightly better than the others.

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

Our aim was to check the prediction for whether a customer will apply for claim that is the YES in the variable claim. And we found out that agency code might be most important variable that can be responsible for an increase in claim. Both in cart and random forest the agency code was the main variable that can cause the increase in claim.

CART:

	Imp
Agency_Code	0.579764
Sales	0.294460
Product Name	0.045725
Commision	0.043767
Duration	0.023026
Age	0.013258
Type	0.000000
Channel	0.000000
Destination	0.000000

RANDOM FOREST:

	Imp
Agency_Code	0.383236
Product Name	0.202148
Sales	0.199188
Commision	0.087202
Duration	0.059552
Age	0.038894
Type	0.019249
Destination	0.009839
Channel	0.000692

We can see above that agency code and sales might be important factor for increase in claim according to cart. According to random forest it is agency code and product name closely followed by sales that might lead to increase in claim.

Agency Code:
There are total of four agencies. C2B, EPX, CWT, JZI.
From the prediction we can confirm that C2B is contributing more to the claim prediction when compared to other. We can see below agency code 0 is allotted for C2B and in the crosstab we can see prediction 1 which is claimed is 168 in 0 (C2b) so we can take measure in future and be prepared for the claim from this agency. This prediction can be valid only for cart and random forest because they have shown us the agency code as the important feature that has predicted this variable will be contributing to claims.

Agency_Code	0	1	2	3
Prediction				
0	98	101	330	59
1	168	39	59	5

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

Product name: The product that is predicted to be claimed more is the customized plan product.

Product Name	0	1	2	3	4
Prediction					
0	119	194	233	12	30
1	75	16	68	26	86

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