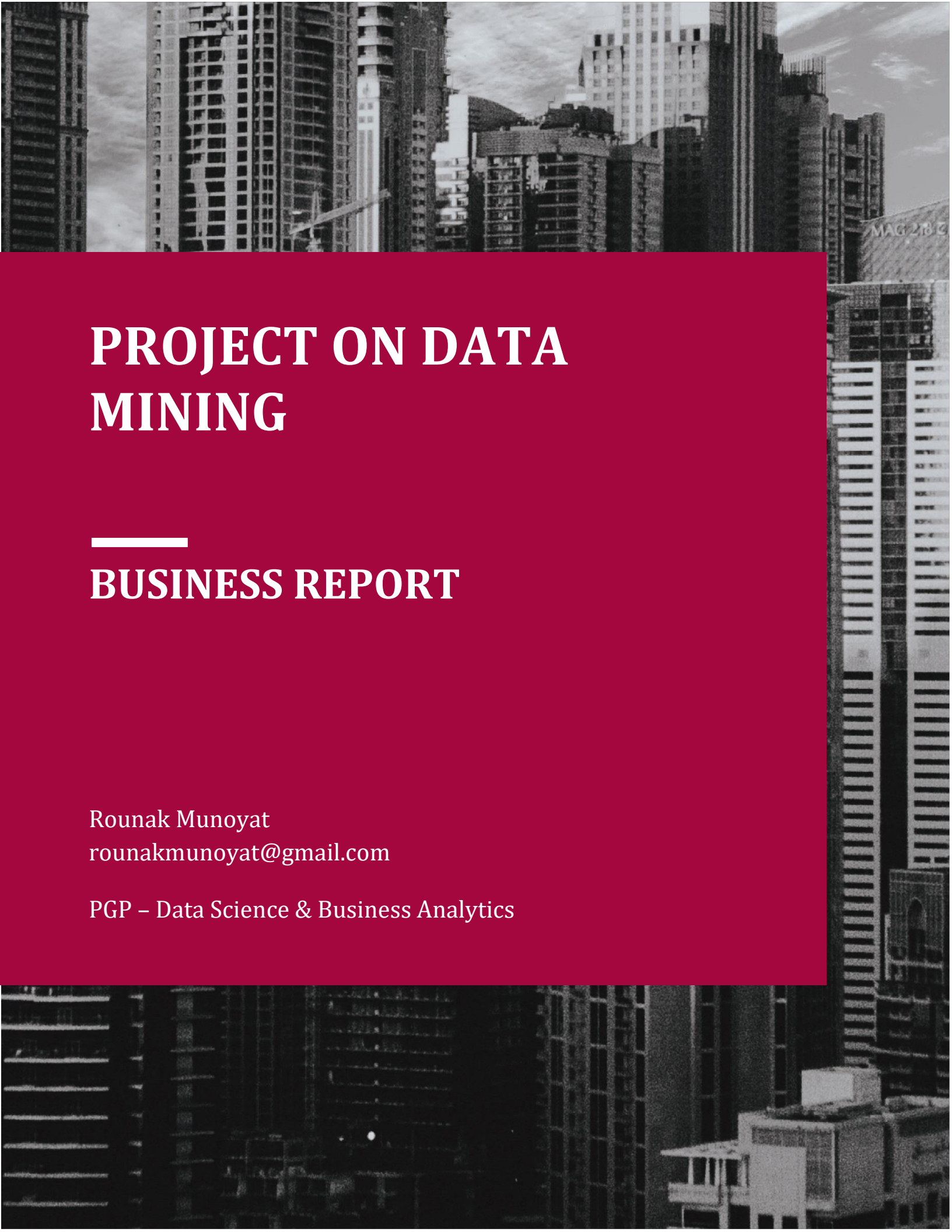

PROJECT ON DATA MINING

BUSINESS REPORT

Rounak Munoyat
rounakmunoyat@gmail.com

PGP – Data Science & Business Analytics

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Case Study 1 - Clustering

Overview:

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.

Summary:

This business report provides detailed explanation on the approach to each problem definition, solution to those the problems provide some key insights/recommendations to the business.

Q1.1) Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis).

	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

Table 1: Sample of the credit card dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   spending                      210 non-null    float64
 1   advance_payments              210 non-null    float64
 2   probability_of_full_payment   210 non-null    float64
 3   current_balance               210 non-null    float64
 4   credit_limit                  210 non-null    float64
 5   min_payment_amt               210 non-null    float64
 6   max_spent_in_single_shopping  210 non-null    float64
dtypes: float64(7)
memory usage: 11.6 KB
```

Table 2: Basic Information of 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

Table 3: Summary of the dataset

Basic observations of the dataset from table 2 -

- The dataset contains 210 rows & 7 columns.
- All the variables/columns are of float datatype.
- There are no null and missing values in the dataset.
- The dataset does not have any duplicate values as well.

Univariate Analysis

1) Outlier Identification

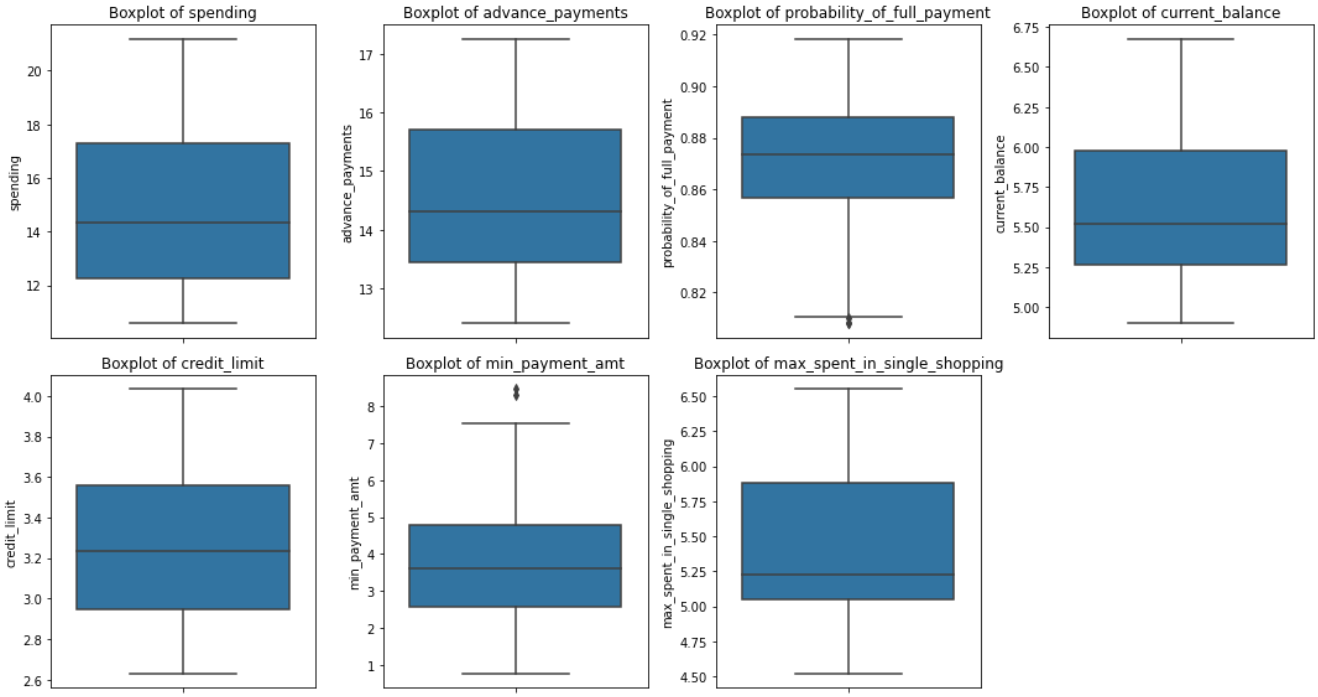

Figure 1: Boxplot for Outlier Identification

From the figure above, we can say that,

- Of all the 7 variables /columns, only variables 'probability_of_full_payment' & 'min_payment_amt' have outliers within them.
- The dataset in question has a very few outliers in total, hence, we could ignore them as they are not so far from extreme.

2) Distribution Check

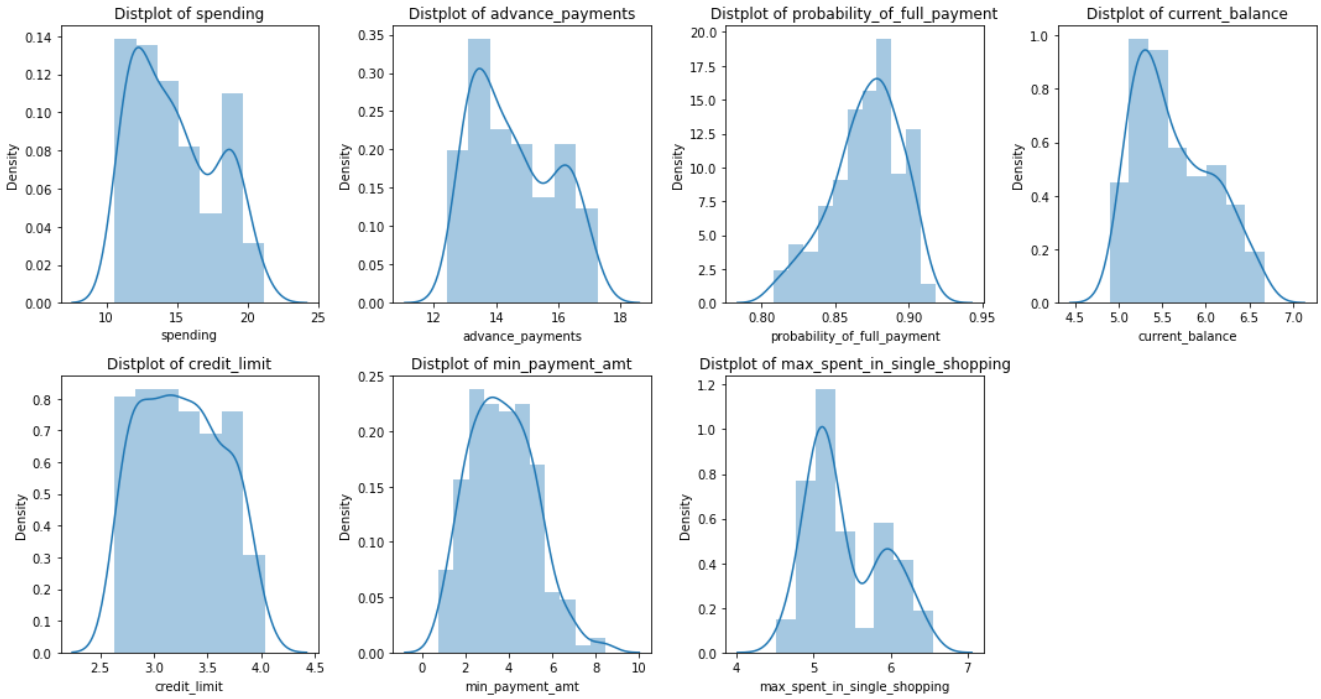

Figure 2: Distribution Check for all variables

Checking skew values for all variables

```
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
```

From the figure & skew values above, we can say that,

- Since, the skewness value of variables/columns 'spending', 'advance_payments', 'credit_limit' and 'min_payment_amt ' is between −0.5 and +0.5, they show approximately symmetric/normal distribution.
- Since, the skewness value of variables/columns 'probability_of_full_payment', 'current_balance' and 'max_spent_in_single_shopping' is between −1 and −0.5 or between +0.5 and +1, they show moderately skewed distribution.

Multivariate Analysis

3) Heatmap to study correlation between variables

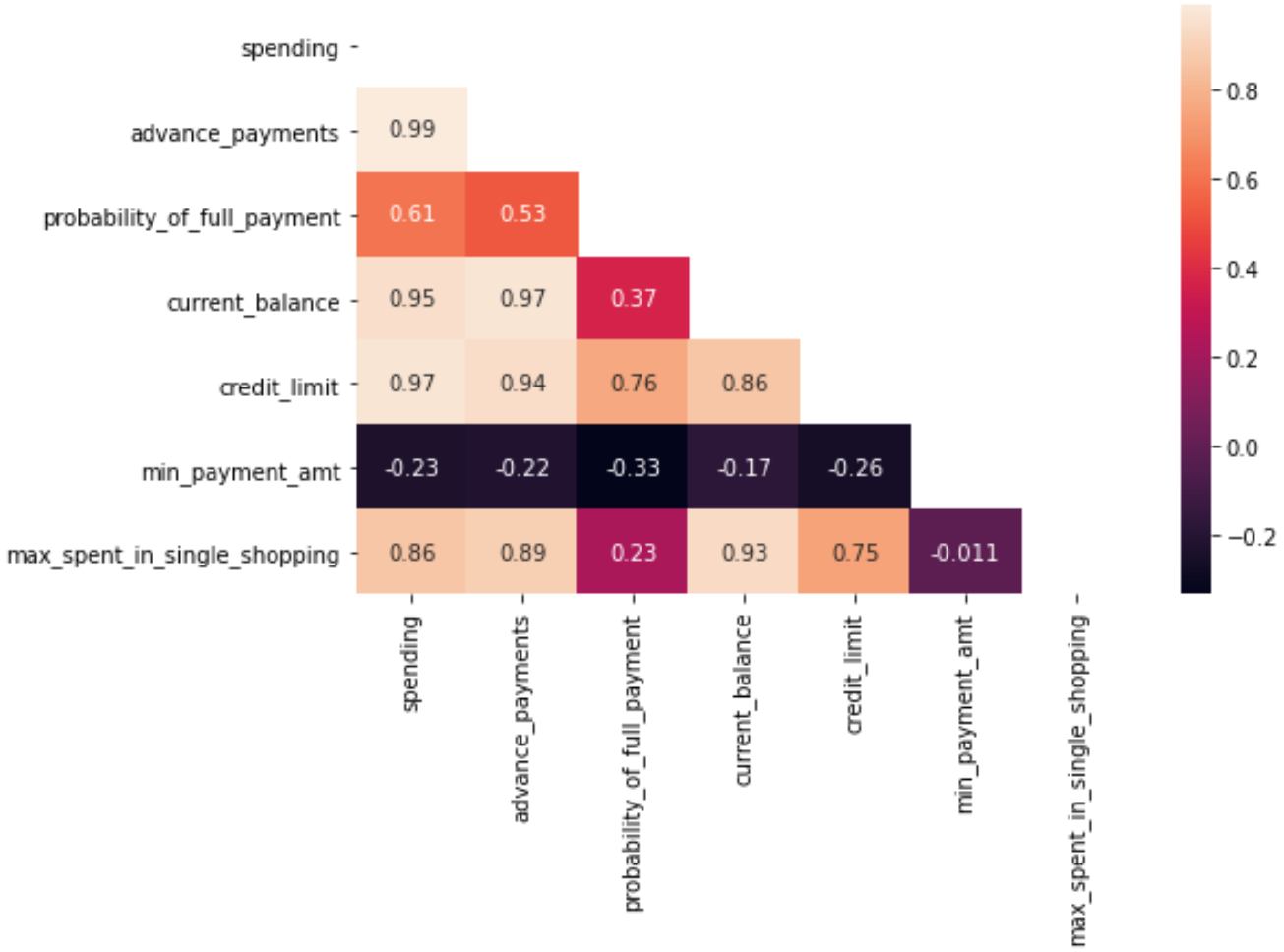

Figure 3: Correlation Matrix

From the figure above, we can say that,

- There is a strong correlation between some variables. ('advance_payments' & 'spending' being the highest i.e. 0.99)
- The variable 'spending' & 'advance_payments' have a strong correlation with variables 'current balance' & 'credit limit' (0.95, 0.97, 0.97 & 0.94 respectively).
- The variable 'current balance' has a strong correlation with the variable 'max_spent_in_single_shopping' (0.93).

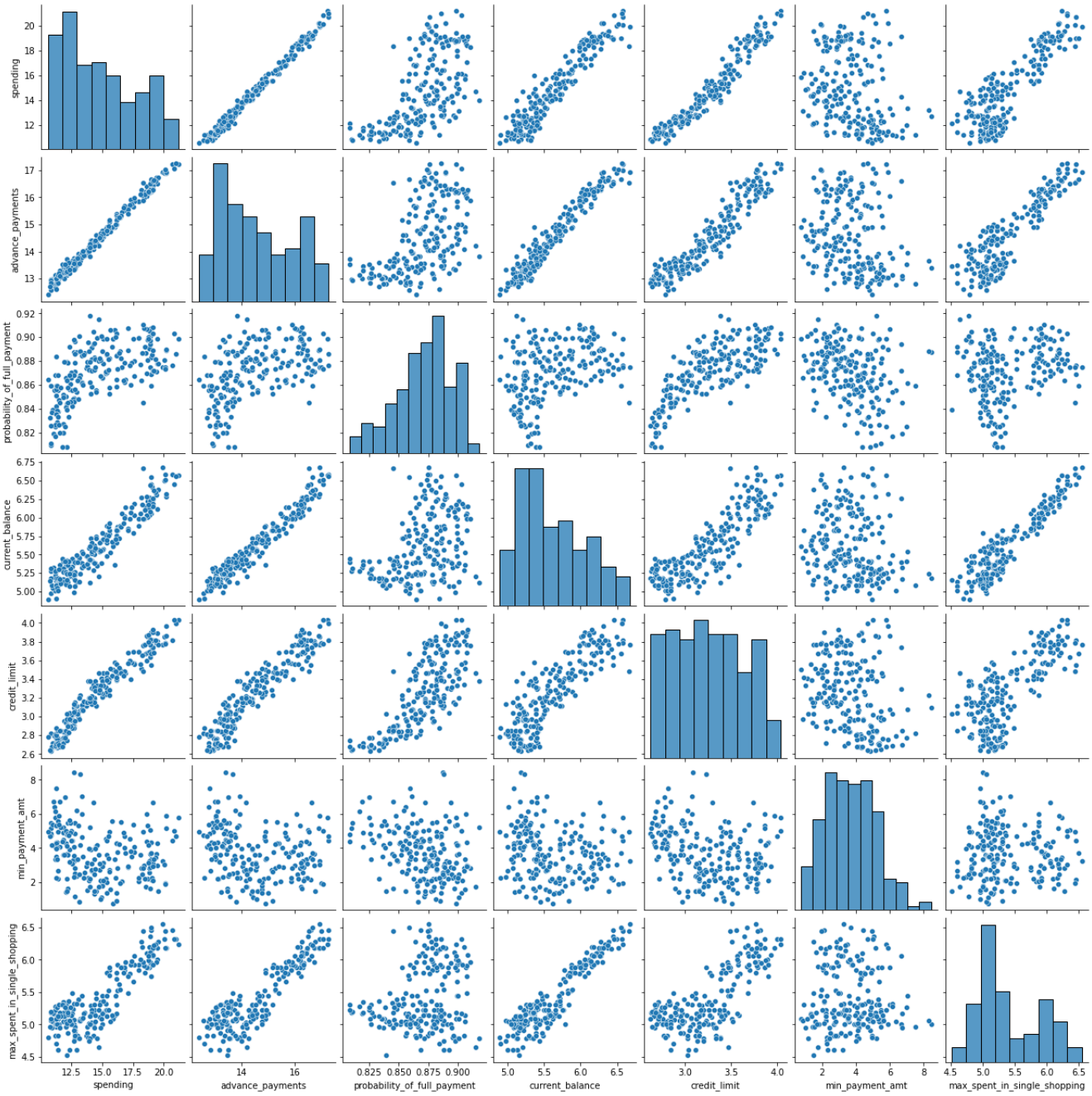

Figure 4: Pair plot for all variable combinations

From the figure above, we can say that,

- All the correlation values can be viewed and verified in the pair plot as the data points are closely packed to each other in the above combinations of variables.
- Some variables do have correlation, but, is weaker in strength which is evident in the graph.

Q1.2) Do you think scaling is necessary for clustering in this case? Justify

Yes, scaling is necessary in this case because clustering algorithms do need feature scaling before they are fed to the algorithm. This is because clustering techniques use mathematical distances to form the cohorts, hence, all the features need to have same scale for proper clustering.

Scaling of data can be done by 2 methods -

- Standard Scaler
- Z-score Method

1) Standard Scaler

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

Table 4: Sample of the dataset after scaling using standard scaler

2) Z-score method

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

Table 5: Sample of the dataset after scaling using Z-score method

	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

Table 6: Summary of the dataset before scaling

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	2.100000e+02	2.100000e+02	2.100000e+02	2.100000e+02	2.100000e+02	2.100000e+02	2.100000e+02
mean	9.148766e-16	1.097006e-16	1.260896e-15	-1.358702e-16	-2.790757e-16	5.418946e-16	-1.935489e-15
std	1.002389e+00	1.002389e+00	1.002389e+00	1.002389e+00	1.002389e+00	1.002389e+00	1.002389e+00
min	-1.466714e+00	-1.649686e+00	-2.668236e+00	-1.650501e+00	-1.668209e+00	-1.956769e+00	-1.813288e+00
25%	-8.879552e-01	-8.514330e-01	-5.980791e-01	-8.286816e-01	-8.349072e-01	-7.591477e-01	-7.404953e-01
50%	-1.696741e-01	-1.836639e-01	1.039927e-01	-2.376280e-01	-5.733534e-02	-6.746852e-02	-3.774588e-01
75%	8.465989e-01	8.870693e-01	7.116771e-01	7.945947e-01	8.044956e-01	7.123789e-01	9.563941e-01
max	2.181534e+00	2.065260e+00	2.006586e+00	2.367533e+00	2.055112e+00	3.170590e+00	2.328998e+00

Table 7: Summary of the dataset after scaling

From the tables 4,5,6 & 7 we can say that,

- Both methods of scaling lead to same output.
- Scaling brings all variables to same scale as we can see from both the summary tables which shows mean, standard deviation, maximum & minimum values, etc.

Q1.3) Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them

Hierarchical Clustering -

Hierarchical clustering method is based on hierarchy representation of clusters where parent cluster node is connected to further to child cluster node. A node represents collection of data points to one cluster.

It is further divided into two types:

- Agglomerative Clustering
- Divisive Clustering

We will be using agglomerative approach (i.e. Dendrogram)

A dendrogram is a diagram that shows the hierarchical relationship between objects. It is most commonly created as an output from hierarchical clustering. The main use of a dendrogram is to work out the best way to allocate objects to clusters. It helps in determining the optimum number of clusters.

Choosing linkage method -

- In most cases, Ward's Linkage is preferred as it usually produces better cluster hierarchies.
- Also, Ward's method is less susceptible to noise and outliers.

Hence, we will be using ward's linkage method. It uses Euclidean distance as its affinity.

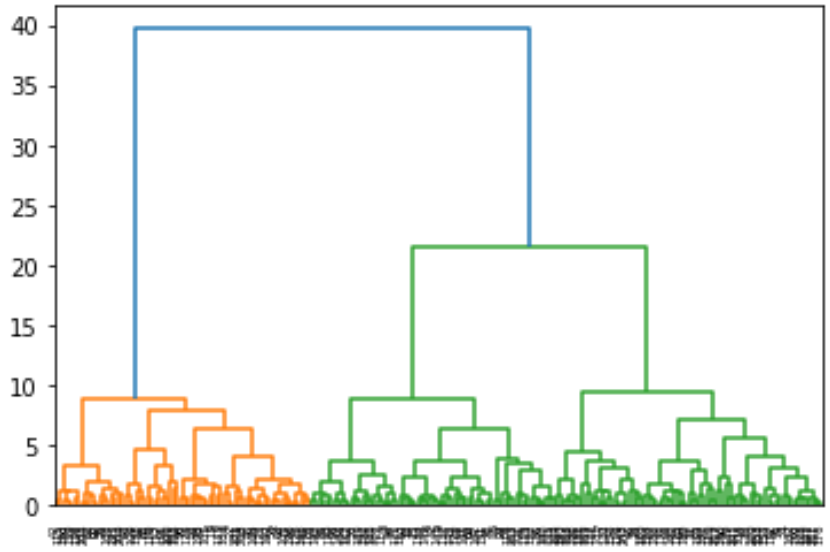

Figure 5: Dendrogram

From the dendrogram above, we come to know that, the optimal number of clusters is 2. However, the dendrogram above is a bit difficult to read, hence, we cut the dendrogram to understand it better.

In order to understand the dendrogram and the data split into clusters better we would cut the dendrogram in such a manner, that, we come to see only last 10 merges.

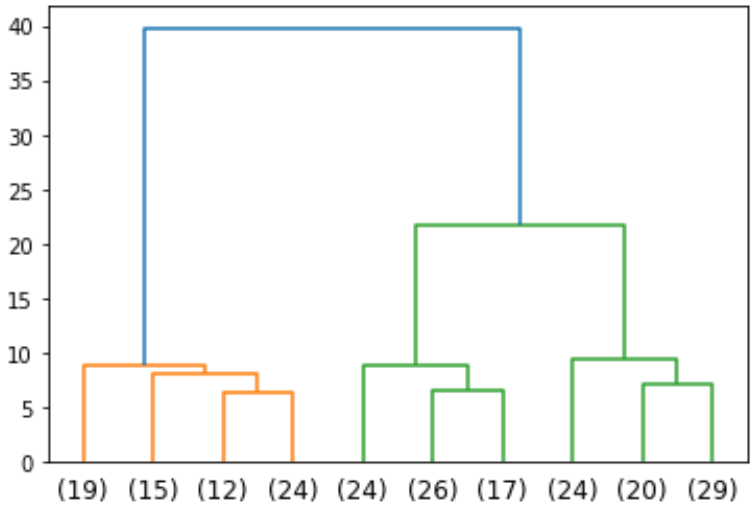

Figure 6: Modified Dendrogram which has only the last 10 merges

From the modified dendrogram above, we can say that,

- There are more no. of customers in cluster 2 compared to cluster 1.

Cluster formation & appending them to the dataset-

This could be done in two ways via criterion - namely maxclust or distance.

- In maxclust, we decide the maximum number of clusters we wish to create for the dataset
- In distance, we decide the distance on the y-axis in dendrogram and mention at which point we wish to cut for cluster formation.

Here, we will be using both, using maxclust=2 & distance=20.

We got 2 optimum cluster numbers from different methods. However, considering the business point of view we'll make 3 clusters to understand customer segmentation and their behaviour in a much better fashion.

Once we have identified which observation belongs to which cluster, we append them to the dataset.

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

Table 8: Sample of the dataset after appending clusters to original dataset

Frequency of each cluster in the dataset

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

From the above output, we can say that, the number of observations in cluster 3 (73) is more compared to cluster 1 (70) & cluster 2 (67).

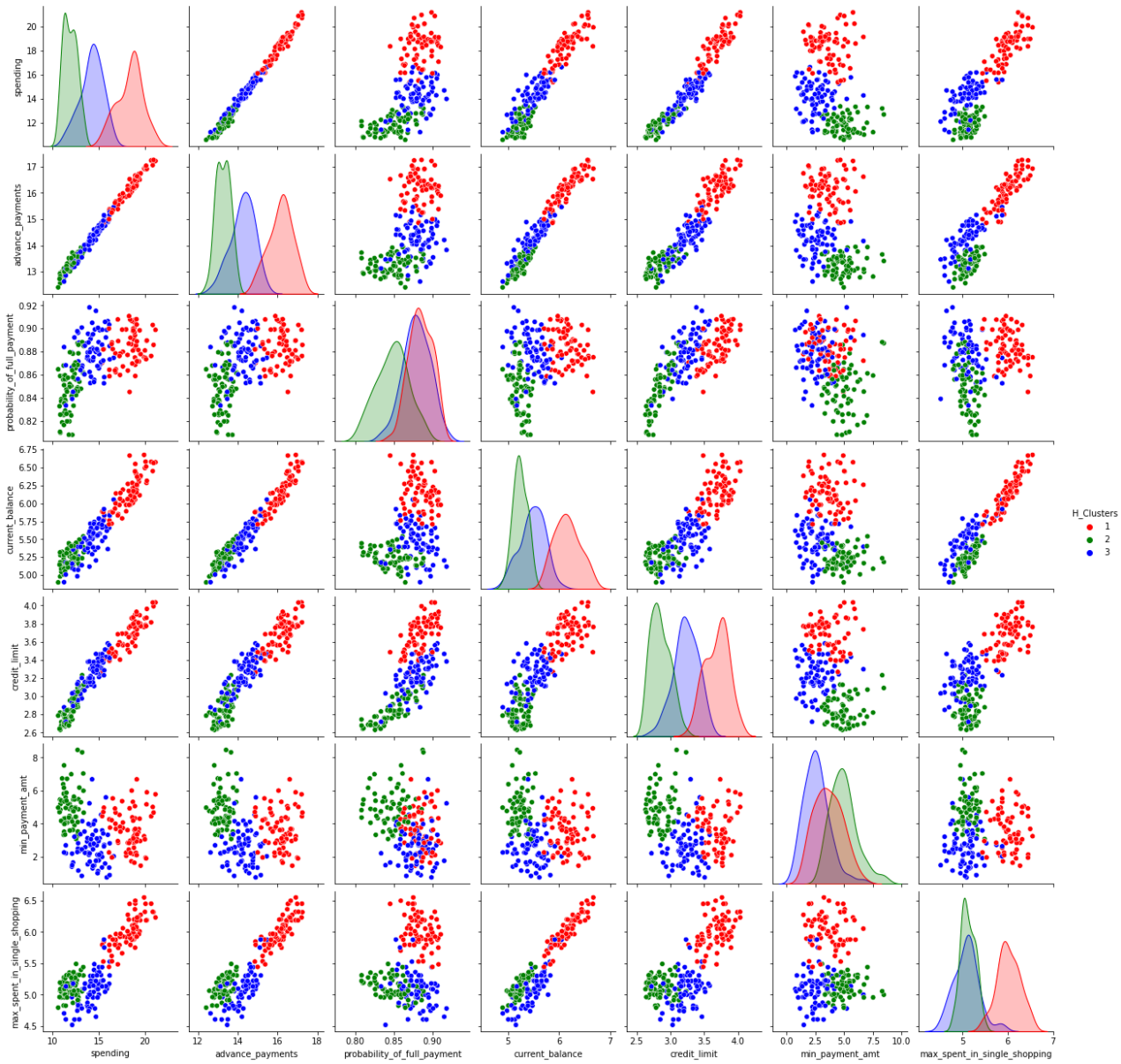

Figure 7: Pair plot for all variable combinations using cluster as hue to understand their behaviour

From the figure above, we can say that,

- Cluster 1 has higher values for all variable combinations followed by cluster 3 followed by cluster 2, from which we can infer that,
  - Cluster 1: High performing customers.
  - Cluster 2: Low performing/new customers.
  - Cluster 3: Medium performing customers.

Q1.4) Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Explain the results properly. Interpret and write inferences on the finalized clusters.

K-means Clustering -

K-means clustering is an unsupervised learning algorithm whose goal is to find groups or assign the data points to clusters on the basis of their similarity. Which means the points in same cluster are similar to each other and in different clusters are dissimilar with each other.

Calculation of within sum of squares for a range of values (1-15) of number of clusters

```
[1469.9999999999998,
 659.171754487041,
 430.6589731513006,
 371.38509060801096,
 327.21278165661346,
 289.31599538959495,
 262.98186570162267,
 241.81894656086033,
 223.91254221002725,
 206.39612184786694,
 193.2835133180646,
 182.97995389115258,
 175.11842017053073,
 166.02965682631788]
```

Elbow Method -

It is a method to find the optimal no. of clusters(k) in the process of clustering. This method is based of plotting the value of cost function against different values of k. As the number of clusters increase, lesser number of points fall within clusters or around the centroids. Hence, the average distortion decreases with the increase of number of clusters. The point where the distortion declines most is said to be the elbow point and define the optimal number of clusters for dataset.

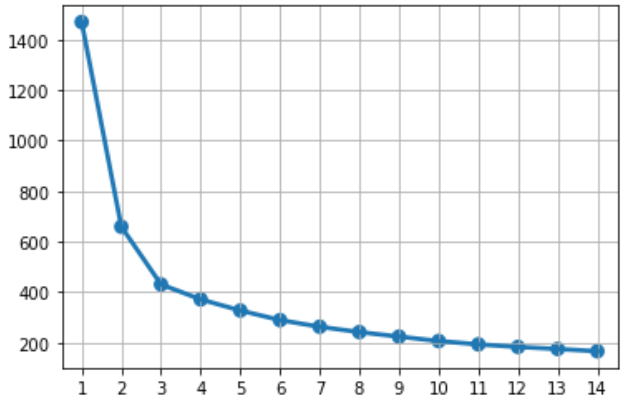

Figure 8: WSS Plot

From the figure above, we can say that,

- The elbows of distortion declines are formed at k=2 & k=3.
- We will be evaluating for both k=2 & k=3 in order to find optimum number of clusters using silhouette score.

Cluster evaluation for k=2 & k=3 using Silhouette Score
We can calculate sil-width for each observation, and when we take the average of sil-widths that is called as silhouette score for a dataset.

spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	H_Clusters	sil_width
19.94	16.92	0.8752	6.675	3.763	3.252	6.550	1	0.603797
15.99	14.89	0.9064	5.363	3.582	3.336	5.144	2	0.008748
18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1	0.678038
10.83	12.96	0.8099	5.278	2.641	5.182	5.185	2	0.495306
17.99	15.86	0.8992	5.890	3.694	2.068	5.837	1	0.548072

Table 9: Sample of the dataset with sil-width for each observation mentioned

- If the silhouette score is close to +1 then we can say the clusters are well separated from each other on an average.
- If the silhouette score is close to 0, then we can say the clusters are not separated from each other.
- If the silhouette score is close to -1 then we can say the model has done a blunder in terms of clustering the data.

Silhouette score for k=2

```
1 silhouette_score(df_scaled1,labels_2)
```

```
0.46577247686580914
```

Silhouette score for k=3

```
1 silhouette_score(df_scaled1,labels_3)
```

```
0.4007270552751299
```

Silhouette sample minimum for k=2

```
1 silhouette_samples(df_scaled1,labels_2).min()
```

```
-0.006171238927461077
```

Silhouette sample minimum for k=3

```
1 silhouette_samples(df_scaled1,labels_3).min()
```

```
0.002713089347678533
```

Silhouette scores above for k=2 & k=3 show that, the optimum number of clusters is 3 as its silhouette score is close to +1 and its silhouette samples minimum value is positive.

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

Table 10: Sample of the dataset after appending clusters to original dataset

Frequency of each cluster in the dataset

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

From the above output we can say that, the number of observations in cluster 1 (72) is more compared to cluster 0 (71) & cluster 2 (67).

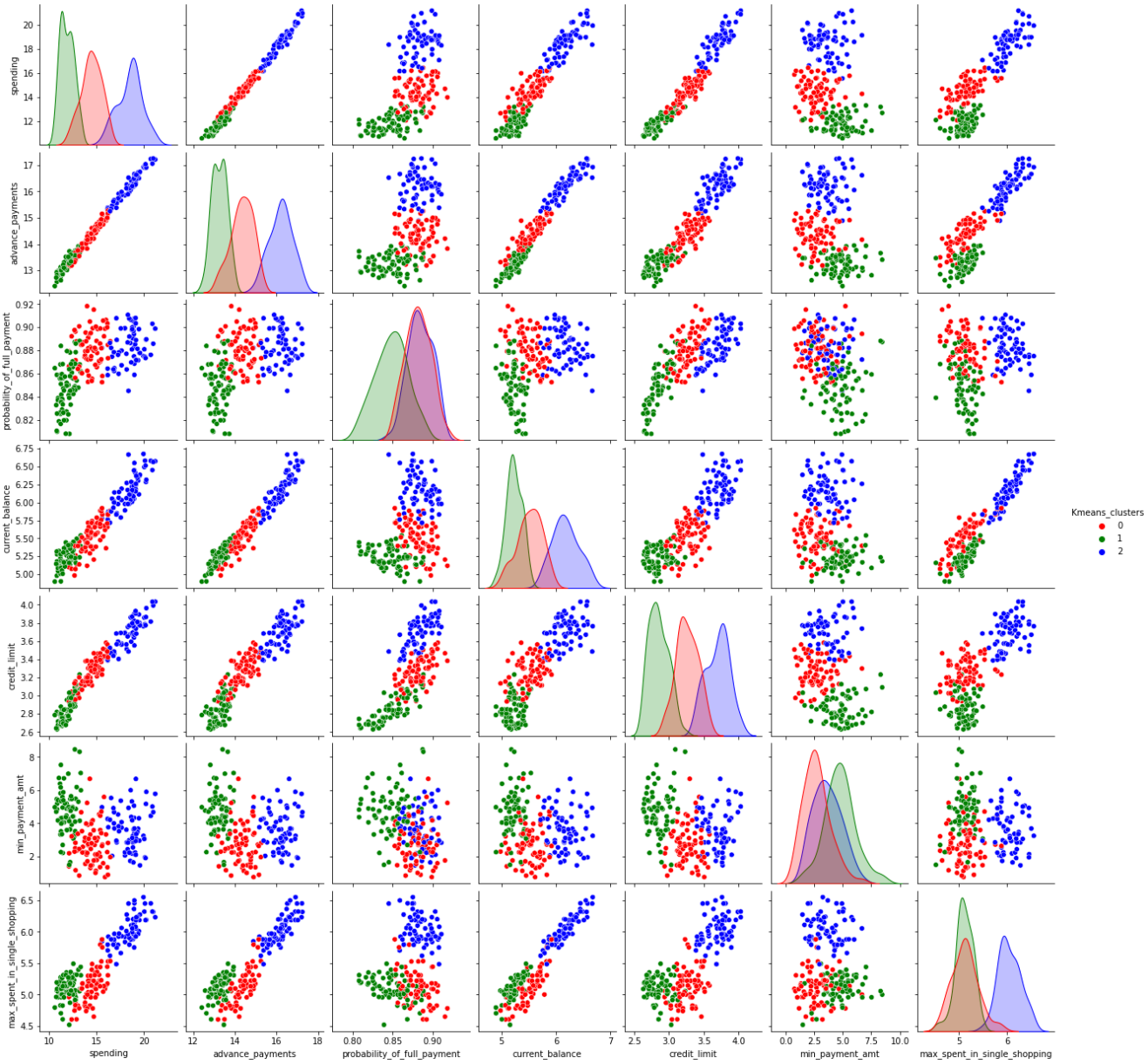

Figure 9: Pair plot for all variable combinations using cluster as hue to understand their behaviour

From the graph above, we can say that,

- Cluster 1 has higher values for all variable combinations compared to cluster 2, from which we can infer that
  - Cluster 0: Medium performing customers.
  - Cluster 1: Low performing/new customers.
  - Cluster 2: High performing customers.

Q1.5) Describe cluster profiles for the clusters defined. Recommend different promotional strategies for different clusters.

Cluster Profiles -

We do cluster profiling by calculating means of every variable for all the observations for every cluster in order to understand its behaviour across all variables.

Profiles of clusters made using Hierarchical Clustering

H_Clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
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

Table 11: Cluster profile of hierarchical method

From the above table, we can say that,

- Cluster 1: Tier 1 customers (High performing customers)
- Cluster 2: Tier 3 customers (Low performing customers/new customers)
- Cluster 3: Tier 2 customers (Medium performing customers)

Comparison of the clusters with respect to variables

- The average amount of money spent per month by the high performing customers is 54.75% more than the low performing or new customers and 29.45% more than the medium performing customers.
- The average amount paid by the high performing customers to the bank in advance by cash is 21.81% more than low performing or new customers and 13.42% more than the medium performing customers.
- The average probability of payment done in full by the customer to the bank is slightly higher for the higher performing customers compared to the medium performing & low performing or new customers (88.44% & 87.91% & 84.80%).
- The average balance amount left in the account is 17.59% more in the account of high performing customers compared to low performing or new customers and 12.43% more than the medium performing customers.
- The average limit of the amount in credit card is 29.57% more for the account of high performing customers compared to low performing or new customers and 14.28% more than the medium performing customers.
- The average minimum amount paid by the customer while making payments for purchases made monthly is 36% more by the low performing or new customers compared to high performing customers and 89.27% more than the medium performing customers.

- The average maximum amount spent in one purchase is 17.38% more by high performing customers compared to low performing or new customers and 18.30% more than the medium performing customers.
- There is not much difference in the number of customers in each category.

Profiles of clusters made using K-means Clustering

Kmeans_clusters	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

Table 12: Cluster Profile of K-means method

From the above table, we can say that,

- Cluster 0: Tier 2 customers (Medium performing customers)
- Cluster 1: Tier 3 customers (Low performing customers/new customers)
- Cluster 2: Tier 1 customers (High performing customers)

Comparison of both clusters with respect to variables

- The average amount of money spent per month by the high performing customers is 56.03% more than the low performing or new customers and 28.13% more than the medium performing customers.
- The average amount paid by the high performing customers to the bank in advance by cash is 22.35% more than low performing or new customers and 13.04% more than the medium performing customers.
- The average probability of payment done in full by the customer to the bank is slightly higher for the higher performing customers compared to the medium performing & low performing or new customers (88.42% & 88.15% & 84.82%).
- The average balance amount left in the account is 17.97% more in the account of high performing customers compared to low performing or new customers and 11.97% more than the medium performing customers.
- The average limit of the amount in credit card is 29.92% more for the account of high performing customers compared to low performing or new customers and 13.53% more than the medium performing customers.
- The average minimum amount paid by the customer while making payments for purchases made monthly is 30.57% more by the low performing or new customers compared to high performing customers and 75.55% more than the medium performing customers.
- The average maximum amount spent in one purchase is 18.43% more by high performing customers compared to low performing or new customers and 17.96% more than the medium performing customers.
- There is not much difference in the number of customers in each category.

From comparing the clusters formed using hierarchical & K-means method, we come to know that, there is very slight variation in difference of means for all variables in the dataset. This might be due to the change in number of observations in each cluster for both methods.

Some Recommendations -

- Since, the spending per month is much larger for high performing customers, after a certain amount of money spent special offers, cashback, and discounts could be given to them which in turn would increase the overall credit consumption by the customers which would increase overall business of the bank.
- Those customers in all the categories who have been using their credit limit to the fullest and been paying the amount back on time could be provided with an increase in their credit limit.
- Since, the minimum amount paid by customers while making payments is larger for low performing/new customers, the customers who are on the borderline of low & high performing customer range can be provided with certain benefits, offers and discounts depending on the amount of money spent which improves their credit performance, which could lead for them to convert to high performing customers.
- A virtual point system for all the categories could be used in which customers get certain amount of points after certain amount of credit usage. The points gained can be used by the customers to redeem for products or certain coupons which might prove beneficial to them.
  The advantages of this are twofold-
  - More credit usage by existing customers, thereby increasing credit activity which leads to better business for the bank.
  - It could attract new customers as well to get into using our credit cards.

Case Study 2- Problem 2: CART-RF-ANN

Overview

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.

Summary:

This business report provides detailed explanation on the approach to each problem definition, solution to those the problems provide some key insights/recommendations to the business.

Q2.1) Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis).

	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

Table 13: Sample of the insurance dataset

```
<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
```

Table 14: Basic Information of the dataset

	count	unique	top	freq	mean	std	min	5%	10%	25%	50%	75%	90%	95%	max
Age	2861	NaN	NaN	NaN	38.2041	10.6781	8	24	26	31	36	43	53	61	84
Agency_Code	2861	4	EPX	1238	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Type	2861	2	Travel Agency	1709	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Claimed	2861	2	No	1947	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Commision	2861	NaN	NaN	NaN	15.081	25.8268	0	0	0	0	5.63	17.82	50.25	63.21	210.21
Channel	2861	2	Online	2815	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Duration	2861	NaN	NaN	NaN	72.1202	135.977	-1	4	6	12	28	66	239	367	4580
Sales	2861	NaN	NaN	NaN	61.7579	71.3997	0	10	13	20	33.5	69.3	178	230	539
Product Name	2861	5	Customised Plan	1071	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Destination	2861	3	ASIA	2327	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 15: Summary of the dataset

Observations -

- The above dataset contains 3000 rows & 10 columns (before removing duplicates) and 2861 rows & 10 columns (after removing duplicates).
- The variables 'Age', 'Commision', 'Duration' and 'Sales' are of numeric datatype (int and float), whereas the remaining six variables are of object dataype.
- There are no null, missing and bad values in the dataset.
- There are duplicates values in the datasets, which are very less in number compared to the size of the dataset. Hence, they can be removed for further processing and avoid bias in analysis.

Univariate Analysis

1) Boxplot for outlier identification

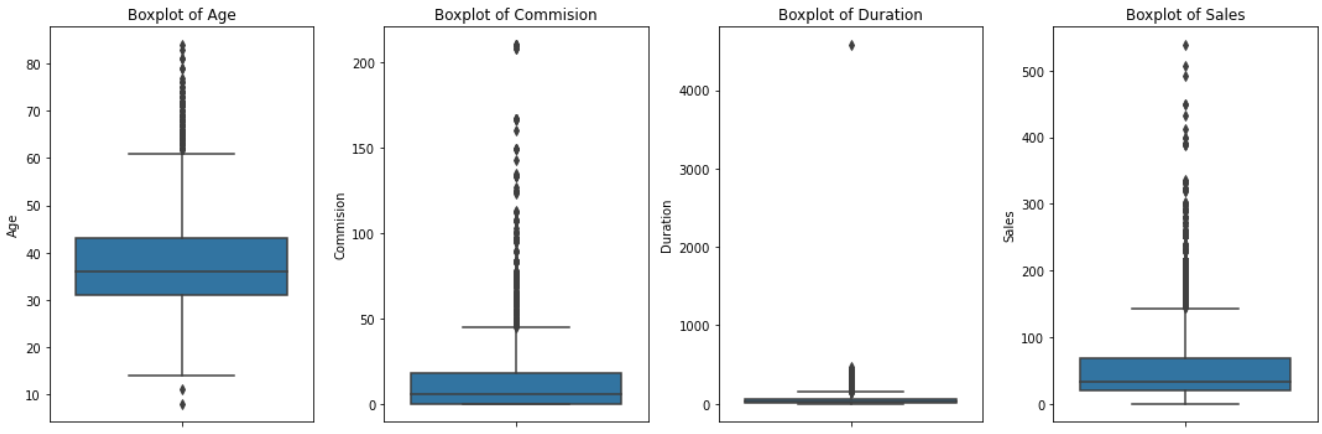

Figure 10: Boxplot for Outlier Identification

From the graph above, we can clearly say that,

All the variables /columns have decent number of outliers within them. Treating outliers sometimes results in the models having better performance, but the models lose out on the generalization. Hence, we will be proceeding further without treating them.

2) Distplot for studying variable distribution

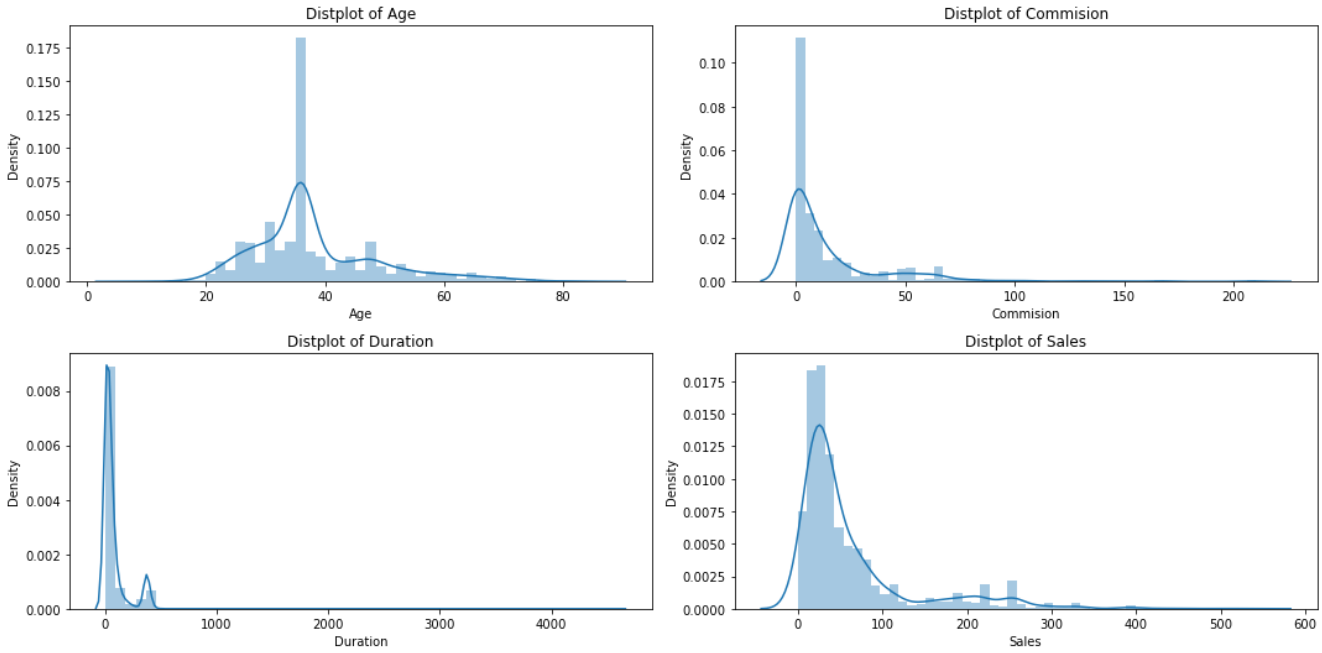

Figure 11: Distribution Check for all variables

Checking skew values for all variables

```
Age          1.103145
Commision     3.104741
Duration    13.786096
Sales        2.344643
dtype: float64
```

From the figure 11 & skew values above, we can say that, since, the skewness value of all the variables/columns is greater than +1, they all are highly skewed in the right.

Multivariate Analysis

1) Heatmap to study correlation between variables

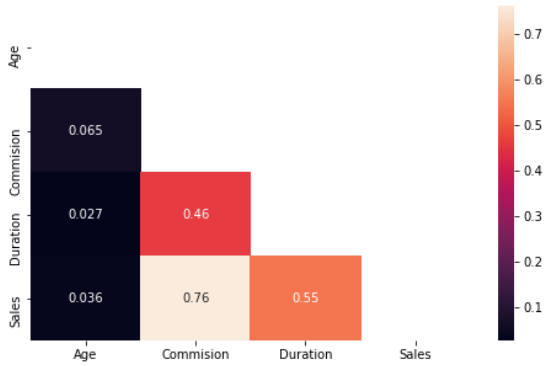

Figure 12: Correlation Heatmap

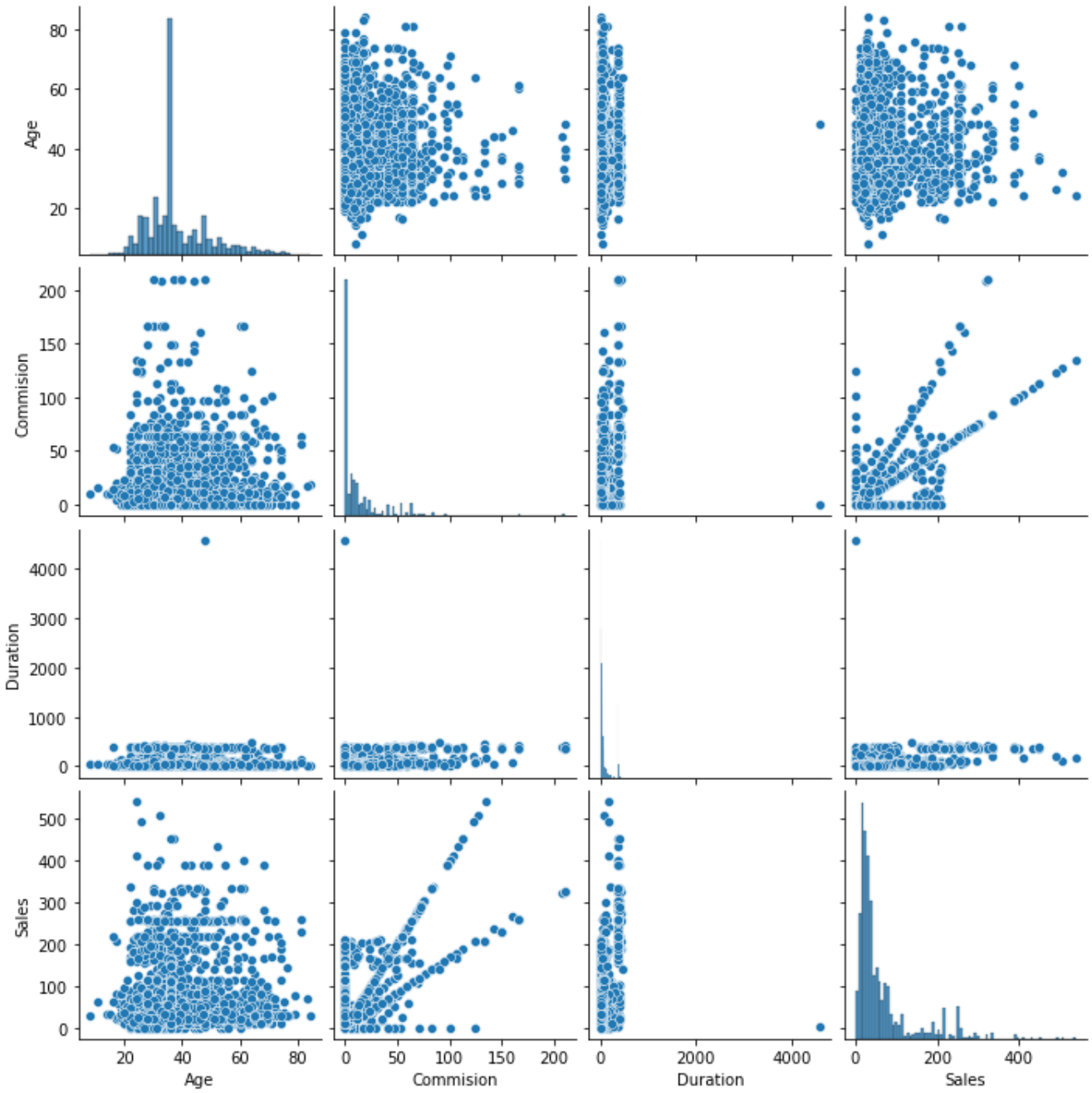

Figure 13: Pair plot for all variable combinations

Observations:

- There is some correlation between the variables 'Sales' & 'Commision' (0.76).
- There is a weak correlation between the variables 'Duration' & 'Commision' (0.46) and 'Duration' and 'sales' (0.55).
- Pairplot verifies everything the heatmap shows with graphical visuals.

2) Understanding Target variable 'Claimed' with different Categorical Variables

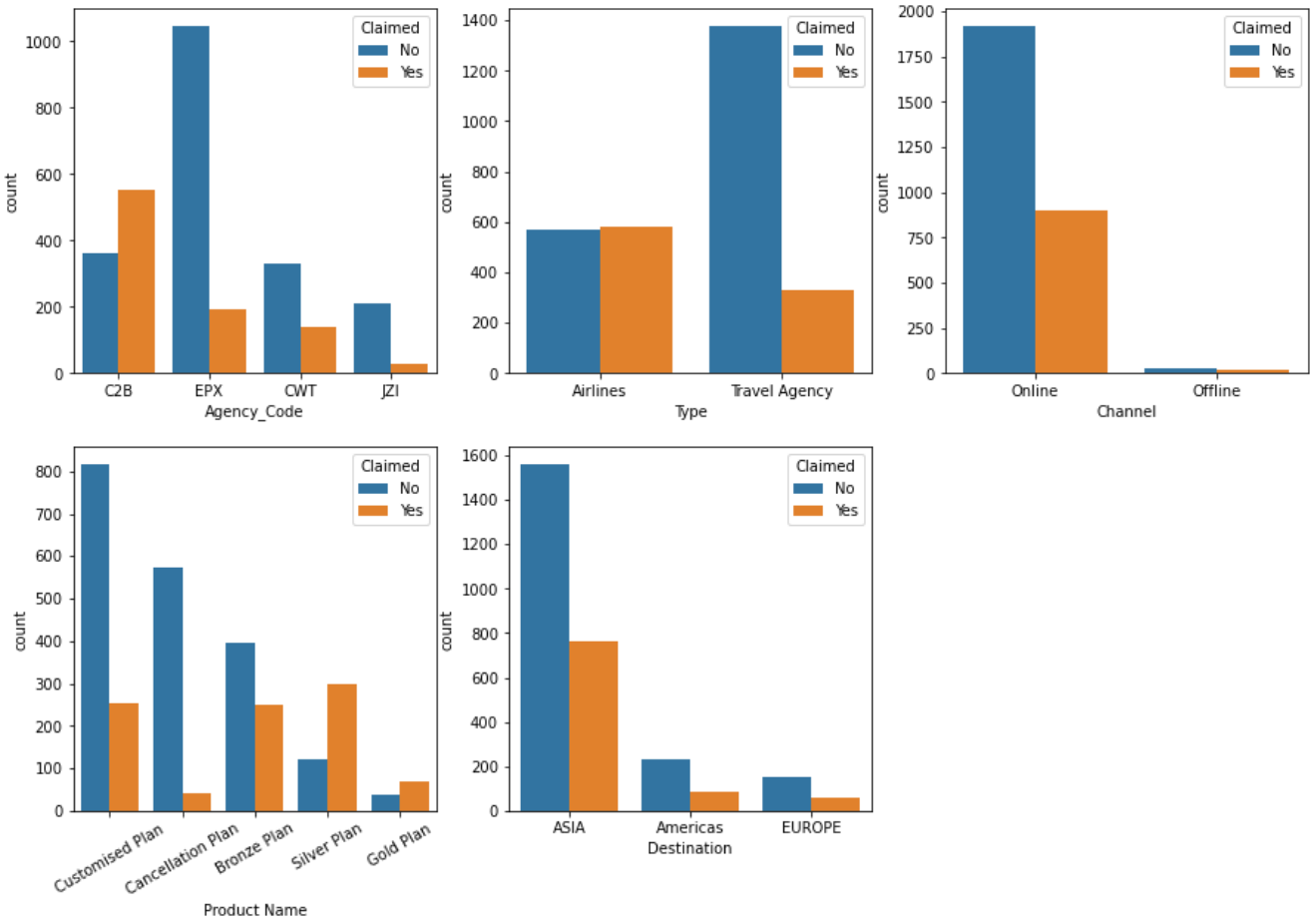

Figure 14: Target variable 'Claimed' with different Categorical Variables

From the figure above, we can say that,

- Agency C2B has the highest number of insurance claims. Whereas, JZI has the lowest.
- Airlines type of insurance has same number of claimed and unclaimed insurances. Whereas, travel agency type has a greater number of unclaimed.
- Total number of insurances (claimed + unclaimed) is substantially greater in online channel compared to offline channel.
- Customized, Cancellation and Bronze plans have a greater number of unclaimed insurances compared to claimed. Whereas, Silver and Gold plans have a greater number of claimed insurances compared to unclaimed.
- Total number of insurances (claimed + unclaimed) is substantially greater in Asian Destinations compared to Americas & Europe.

Q2.2) Data Split: Split the data into test and train, build classification model CART, Random Forest, Artificial Neural Network

In order to build any model for prediction we need to make sure all the object datatype variables are converted into int or categorical variable.

```
<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
```

Table 16: Dataset information after datatype conversion

Splitting data into training and test set

Firstly, we extract the target column into separate vectors for training set and test set.
We split the entire data as follows
Training set – 70%
Test set – 30%

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

Above is the python output stating shape (i.e. rows, columns of each subset formed)

Building a Decision Tree Classifier

The parameters used to find the optimum ones are
Criterion – gini
Max depth – 5, 10, 15, 20
Min samples leaf - 25, 50, 75
Min samples split - 150, 300, 450
The value of used random state = 1.
The best parameter values were found using grid search. Below is the python code output,

```
{'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 50, 'min_samples_split': 300}
```

```
DecisionTreeClassifier(max_depth=5, min_samples_leaf=50, min_samples_split=300,
                       random_state=1)
```

Building a Random Forest Classifier

The parameters used to find the optimum ones are

Max depth - 5, 10, 15

Max features - 4, 5, 6

Min samples leaf - 5, 10, 15

Min samples split - 50, 60, 70

N estimators - 100, 200, 300

The value of used random state = 1.

The best parameter values were found using grid search. Below is the python code output,

```
RandomForestClassifier(max_depth=15, max_features=5, min_samples_leaf=5,
                       min_samples_split=60, random_state=1)
```

Building an Artificial Neural Network

The parameters used to find the optimum ones are

Hidden layer sizes - 50, 100, 150

Max iterations - 250, 500, 750

Solver - adam, sgd

Tolerance - 0.01

The value of used random state = 1.

The best parameter values were found using grid search. Below is the python code output,

```
MLPClassifier(hidden_layer_sizes=100, max_iter=250, random_state=1, tol=0.01)
```

Q2.3) Performance Metrics: Comment and Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score, classification reports for each model.

For CART Model

1) AUC and ROC for the training data

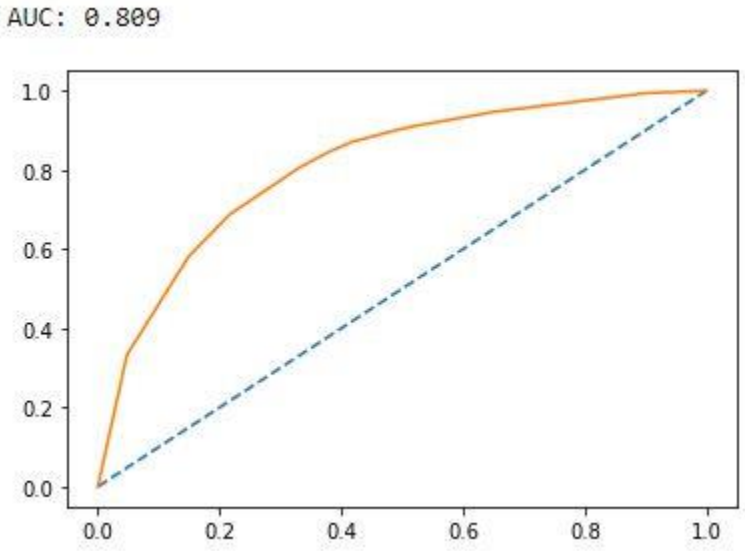

Figure 15: AUC & ROC curve for training data

2) AUC and ROC for the test data

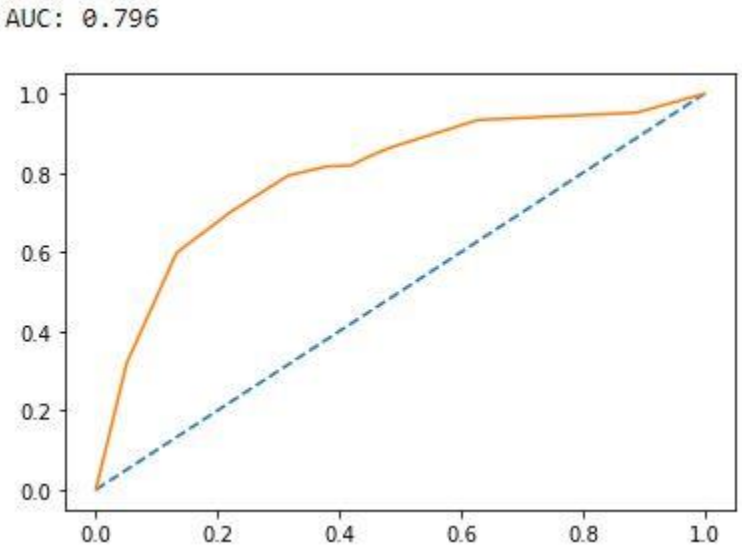

Figure 16: AUC & ROC curve for testing data

3) Accuracy of training & testing data

Training Accuracy – 0.7642357642357642

Testing Accuracy – 0.7823050058207218

4) Confusin Matrix for training data

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

5) Classification Report for training data

	precision	recall	f1-score	support
0	0.81	0.85	0.83	1359
1	0.65	0.58	0.61	643
accuracy			0.76	2002
macro avg	0.73	0.72	0.72	2002
weighted avg	0.76	0.76	0.76	2002

6) Confusin Matrix for testing data

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

7) Classification Report for testing data

	precision	recall	f1-score	support
0	0.82	0.87	0.85	588
1	0.68	0.60	0.63	271
accuracy			0.78	859
macro avg	0.75	0.73	0.74	859
weighted avg	0.78	0.78	0.78	859

CART Conclusion

Train Data:
AUC: 80%
Accuracy: 76%
Precision: 65%
f1-Score: 51%

Test Data:
AUC: 80%
Accuracy: 78%
Precision: 68%
f1-Score: 63%

Training and Test set results are almost similar, and with the overall measures being moderate, the model is an average model.

Agency_Code is the most important variable for predicting insurance claims.

For Random Forest Model

1) AUC and ROC for the training data

```
Area under Curve is 0.8653793556464193
```

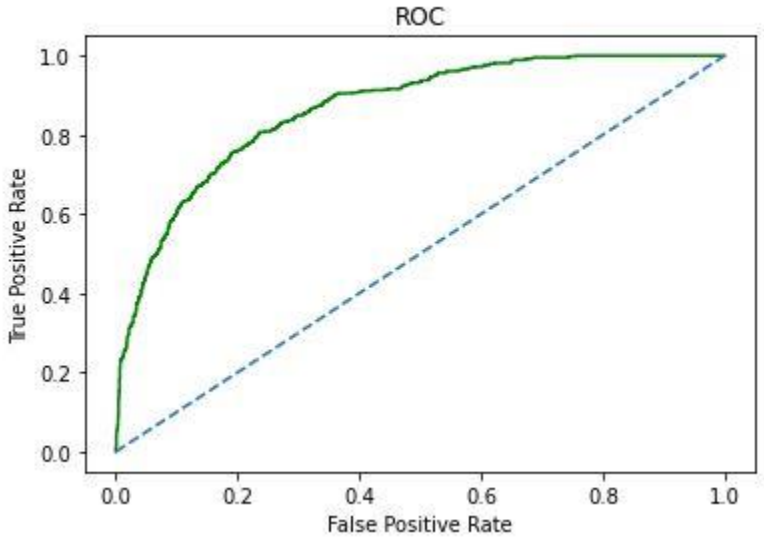

Figure 17: AUC & ROC curve for training data

2) AUC and ROC for the test data

```
Area under Curve is 0.8141394934357506
```

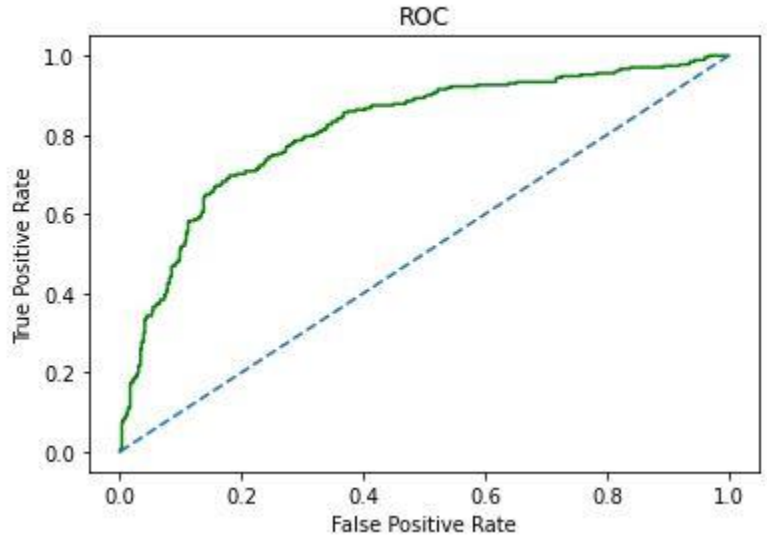

Figure 18: AUC & ROC curve for testing data

3) Accuracy of training & testing data

Training Accuracy –

```
0.8036963036963037
```

Testing Accuracy –

```
0.7834691501746216
```

4) Confusin Matrix for training data

```
array([[1225,  134],
       [ 259,  384]], dtype=int64)
```

5) Classification Report for training data

	precision	recall	f1-score	support
0	0.83	0.90	0.86	1359
1	0.74	0.60	0.66	643
accuracy			0.80	2002
macro avg	0.78	0.75	0.76	2002
weighted avg	0.80	0.80	0.80	2002

6) Confusin Matrix for testing data

```
array([[523,  65],
       [121, 150]], dtype=int64)
```

7) Classification Report for testing data

	precision	recall	f1-score	support
0	0.81	0.89	0.85	588
1	0.70	0.55	0.62	271
accuracy			0.78	859
macro avg	0.75	0.72	0.73	859
weighted avg	0.78	0.78	0.78	859

Random Forest Conclusion

Train Data:
AUC: 87%
Accuracy: 80%
Precision: 74%
f1-Score: 66%

Test Data:
AUC: 81%
Accuracy: 78%
Precision: 70%
f1-Score: 62%

Training and Test set results are almost similar, and with the overall measures being moderate, the model is an average model.

Agency_Code is the most important variable for predicting insurance claims.

For Neural Network Model

1) AUC and ROC for the training data

```
Area under Curve is 0.7920464571767961
```

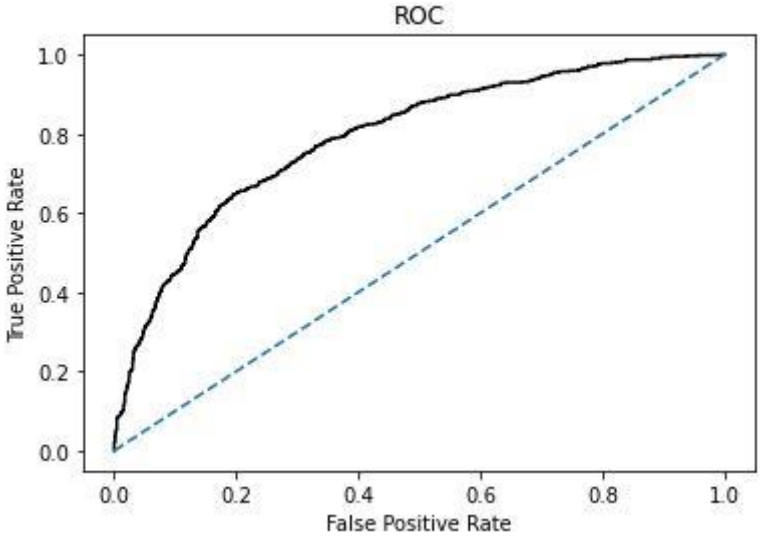

Figure 19: AUC & ROC curve for training data

2) AUC and ROC for the test data

```
Area under Curve is 0.7911175540326832
```

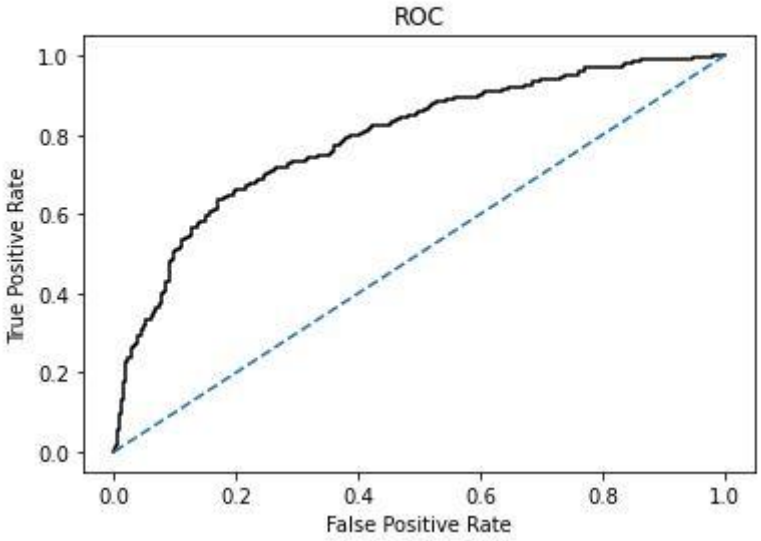

Figure 20: AUC & ROC curve for testing data

3) Accuracy of training & testing data

Training Accuracy –

```
0.7622377622377622
```

Testing Accuracy –

```
0.7718277066356228
```

4) Confusin Matrix for training data

```
array([[1163,  196],
       [ 280,  363]], dtype=int64)
```

5) Classification Report for training data

	precision	recall	f1-score	support
0	0.81	0.86	0.83	1359
1	0.65	0.56	0.60	643
accuracy			0.76	2002
macro avg	0.73	0.71	0.72	2002
weighted avg	0.76	0.76	0.76	2002

6) Confusin Matrix for testing data

```
array([[510,  78],
       [118, 153]], dtype=int64)
```

7) Classification Report for testing data

	precision	recall	f1-score	support
0	0.81	0.87	0.84	588
1	0.66	0.56	0.61	271
accuracy			0.77	859
macro avg	0.74	0.72	0.72	859
weighted avg	0.76	0.77	0.77	859

Neural Network Conclusion

Train Data:
AUC: 79%
Accuracy: 76%
Precision: 65%
f1-Score: 60%

Test Data:
AUC: 79%
Accuracy: 77%
Precision: 66%
f1-Score: 61%

Training and Test set results are almost similar, and with the overall measures being moderate, the model is an average model.

Agency_Code is the most important variable for predicting insurance claims.

Q2.4) Final Model: Compare all the models and write an inference which model is best/optimized.

Comparison of the performance metrics from the 3 models

	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.76	0.78	0.80	0.78	0.76	0.77
AUC	0.81	0.80	0.87	0.81	0.79	0.79
Recall	0.58	0.60	0.60	0.55	0.56	0.56
Precision	0.65	0.68	0.74	0.70	0.65	0.66
F1 Score	0.61	0.63	0.66	0.62	0.60	0.61

Table 17: Comparision table of the performance metrics from the 3 models

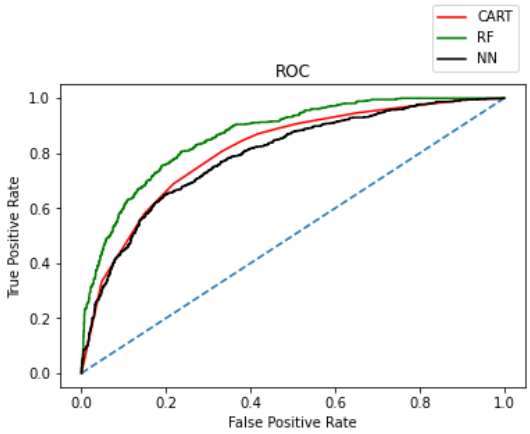

Figure 21: AUC & ROC comparision for all 3 models for training data

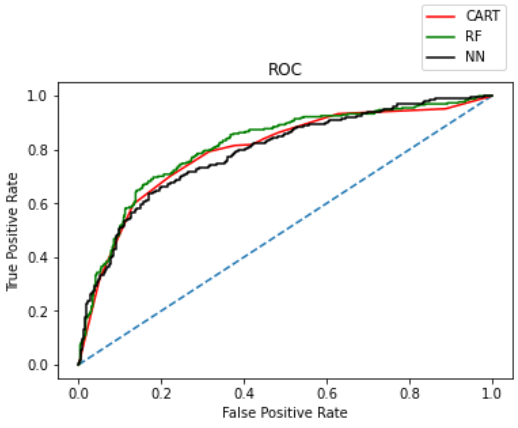

Figure 22: AUC & ROC comparision for all 3 models for testing data

Observations -

- Out of the 3 models, Random Forest has slightly better performance than the Cart and Neural network model.
- Overall all the 3 models are moderately stable to be used for making any future predictions.

Q2.5) Inference: Based on the whole Analysis, what are the business insights a nd recommendations

Business Insights -

- Overall, all the 3 models are moderately stable to be used for making any future predictions.
- From Cart and Random Forest Model, the variable Agency_Code is found to be the most useful feature amongst all other features for predicting if an agency has claimed insurance or not and product names is very slightly useful but could be useful for predictions.
- If Agency_Code is C2B, then those have more chances of claiming insurance.
- And if it is EPX, CWT or JZI then those have more chances of not claiming insurance.
- If the product names are customised, cancellation and bronze plans, they are more likely to have higher chances of no insurance claims, whereas, if the names are gold and silver plans, they have a higher chance of having insurance claims.

Recommendations -

- We know agency code is the most important feature for insurance claim predictions & product name somewhat important,
- For whichever Agency code the insurance claim rate is higher,
    - We could charge more premiums in exchange for insurance coverage where the claim rates are higher (C2B), then reinvesting those premiums into other interest-generating assets.
    - Reducing and sealing the claim amount so that not much money is spent on insurance claims.
    - Increasing the number of insurances done wherein there are low claiming rates (EPX, CWT or JZI), which in turn would lead to increase in revenue and more to use for the company for investments.
- For whichever product name/ insurance plan type
    - We could charge more premiums in exchange for insurance coverage where the claim rates are higher (Gold and Silver plans), then reinvesting those premiums into other interest-generating assets.
    - Reducing and sealing the claim amount so that not much money is spent on insurance claims.
    - Increasing the number of insurances done wherein there are low claiming rates (customised, cancellation and bronze plans), which in turn would lead to increase in revenue and more to use for the company for investments.