PGP - DSBA

Data Mining

Project Report – September 2022

Shruti Jha
9-21-2022

Contents

List of Tables

List of Figures

Problem 1 – Clustering

Introduction

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. The purpose of this case study is to identify the segments based on credit card usage.

Data Dictionary for Market Segmentation

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

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

1.1.1 Sample of dataset

Here are the top 5 rows (sample) of the dataset:

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

Table 1. 1: Dataset Sample

- Dataset has 7 variables.
- As mentioned in the Data Dictionary, most of the variables have some units assigned to them (100s, 1000s etc). For the sake of further analysis of the data, the values have been converted to their true forms. This is how the data appears now:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	19940.0	1692.0	0.8752	6675.0	37630.0	325.2	6550.0
1	15990.0	1489.0	0.9064	5363.0	35820.0	333.6	5144.0
2	18950.0	1642.0	0.8829	6248.0	37550.0	336.8	6148.0
3	10830.0	1296.0	0.8099	5278.0	26410.0	518.2	5185.0
4	17990.0	1586.0	0.8992	5890.0	36940.0	206.8	5837.0

Table 1. 2: Transformed Dataset Sample

1.1.2 Check for Duplicate Records

```
Number of duplicate records: 0
```

1.1.3 Types of variables in the dataset

```
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 in numeric (float64) format.
- There are a total of 210 rows and 7 columns in the dataset.

1.1.4 Missing values in the dataset

```
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
```

From the above results we can say that there is no missing value present in the dataset.

1.1.5 Descriptive Statistics

Describe function provides a table indicating the count of variables, mean, standard deviation and other values for the 5-point summary that includes (min, 25%, 50%, 75% and max). 50% in the table is also known as median.

	count	mean	std	min	25%	50%	75%	max
spending	210.0	14847.523810	2909.699431	10590.0000	12270.0000	14355.00000	17305.000000	21180.0000
advance_payments	210.0	1455.928571	130.595873	1241.0000	1345.0000	1432.00000	1571.500000	1725.0000
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.8569	0.87345	0.887775	0.9183
current_balance	210.0	5628.533333	443.063478	4899.0000	5262.2500	5523.50000	5979.750000	6675.0000
credit_limit	210.0	32586.047619	3777.144449	26300.0000	29440.0000	32370.00000	35617.500000	40330.0000
min_payment_amt	210.0	370.020095	150.355713	76.5100	256.1500	359.90000	476.875000	845.6000
max_spent_in_single_shopping	210.0	5408.071429	491.480499	4519.0000	5045.0000	5223.00000	5877.000000	6550.0000

Table 1. 3: Data Description

From the above descriptive statistics, we can infer:

- On an average, customers spend INR 14847.52 per month.
- Advance payments done by the customers ranges between INR 1241.00 and 1725.00.
- The average probability of full payment made by the customer to the bank is 0.870999 (87.09%); the highest probability is 0.9183 (92%) and the lowest probability is 0.8081 (81%).
- If we observe the values across the different features, we see in most of the cases the mean and median seem to be very near to each other, indicating that the shape of all the numerical values seem to be more or less normally distributed.
- The highest spending customer (INR 21180.00) has made advance payment of INR 1721.00 and has INR 6573.00 as current_balance. Also, the probability of full payment by that customer is very close to 90%.

spending	advance_payments	probability_of_full_payment	current_balance
21180.0	1721.0	0.8989	6573.0

Table 1. 4: High Spending Customer

- The lowest spending customer (INR 10590.00) has made advance payment of INR 1241.00 and has INR 4899.00 currently in the bank account. The probability of full payment by that customer is 86.48%.

spending	advance_payments	probability_of_full_payment	current_balance
10590.0	1241.0	0.8648	4899.0

Table 1. 5: Low Spending Customer

- Customer who spent the highest maximum amount in one purchase (INR 6550.00) also has highest current_balance (INR 6675.00).

current_balance	max_spent_in_single_shopping
6675.0	6550.0

Table 1. 6: Customer Current Balance

1.1.6 Check for outliers

To check for outliers, box plots have been plotted:

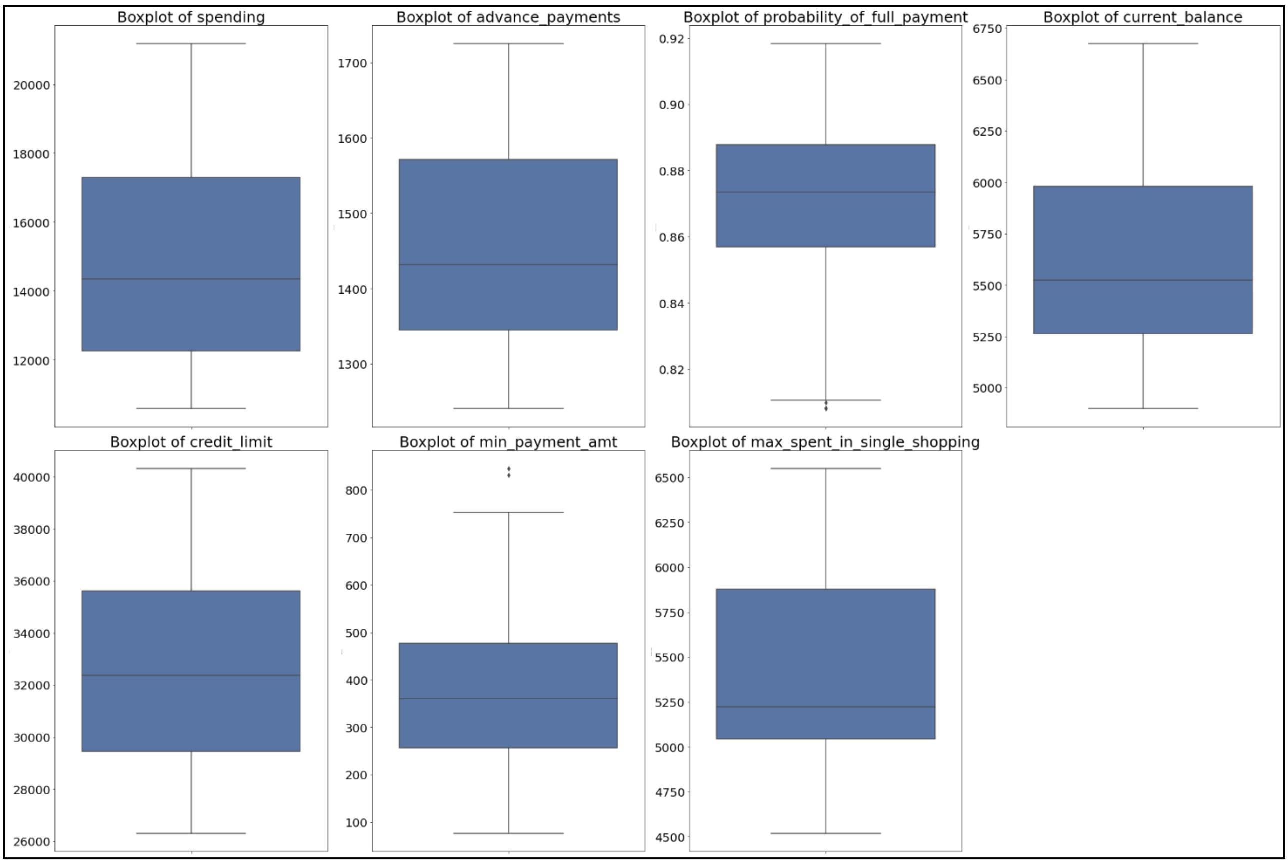


Figure 1. 1: Boxplot for Outliers

- The small dots outside the whiskers of boxplots denote outliers. As we can infer from the above plot, only 'probability_of_full_payment' and 'min_payment_amt' columns have outliers / extreme values present in them.
- Records with outliers in 'probability_of_full_payment' column:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
3	10830.0	1296.0	0.8099	5278.0	26410.0	518.2	5185.0
77	12130.0	1373.0	0.8081	5394.0	27450.0	482.5	5220.0
189	11750.0	1352.0	0.8082	5444.0	26780.0	437.8	5310.0

Table 1. 7: Outliers in Probability Field

- Records with outliers in 'min_payment_amt' column:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
5	12700.0	1341.0	0.8874	5183.0	30910.0	845.6	5000.0
89	13200.0	1366.0	0.8883	5236.0	32320.0	831.5	5056.0

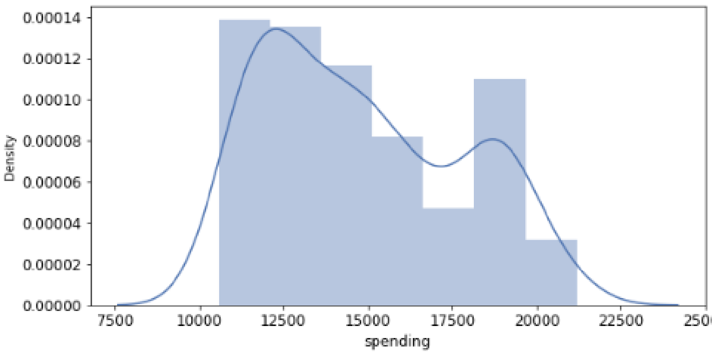
Table 1. 8: Outliers in Minimum Payment Field

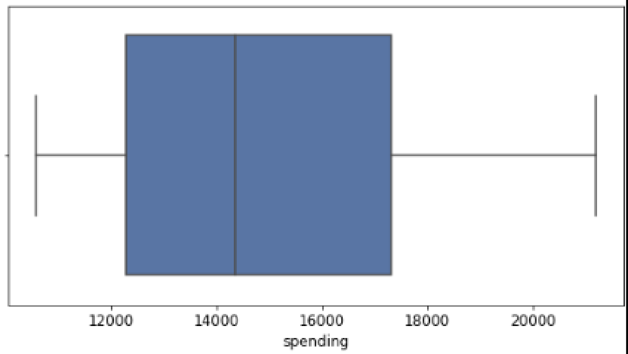
- Clustering results are sensitive to outliers. Hence, outlier treatment has been performed by imputing extreme values with the lower limit (Q1 – 1.5*IQR) and upper limit (Q3 + 1.5*IQR) of the respective variables.

1.1.7 Univariate analysis

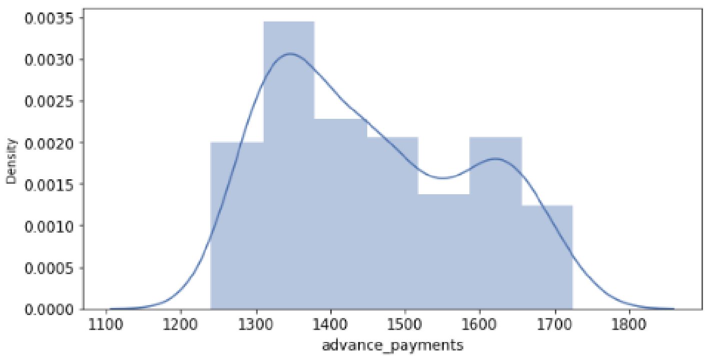
Univariate analysis is performed for all the numeric variables individually to display their statistical description. Visualized the variables using distplot to view the distribution and the box plot to view 5-point summary and outliers if any.

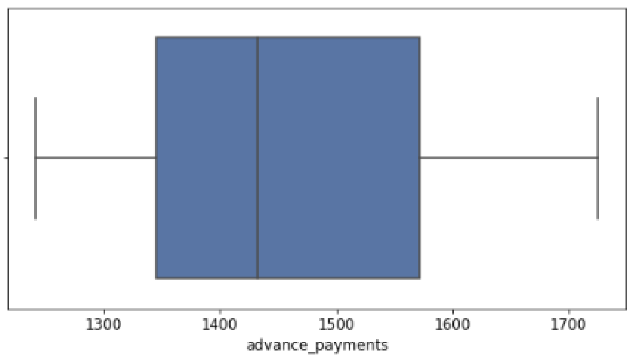
```
Description of spending
.........................................................................
count      210.000000
mean     14847.523810
std       2909.699431
min      10590.000000
25%      12270.000000
50%      14355.000000
75%      17305.000000
max      21180.000000
Name: spending, dtype: float64

 Distribution of spending                                          Countplot of spending
.................................................................................................................................
```





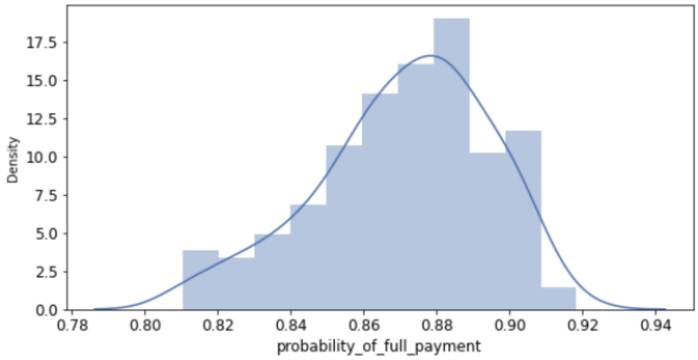

```
Description of advance_payments
.........................................................................
count     210.000000
mean     1455.928571
std       130.595873
min      1241.000000
25%      1345.000000
50%      1432.000000
75%      1571.500000
max      1725.000000
Name: advance_payments, dtype: float64

 Distribution of advance_payments                                          Countplot of advance_payments
.................................................................................................................................
```

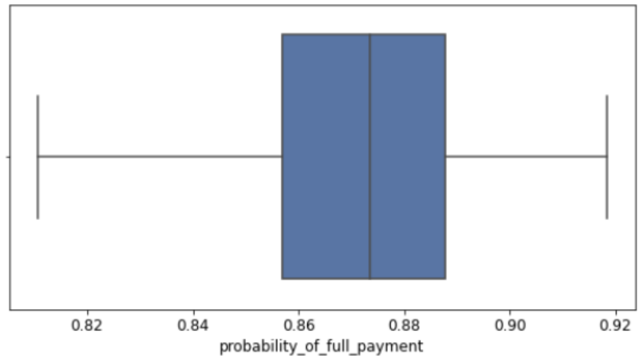




```
Description of probability_of_full_payment
.........................................................................
count    210.000000
mean       0.871025
std        0.023560
min        0.810588
25%        0.856900
50%        0.873450
75%        0.887775
max        0.918300
Name: probability_of_full_payment, dtype: float64
```

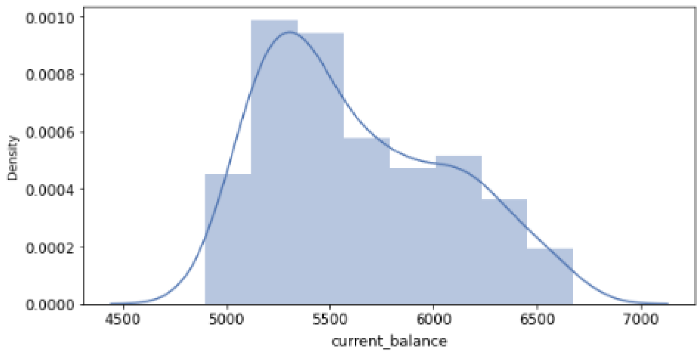
```
 Distribution of probability_of_full_payment                                    Countplot of probability_of_full_payment
..................................................................................................................................
```

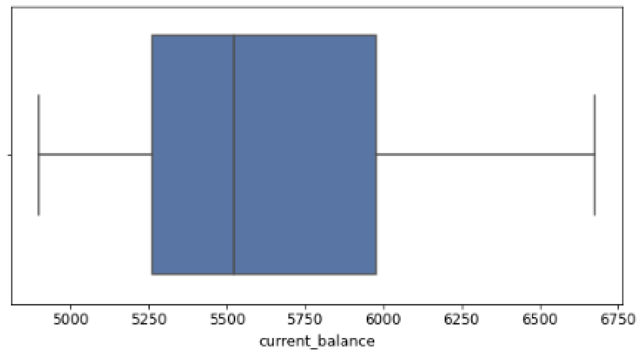





```
Description of current_balance
.........................................................................
count     210.000000
mean     5628.533333
std       443.063478
min      4899.000000
25%      5262.250000
50%      5523.500000
75%      5979.750000
max      6675.000000
Name: current_balance, dtype: float64
```

```
 Distribution of current_balance                                    Countplot of current_balance
..................................................................................................................................
```

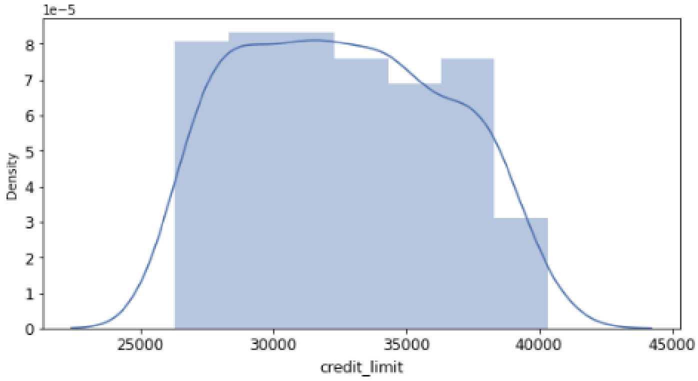


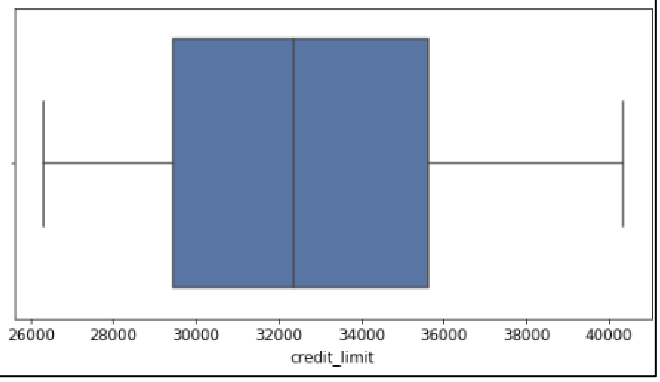


```
Description of credit_limit
......................................................................
count      210.000000
mean     32586.047619
std       3777.144449
min      26300.000000
25%      29440.000000
50%      32370.000000
75%      35617.500000
max      40330.000000
Name: credit_limit, dtype: float64
```

Distribution of credit_limit　　　　Countplot of credit_limit

....

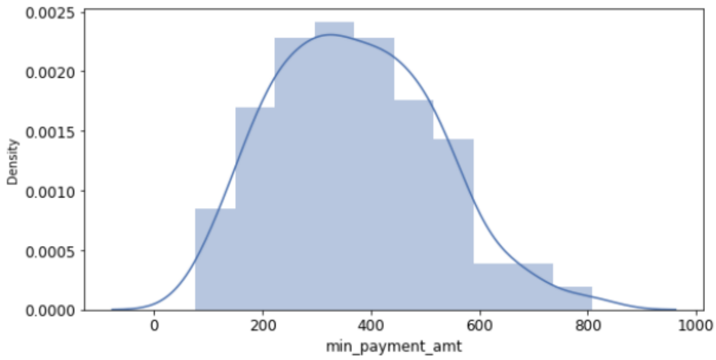


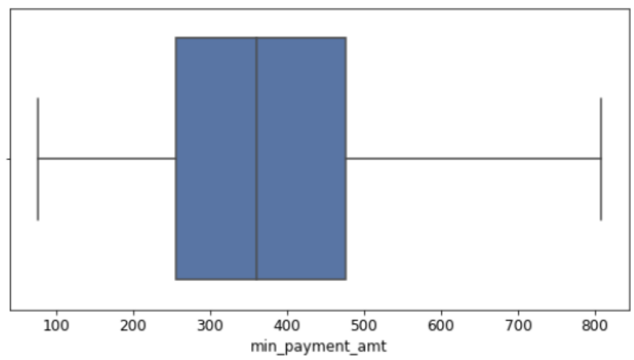



```
Description of min_payment_amt
......................................................................
count    210.000000
mean     369.728786
std      149.468900
min       76.510000
25%      256.150000
50%      359.900000
75%      476.875000
max      807.962500
Name: min_payment_amt, dtype: float64
```

Distribution of min_payment_amt　　　　Countplot of min_payment_amt

....





```
Description of max_spent_in_single_shopping
..........................................................................
count     210.000000
mean     5408.071429
std       491.480499
min      4519.000000
25%      5045.000000
50%      5223.000000
75%      5877.000000
max      6550.000000
Name: max_spent_in_single_shopping, dtype: float64

 Distribution of max_spent_in_single_shopping                                    Countplot of max_spent_in_single_shopping
..................................................................................................................................
```

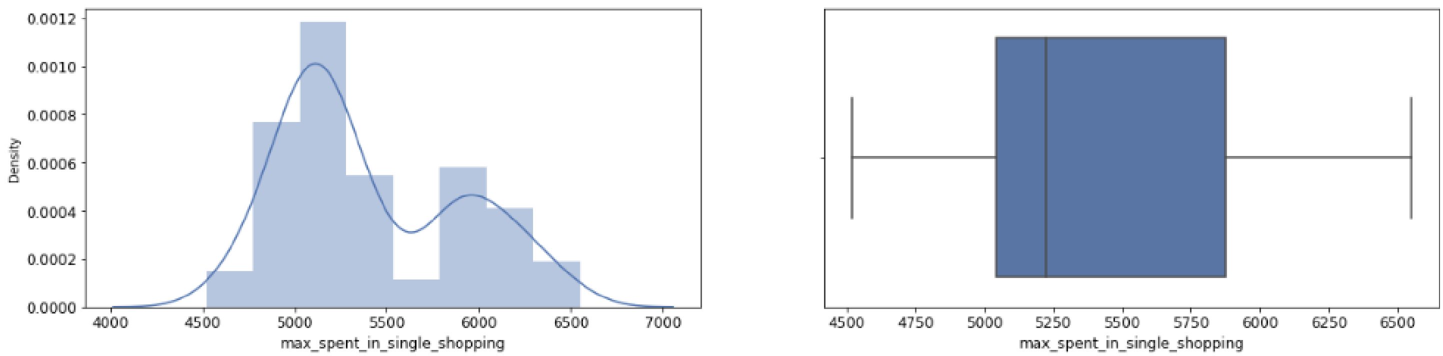


Figure 1. 2: Univariate Analysis

	Kurtosis	Skewness
spending	-1.084266	0.399889
advance_payments	-1.106703	0.386573
probability_of_full_payment	-0.186398	-0.522793
current_balance	-0.785645	0.525482
credit_limit	-1.097697	0.134378
min_payment_amt	-0.218796	0.360001
max_spent_in_single_shopping	-0.840792	0.561897

Table 1. 9: Kurtosis & Skewness

Observations

- There are 7 numeric fields in the dataset.
- From the boxplots we can see that there are no outliers present in the data anymore.
- The distribution for ‘spending’, ‘advance_payments’, ‘max_spent_in_single_shopping’ is bimodal.
- The distribution appears to be right/positive skewed for most of the variables; except for ‘probability_of_full_payment’, the data is left/negative skewed for it.
- ‘min_payment_amt’ and ‘credit_limit’ seems to have data that is normally distributed.

1.1.8 Multivariate analysis

Pair plot:

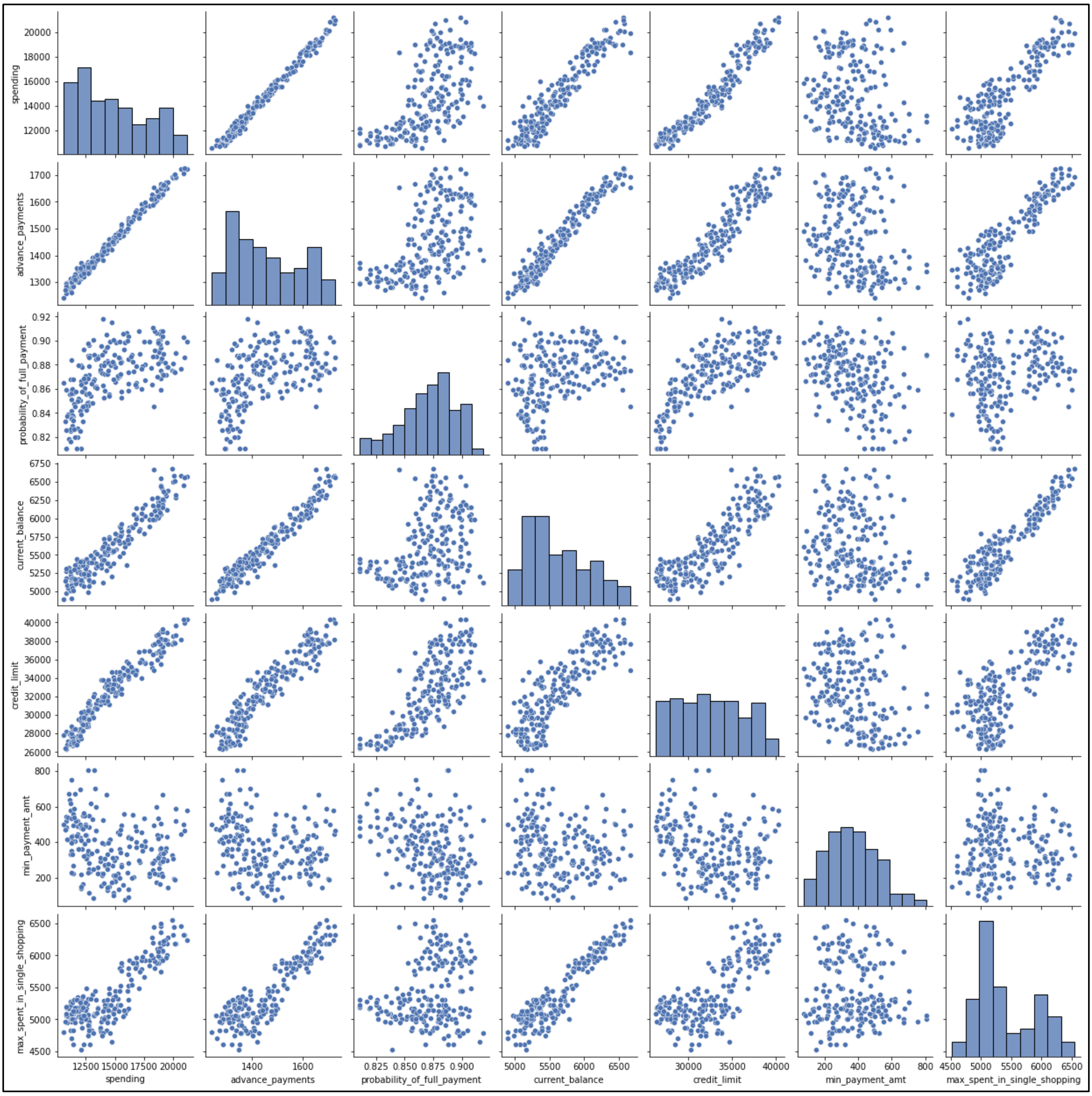


Figure 1. 3: Pairplot

- Customers with higher spendings tend to make higher advance_payments. Looking at these factors, it explains the higher credit_limit they have been provided with.
- We can observe that as credit_limit increasing, current_balance (remaining balance in the credit card) is also increasing.
- Customers with the high probability_of_full_payment have been provided with higher credit_limit, because there could be a lesser chance for them defaulting any payment.
- Customers with higher current_balance tend to make higher max_spent_in_single_shopping.
- Higher the credit_limit enables higher spending capacity of the customer.

Correlation plot (Heatmap):

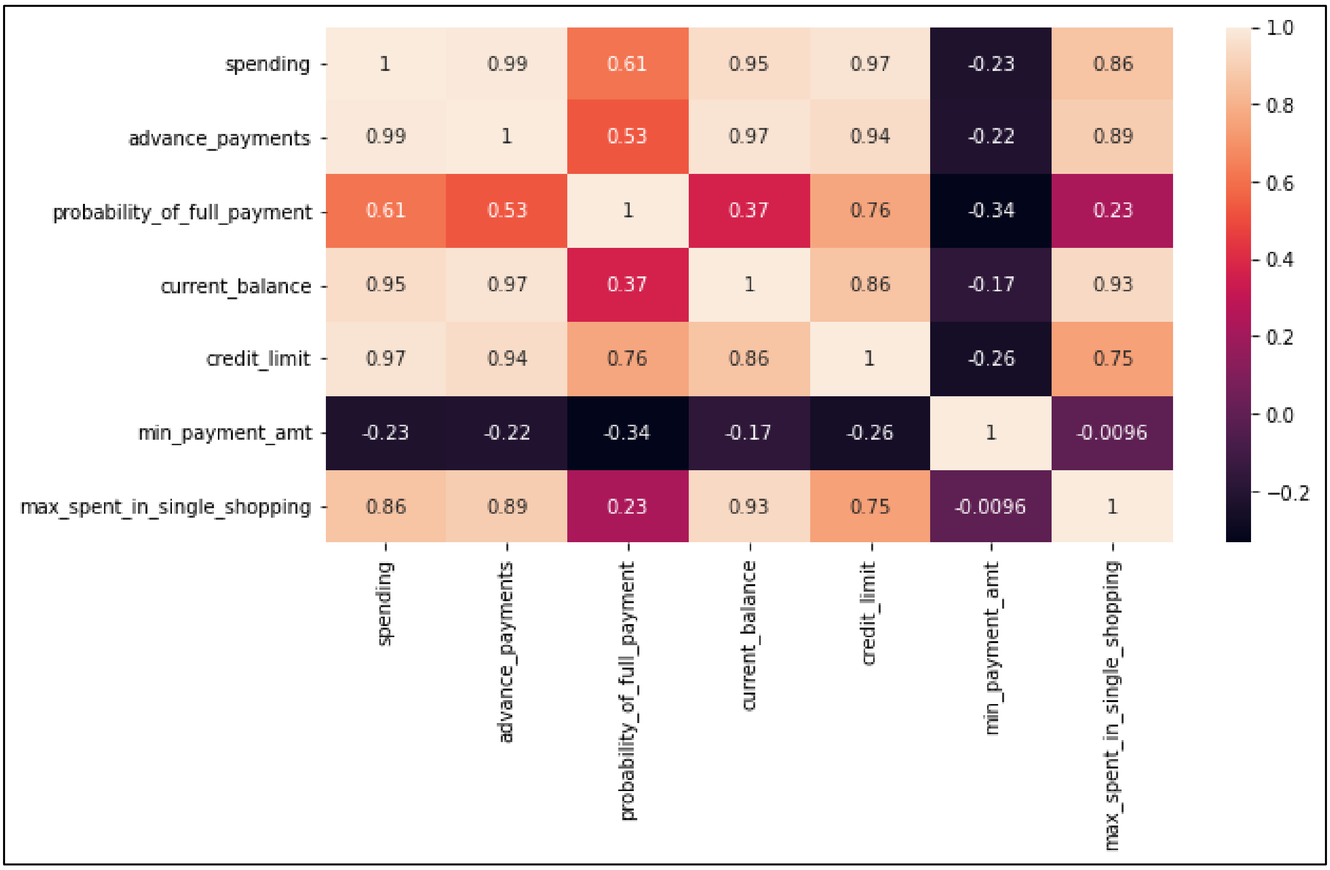


Figure 1. 4: Correlation Plot

- Spending is highly positively correlated with advance_payments, current_balance, credit_limit and max_spent_in_single_shopping. We can say that higher credit_limit increases customers' spending capacity using credit card, hence the higher max_spent_in_single_shopping. Higher credit limit explains the higher current balance remained in the credit card.
- advance_payments is also highly correlated with current_balance and credit_limit.
- probability_of_full_payment is moderately correlated with credit_limit. This explains that customers with higher probability of making full payment have been granted high credit limit, assuming that they won't default.
- min_payment_amt is negatively correlated with all the columns, but the correlation is not significant enough to derive any inferences.

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

Scaling of the data is necessary when the variables of the dataset are of different scales, i.e. one variable is in thousands and other in only hundreds.

	std	max
spending	2909.699431	21180.0000
advance_payments	130.595873	1725.0000
probability_of_full_payment	0.023560	0.9183
current_balance	443.063478	6675.0000
credit_limit	3777.144449	40330.0000
min_payment_amt	149.468900	807.9625
max_spent_in_single_shopping	491.480499	6550.0000

Table 1. 10: Standard Deviation & Maximum Values

In the problem statement we have at hand, there are certain variables which have values of different scales, like spending and credit_limit which have values in the multiples of 10 thousands; advance_payments, current_balance and max_spent_in_single_shopping have values in the multiples of thousands; and probability_of_full_payment have values less than 1. Since the data in these variables are of different scales and the standard deviation of each variable also vary, it is tough to compare them. Hence, the scaling of the variables is necessary for clustering in this case.

Feature scaling (also known as data normalization) is the method used to standardize the range of features of data. Since, the range of values of data may vary widely, it becomes a necessary step in data pre-processing while using machine learning algorithms.

In this method, we convert variables with different scales of measurements into a single scale. StandardScaler normalizes the data using the z-score formula "(x-mean)/standard deviation"; the mean of the data tends to 0 and standard deviation tends to 1.

After performing scaling for the 7 numerical variables, below is the sample of our dataset:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754355	1.811968	0.177628	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

Table 1. 11: Scaled Data Sample

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

For hierarchical clustering, the number of optimum clusters are obtained after the model is run, then we analyse the dendrogram to decide on how many clusters we need.

To perform hierarchical clustering, we are selecting dendrogram and linkage functions.

- Dendrogram function is used for the visualization.
- Linkage function is used to compute the distances and merging the clusters.
    - The linkage method we are choosing is 'Ward's Linkage', which joins records/clusters together progressively to produce larger and larger clusters. It uses the within cluster variance and increase in within cluster variance as a factor to identify the merges in the agglomerative procedure.

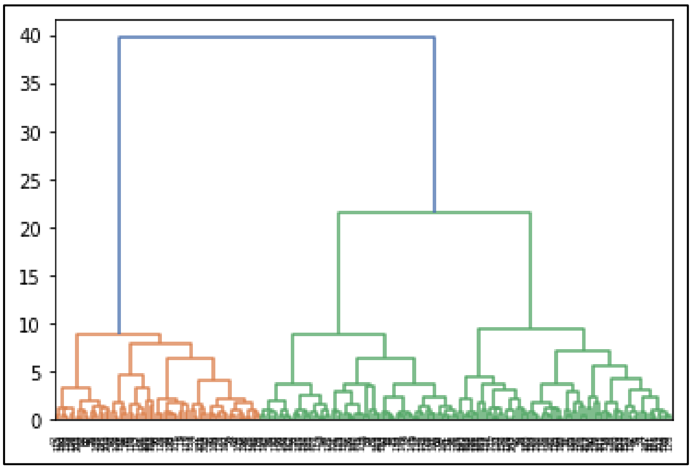


Figure 1. 5: Dendrogram 1

A dendrogram of our scaled data is prepared. Although, the size of the dendrogram is very compact, but we can see that 2 clusters (orange and green) have been created.

We truncated the dendrogram by passing additional parameters to get a neater visual, from which we can decide on the optimum number of clusters:

- truncate_mode='lastp'
- p = 15
    - Since the truncate_mode is 'lastp', the dendrogram will only show last 15 merges.

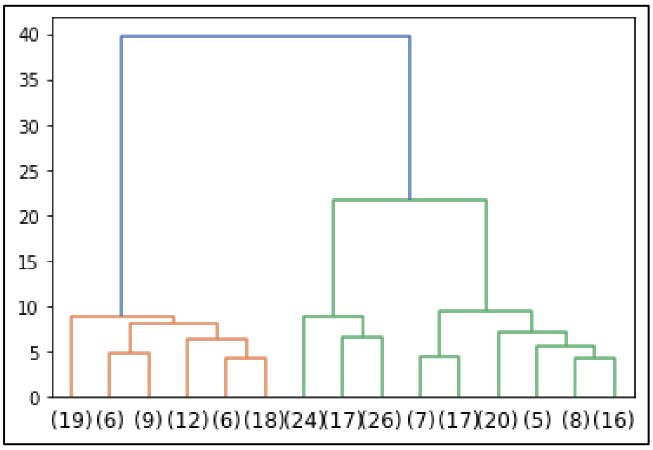


Figure 1. 6: Dendrogram 2

By visualizing the last 15 merges, we observe that we can form 3 clusters to explain the behaviour of the variables. Under clurster 1 we have 70; cluster 2 has 67 and cluster 3 has 72 observations. Which comes to a total of 210 observations, which we have in our data.

Cluster 2 has the minimum and cluster 3 has the maximum number of observations under them.

After establishing linkages and visualizing them using dendrogram, next we are going to obtain the observations that belongs under these 3 clusters for our final verification, using fcluster function.

We have used 'maxclust' criterion to form the clusters and added the clusters to our scaled data. Here is how the new sample looks like:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Clusters
0	1.754355	1.811968	0.177628	2.367533	1.338579	-0.298625	2.328998	1
1	0.393582	0.253840	1.505071	-0.600744	0.858236	-0.242292	-0.538582	3
2	1.413300	1.428192	0.505234	1.401485	1.317348	-0.220832	1.509107	1
3	-1.384034	-1.227533	-2.571391	-0.793049	-1.639017	0.995699	-0.454961	2
4	1.082581	0.998364	1.198738	0.591544	1.155464	-1.092656	0.874813	1

Table 1. 12: Data Sample with Cluster Values

Hierarchical Cluster visualizations:

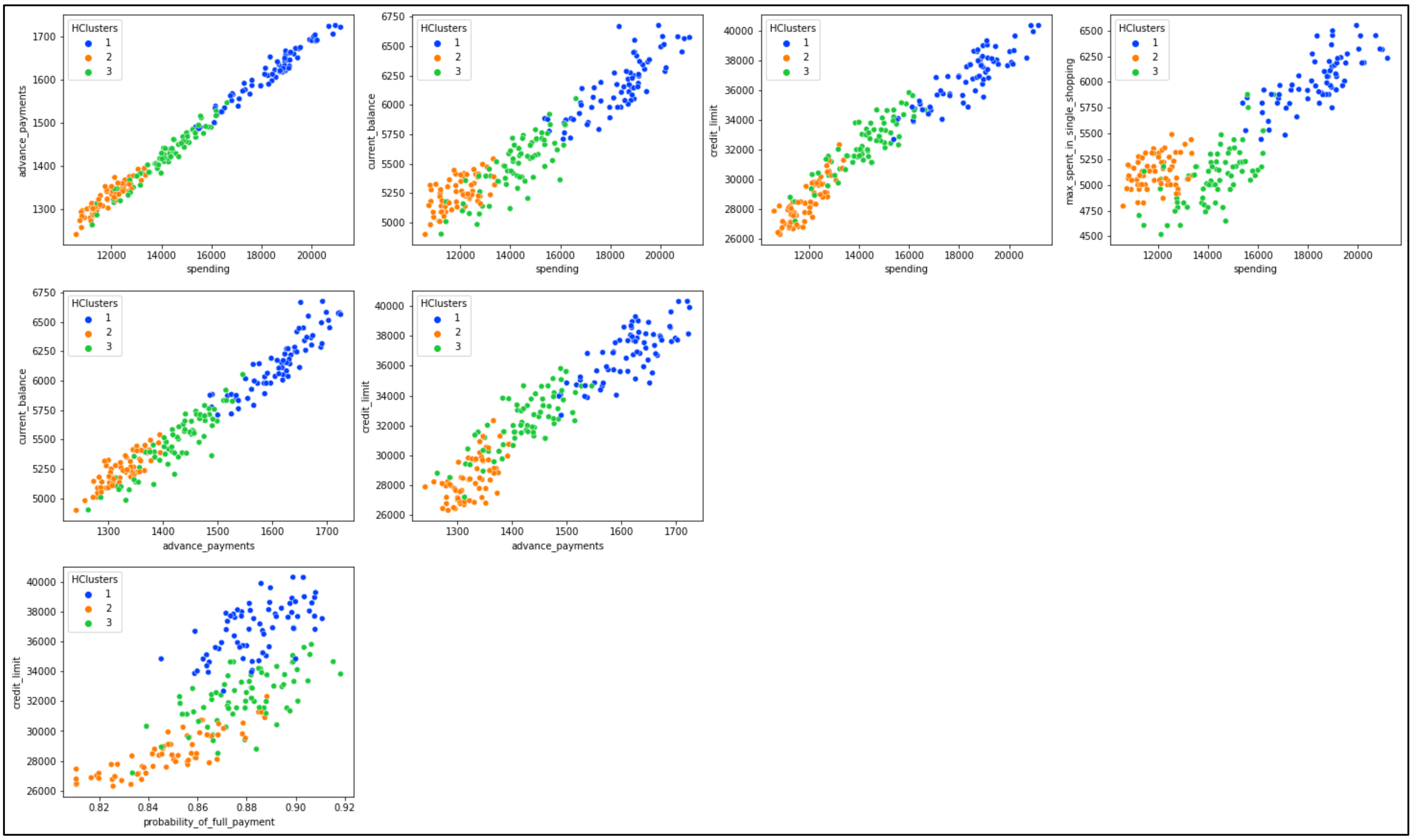


Figure 1. 7: Hierarchical Clustering visualization

- As we can see that the clustering is fairly distinguished. Hence, for certain business problems, individual clusters can be analysed.
- In most of the graphs, cluster 2 (orange) is at the lower end, and cluster 1 (blue) captures the higher end of the values. Cluster 3 (green) captures the values in between clusters 1 & 2.

In depth validation of obtained clusters are done further in the report.

1.4 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.

For K-Means clustering, we need to know the optimum number of clusters we require, before the model is run. In order to decide the optimum number of clusters that we require, a WSS (within sums of square) plot is created:

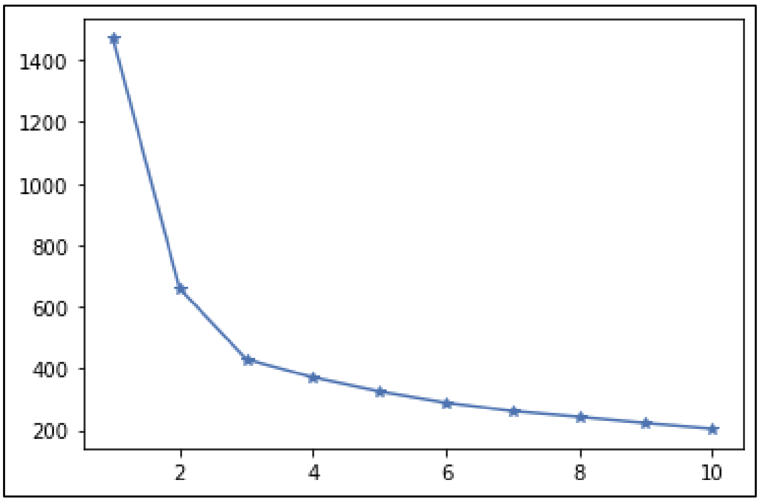


Figure 1. 8: WSS Plot

As we can observe that between k=1, k=2 and k=3, there is a significant drop in within sums of square. Beyond 3 there is a gradual drop. Hence, we can derive that 3 is the optimum number of clusters.

The optimum number of clusters can also be verified using the Silhouette Score. Silhouette Score shows if the sample is enough far away from the neighbouring clusters. The Silhouette Score value:

- close to +1 indicates clusters are well separated
- close 0 indicates clusters are not separated well enough
- close to -1 indicates clustering is not done properly

In our case, the Silhouette Score is 0.4, we can say that the set of clusters are well distinguished/separated.

To check if all the customer records are mapped correctly, we calculated Silhouette Samples for each customer record. The minimum value of Silhouette Sample is 0.002, which means that rest all the values are positive. We can say that there are no customer records mapped incorrectly to any cluster.

The 3 clusters originally obtained using K-means clustering, ranges from 0 to 2. After assigning cluster values to the database, the cluster range as been converted to 1 to 3, to make it easy to compare both clustering methods.

K-Means Cluster Visualization:

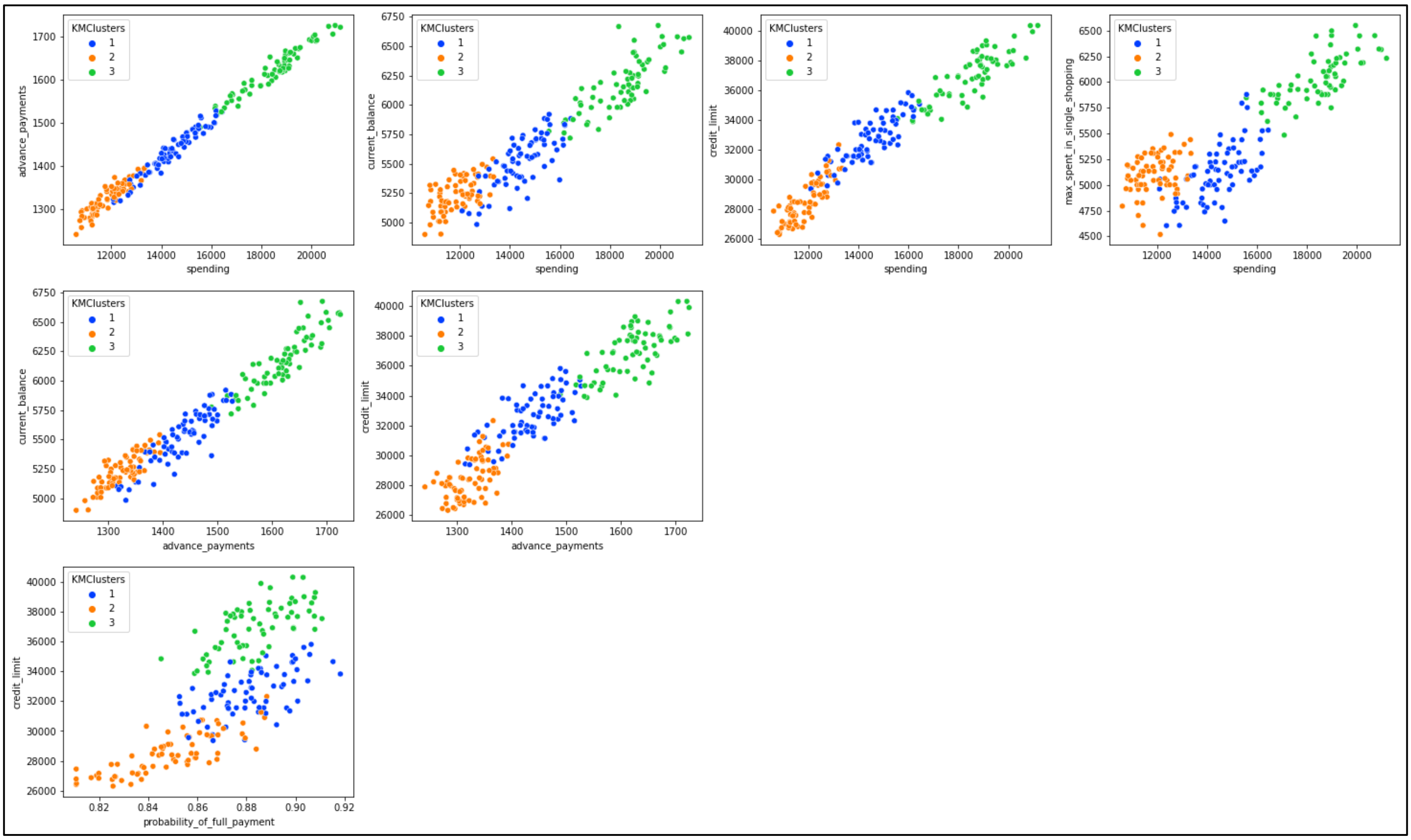


Figure 1. 9: K-Means Clustering Visualization

- Using K-means clustering method also the clusters obtained are fairly distinguished, which also can be very helpful in gathering various inferences for business problems, using individual clusters.
- In most of the graphs, cluster 2 (orange) is at the lower end, and cluster 3 (green) captures the higher end of the values. Cluster 1 (blue) captures the values in between clusters 1 & 2.

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

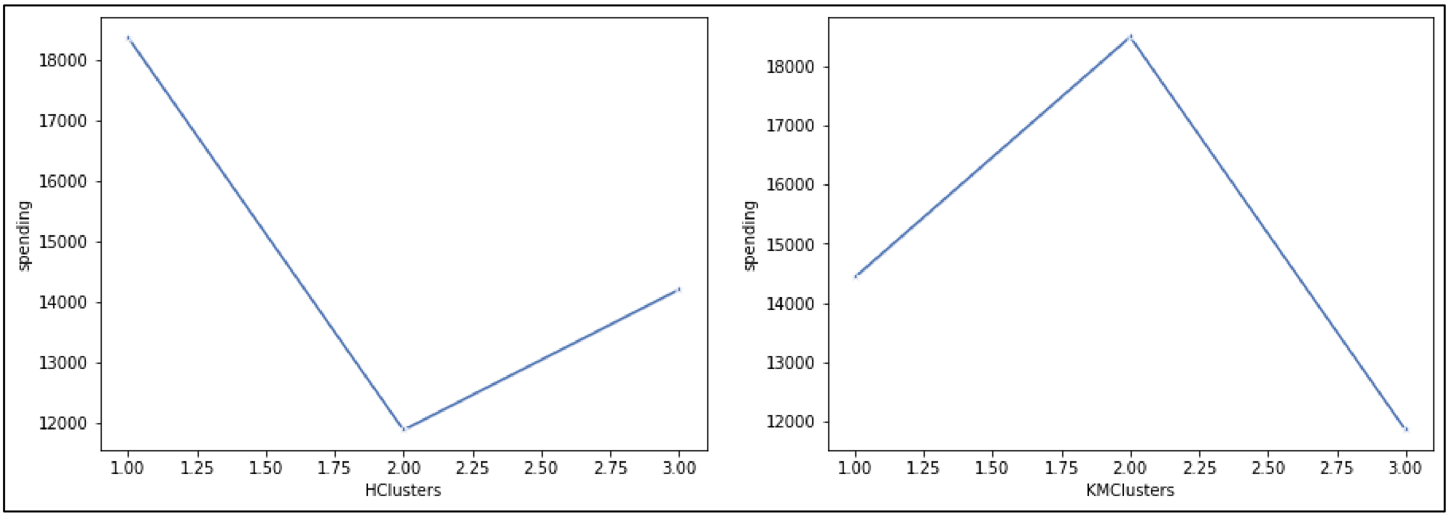


Figure 1. 10: Clusters Profiling

- As we can see that in hierarchical clustering, customers under cluster 1 are the higher spenders, cluster 3 mediocre and cluster 2 lowest.
- In K-means clustering, customers under cluster 2 are the higher spenders, cluster 1 mediocre and cluster 3 lowest.

```
HIERARCHICAL CLUSTERING
Average spending from HCluster 1 =  18371.428571428572
Average spending from HCluster 2 =  11872.388059701492
Average spending from HCluster 3 =  14199.04109589041
```

```
K-MEANS CLUSTERING
Average spending from kmeans Cluster 1 =  14437.887323943662
Average spending from kmeans Cluster 2 =  18495.373134328358
Average spending from kmeans Cluster 3 =  11856.944444444445
```

- As we look at the averages, the values from both the clusters are very similar. Hence, the clustering using both methods, under each obtained clusters have almost identical customer records. Just the numbering of clusters doesn't match, but that doesn't put any impact on interpreting the results.

- As such, we moved forward with profiling hierarchical clustering (HClusters).

- Average of each variable under HClusters:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
1	18371.43	1614.54	0.88	6158.17	36846.29	363.92	6017.37
2	11872.39	1325.7	0.85	5238.94	28485.37	494.03	5122.21
3	14199.04	1423.36	0.88	5478.23	32264.52	261.22	5086.18

Table 1. 13: Variable Means per Cluster

- Cluster 1 captures most of the higher end value. Their credit limit is high, as such the spending is also higher, but also, they end up with the higher balance in their credit card. The higher credit limit is provided to the customers with higher income, so that they are able to pay back without any default. And the track record of paying amount in full has been fairly good (88%). These customers can be identified as economically stable and have high spending capacity.

- For customers from cluster 1, the bank can provide them with enhanced benefits focusing on international travel booking, dining, boarding, shopping and spending. That will promote them to avail these services to increase spending.

- Cluster 2 captures customer segment which seems to be using the credit card very less, as the credit limit provided to them is lower, but they end up with a significant balance in their credit card. That means they are either not using credit card issued by this bank that often or not using the credit card at all.
    - For this customer segment, bank can focus on making them aware of their existing benefits by assigning personal relationship managers. Also, as per their requirements and spending habits, they can be provided with promotional offers focusing on exclusive cashbacks, discounts, redeemable reward points.
    - Also make them aware if they start using this credit card more often, they will be exposed to more exciting offer and additional benefits that bank's elite customers enjoy.
    - Loan and EMI options may also attract them to spend on items they have been holding back on, given the low credit limit.

- Cluster 3 customer segment is medium spending group. Their average probability of making full payment is same as the cluster 1 customers but the credit limit is less. On an average their minimum payment amount is the lowest, indicating they are making full payments more often.
    - Bank can start by increasing their credit limit along with additional benefits, to promote them to make higher usage of the credit to avail those benefits.

Problem 2 – CART-RF-ANN

Introduction

An Insurance firm providing tour insurance is facing higher claim frequency. The management decides to collect data from the past few years. We are assigned the task to make a model which predicts the claim status and provide recommendations to management. The purpose of this case study is to use CART, RF & ANN and compare the models' performances in train and test sets.

Data Dictionary for Models' Performances

1. Age: Age of insured
2. Agency_Code: Code of tour firm
3. Type: Type of tour insurance firms
4. Claimed: Claim Status (target variable)
5. Commision: The commission received for tour insurance firm (Commission is in percentage of sales)
6. Channel: Distribution channel of tour insurance agencies
7. Duration: Duration of the tour (in days)
8. Name of the tour insurance products (Product)
9. Sales: Amount worth of sales per customer in procuring tour insurance policies in rupees (in 100's)
10. Destination: Destination of the tour

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

2.1.1 Sample of dataset

Here are the top 5 rows (sample) of the dataset:

	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 2. 1: Data Sample

- Dataset has 10 variables.
- As mentioned in the Data Dictionary, 'Sales' values are in 100s. For further analysis of the data, 'Sales' values have been converted to its true forms. This is how the data appears now:

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	C2B	Airlines	No	0.70	Online	7	251.0	Customised Plan	ASIA
1	36	EPX	Travel Agency	No	0.00	Online	34	2000.0	Customised Plan	ASIA
2	39	CWT	Travel Agency	No	5.94	Online	3	990.0	Customised Plan	Americas
3	36	EPX	Travel Agency	No	0.00	Online	4	2600.0	Cancellation Plan	ASIA
4	33	JZI	Airlines	No	6.30	Online	53	1800.0	Bronze Plan	ASIA

Table 2. 2: Transformed Data Sample

2.1.2 Check for Duplicate Records

```
Number of duplicate records: 139
```

As we can see there are 139 duplicate records. In the data, there is no unique identifier which can be helpful in validating if these 139 duplicate records contain some kind of erroneous observations or just 2 different customers happened to have same characteristics and preference. Having said that and given the fact that travel company can sell the same kind of tour package to similar demography, we are not considering there are any duplicate entries in the data.

2.1.3 Types of variables in the dataset

```
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)
```

- There are a total of 3000 observations (rows) under 10 features (columns) in the dataset.
- There are 2 variables of float64, 2 of int64 and 6 of object datatype.

2.1.4 Missing values in the dataset

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

There are no missing values present in the dataset.

2.1.5 Descriptive Statistics

Describe function provides a table indicating the count of variables, mean, standard deviation and other values for the 5-point summary that includes (min, 25%, 50%, 75% and max) for numeric variables. 50% in the table is also known as median.

	count	mean	std	min	25%	50%	75%	max
Age	3000.0	38.091000	10.463518	8.0	32.0	36.00	42.000	84.00
Commision	3000.0	14.529203	25.481455	0.0	0.0	4.63	17.235	210.21
Duration	3000.0	70.001333	134.053313	-1.0	11.0	26.50	63.000	4580.00
Sales	3000.0	6024.991333	7073.395353	0.0	2000.0	3300.00	6900.000	53900.00

Table 2. 3: Data Description for Continuous Columns

For object/categorical columns, describe function shows the total count, unique values in each column, most frequent value and value frequency in each column.

	count	unique	top	freq
Agency_Code	3000	4	EPX	1365
Type	3000	2	Travel Agency	1837
Claimed	3000	2	No	2076
Channel	3000	2	Online	2954
Product Name	3000	5	Customised Plan	1136
Destination	3000	3	ASIA	2465

Table 2. 4: Data Description for Categorical Columns

- Age of customers ranges from 8 till 84 who are insured, with the average age of 39.
- Commision and Sales variables have 0 as minimum values.
- Duration contains -1 and 0 as values, which seems to be an anomaly as the days can't be denoted as -1 and 0. Also, the maximum value in this field is 4580, which is far apart from the second highest value 466 and seems to be a data entry error. This variable needs to be cleaned by replacing -1 and 0 with nearest valid value '1' and 4580 with nearest maximum value '466'.

```
Sample of 'Duration' values:

1508   -1
1746    0
2628    0
424     1
1430    1
```

```
Maximum values in 'Duration':

873      428
1398     431
2260     434
2914     466
2845    4580
```

- Data is focused on 4 agencies with codes 'C2B', 'EPX', 'CWT' and 'JZI'; with 'EPX' having maximum number of records (1365).
- There are 2 'Types' of agencies, 'Airlines' and 'Travel Agency'; where 'Travel Agency' has maximum number of records (1837).

- The target/dependent variable 'Claimed' has 2 categorical values 'No' (69.2%) and 'Yes' (30.8%). The data seems to be well balanced.

```
Proportion of categories in the target variable (in %):

No      69.2
Yes     30.8
```

- Customers have been provided with 2 types of 'Channel' – Online and Offline; where online channel is majorly used (2954).
- There are 5 types of product packages provided - Customised Plan, Cancellation Plan, Bronze Plan, Silver Plan and Gold Plan. 'Customized Plan' seems to be the most popular among customers.
- Among 'ASIA', 'Americas' and 'Europe', customers travelled to Asian countries the most.

NOTE: Anomaly identified in Duration column, has been treated before checking for outliers.

```
count    3000.000000
mean       68.631333
std       106.010500
min         1.000000
25%        11.000000
50%        26.500000
75%        63.000000
max       466.000000
Name: Duration, dtype: float64
```

2.1.6 Check for outliers

Boxplots have been plotted for numerical variables to check for outliers:

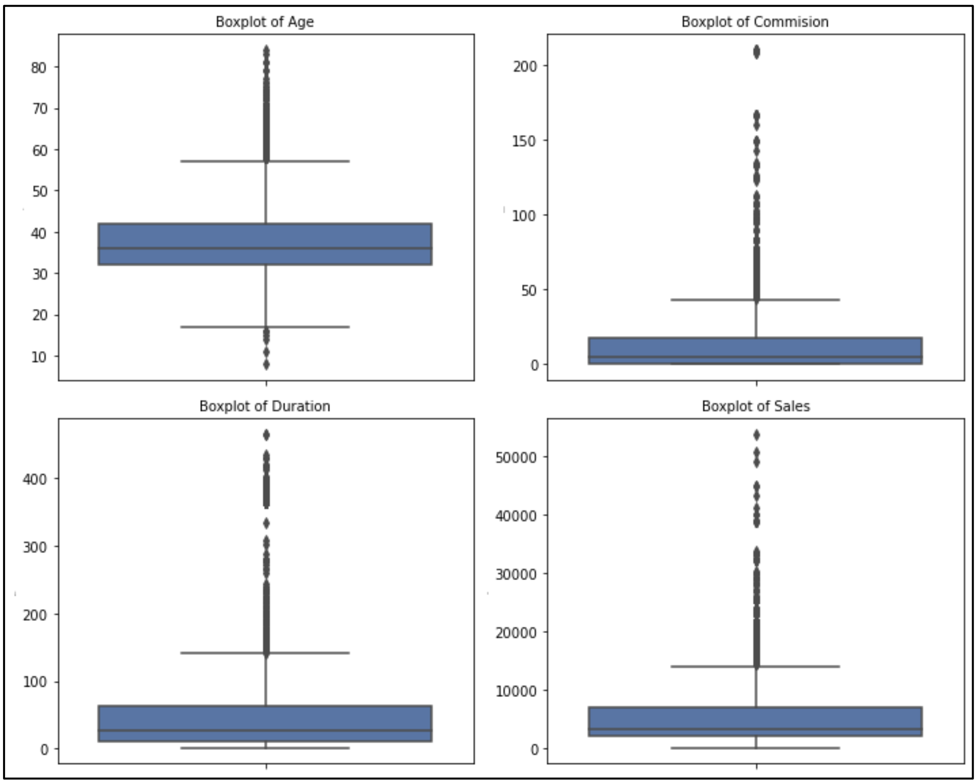


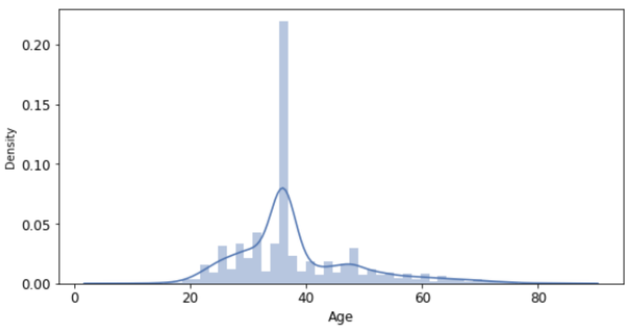
Figure 2. 1 Boxplot for Outliers

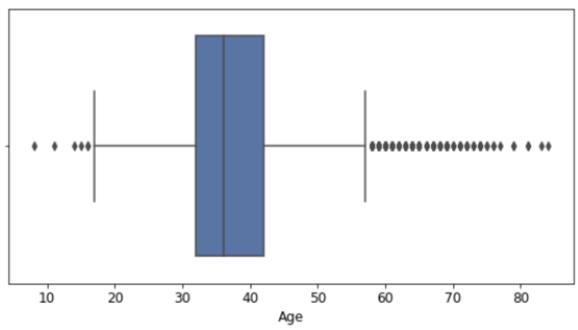
There many outliers present in the dataset. However, an observation is considered to be an outlier if that particular has been mistakenly captured in the data set. Treating outliers sometimes results in the models having better performance but the models lose out on the generalization. Hence, the models are built without treating outliers.

2.1.7 Univariate analysis

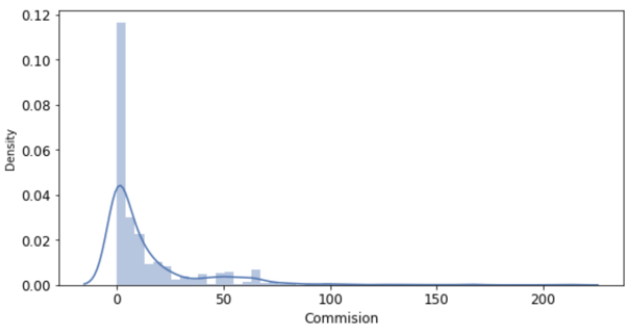
Univariate analysis is performed for all the numeric variables individually to display their statistical description. Visualized the variables using distplot to view the distribution and the box plot to view 5-point summary and outliers if any.

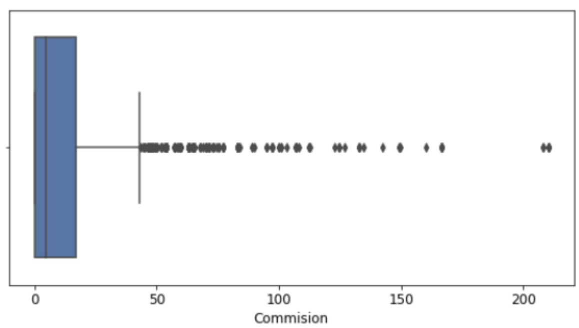
```
Description of Age
........................................................................
count    3000.000000
mean       38.091000
std        10.463518
min         8.000000
25%        32.000000
50%        36.000000
75%        42.000000
max        84.000000
Name: Age, dtype: float64

 Distribution of Age                                        Countplot of Age
.............................................................................................................................
```





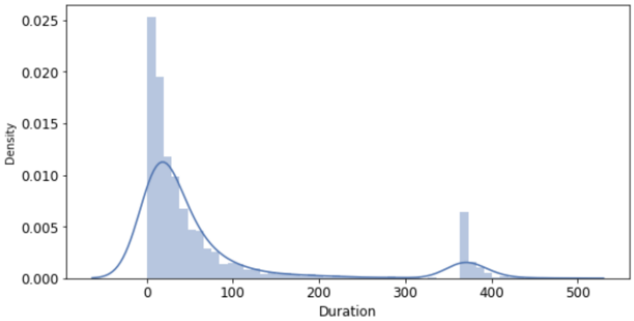

```
Description of Commision
........................................................................
count    3000.000000
mean       14.529203
std        25.481455
min         0.000000
25%         0.000000
50%         4.630000
75%        17.235000
max       210.210000
Name: Commision, dtype: float64

 Distribution of Commision                                        Countplot of Commision
.............................................................................................................................
```

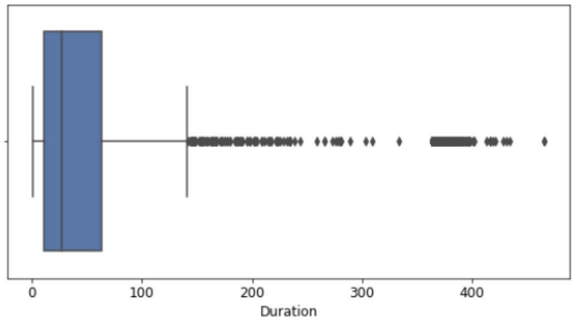




```
Description of Duration
......................................................................
count    3000.000000
mean       68.631333
std       106.010500
min         1.000000
25%        11.000000
50%        26.500000
75%        63.000000
max       466.000000
Name: Duration, dtype: float64
```

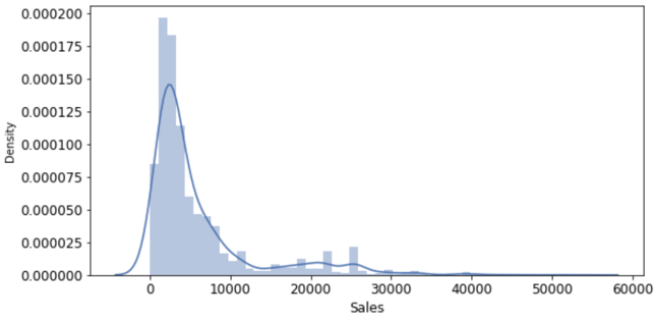
Distribution of Duration — Countplot of Duration






```
Description of Sales
......................................................................
count    3000.000000
mean     6024.991333
std      7073.395353
min         0.000000
25%      2000.000000
50%      3300.000000
75%      6900.000000
max     53900.000000
Name: Sales, dtype: float64
```

Distribution of Sales — Countplot of Sales



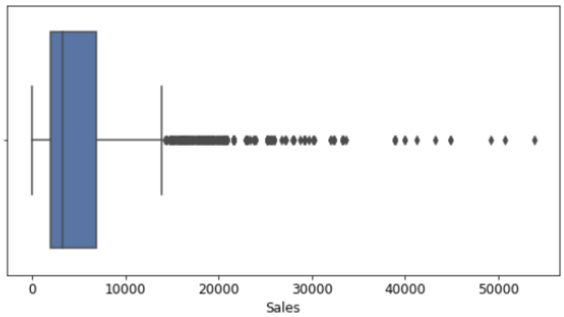


Figure 2. 2: Univariate Analysis

	Kurtosis	Skewness
Age	1.652124	1.149713
Commision	13.984825	3.148858
Duration	3.690495	2.237271
Sales	6.155248	2.381148

Table 2. 5: Kurtosis & Skewness

- There are 4 numeric fields in the dataset.
- From the boxplots we can see that there are outliers present in the data set, but there is no need to treat them since they are not going to affect the prediction models.

- Distribution for all the variables is positively skewed, with 'Commision' having the highest kurtosis/peak.
- For 'Age', 'Commision', 'Duration' distribution is bi-modal and for 'Sales' distribution is multi-modal.
- We observe that 25% (Q1) is comprised of 0 commision. Most of the data in 'Commision' feature lies beyond 75% (Q3) of the distribution.

2.1.8 Bivariate analysis

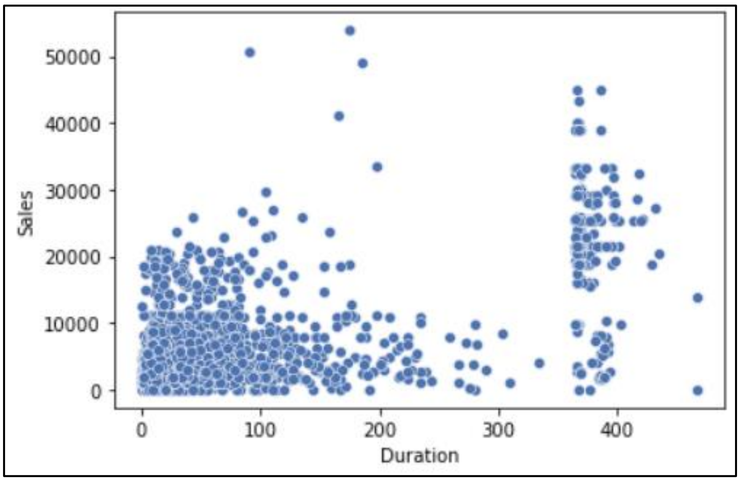


Figure 2. 3: Bivariate Analysis 1

- Customers travelling for longer duration would practically opt for hight amount of insurance policy, but this doesn't appear to be the case for our sample. Majority of customers travelled for approximately 180 days and have opted for insurance policies valued not more than INR 20,000.

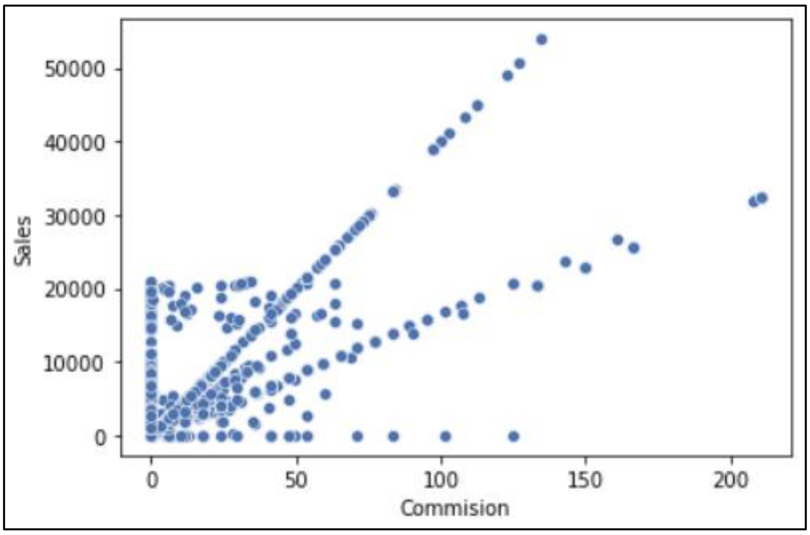


Figure 2. 4: Bivariate Analysis 2

- Commision is increasing with the increase in Sales, which is a good indicator.

Let's have a look at the numeric variables against the target variable 'Claimed':

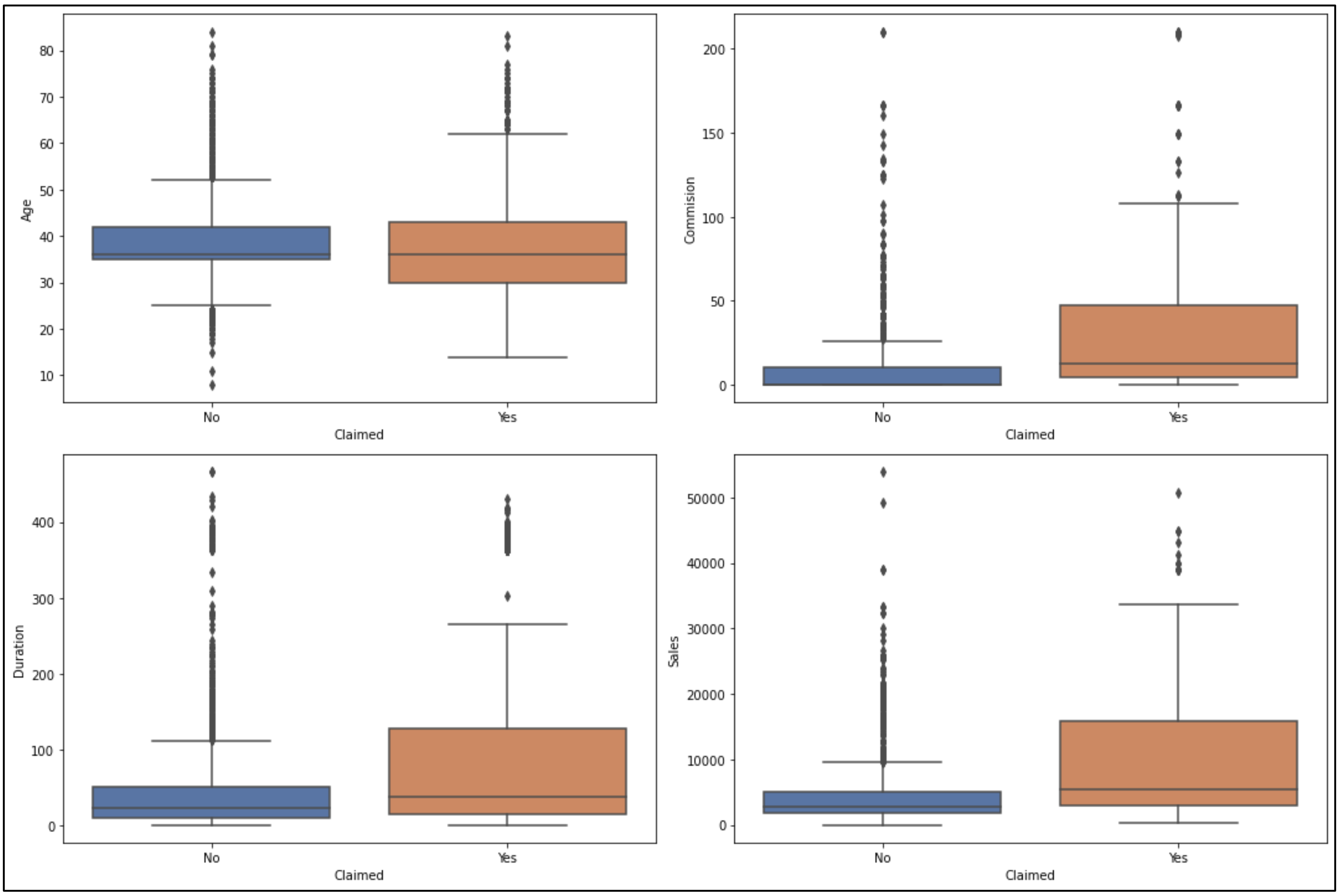


Figure 2. 5: Bivariate Analysis 3

- The median age of the customers who have made the claim and who have not made the claim is almost the same. So, based on age we cannot differentiate which category, young or old, is causing higher claim frequency.
- The median values of the commision, duration and sales are higher for the customers who have made the claim. Hence, we can say that the customers who brings in higher sales and commission and travels for long duration tend to claim their insurance policy.

Let's have a look at the patterns of categorical variables against the target variable 'Claimed':

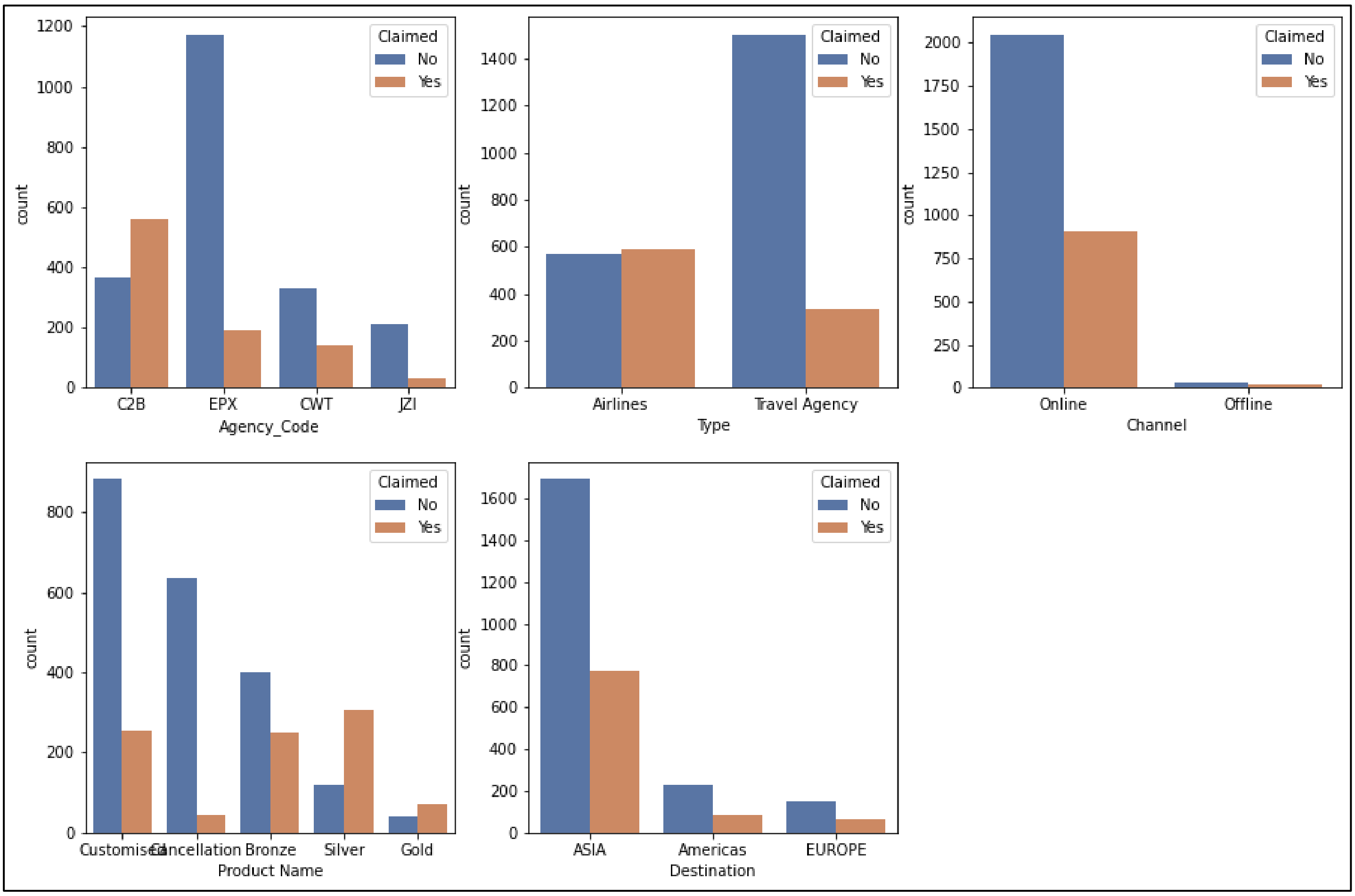


Figure 2. 6: Bivariate Analysis 4

- We can observe that C2B insurance agency faces the highest number of claims, among all the other agencies.
- Airlines type of insurance firms have almost equal amount of customers who claim and don't claim. Although, Travel Agency firms have more customers and their claim frequency is comparatively very low.
- As majority of the customers opt for online channels for insurance policies, that explains the high number of claims as compared to that of offline channels. However, in online channels the claim ratio is low.
- More number of customers among who opted for Silver and Gold plans claimed for insurance.
- We can put our main focus on the C2B agency which faces the highest number of claims and also belongs to Airlines industry.

2.1.9 Multivariate analysis

Pair plot (numeric variables):

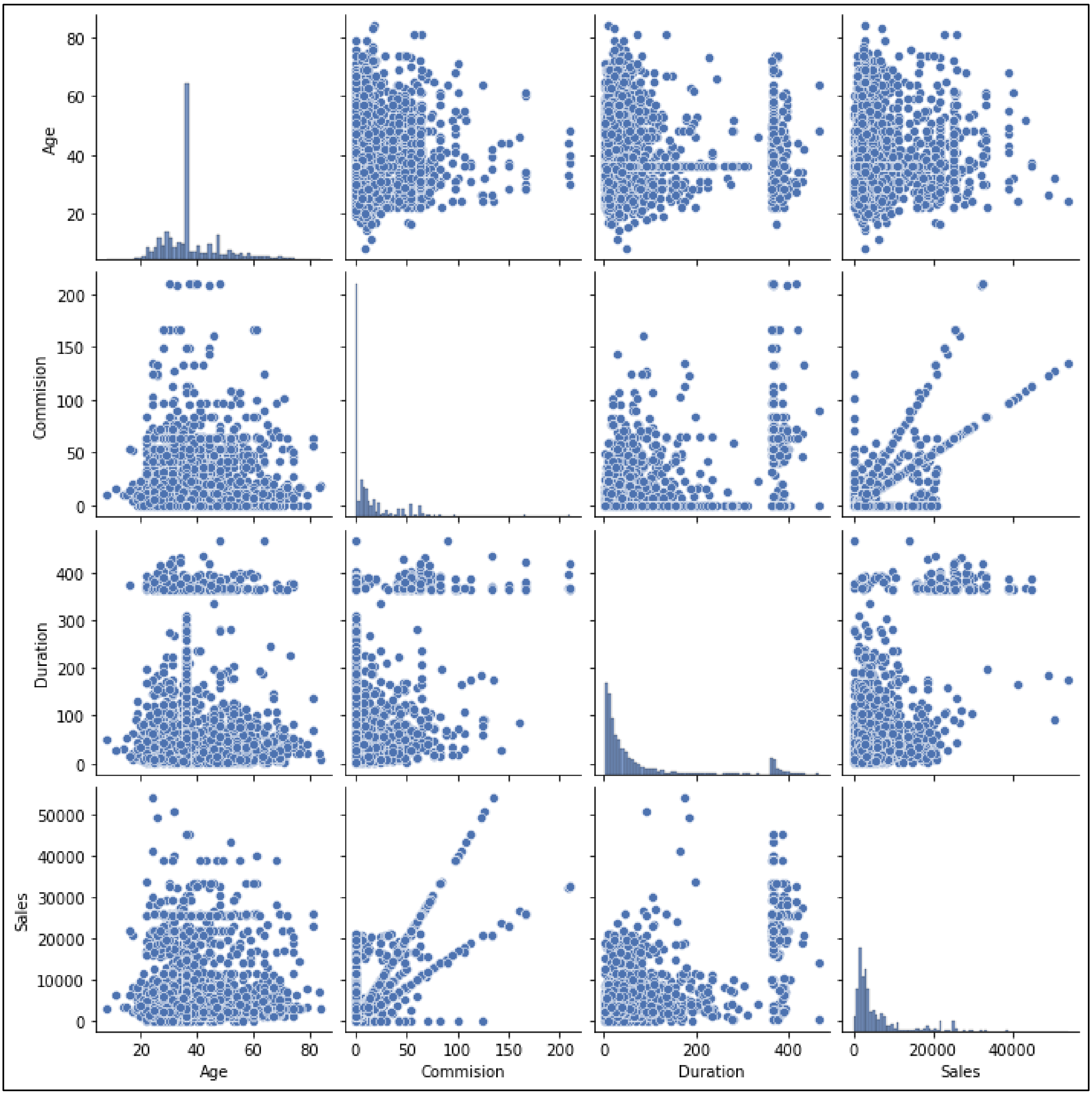


Figure 2. 7: Pairplot

- We can only find an interpretable relationship between Sales and Commision. Commision is increasing with the increase in Sales.
- Rest of the variables don't seem to have definite patterns between them to make inferences on.

Correlation plot (Heatmap) of numeric variables:

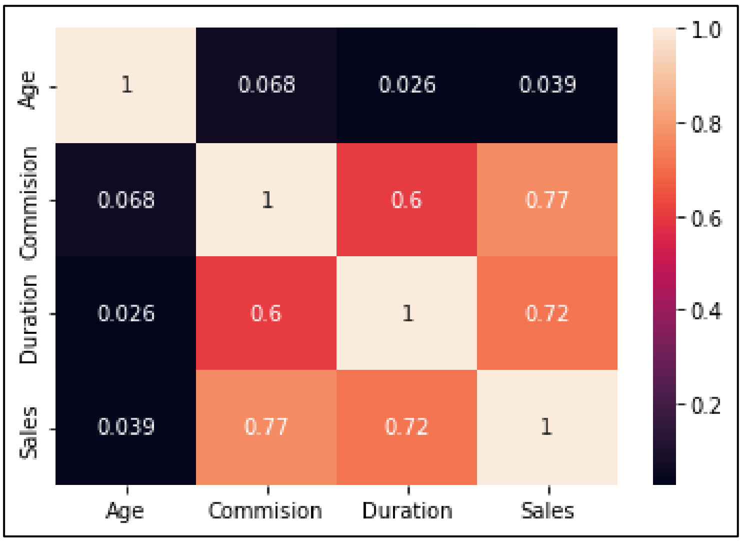


Figure 2. 8: Correlation Plot

- There is a moderately good correlation among Duration, Sales and Commision. We can infer that as the travel duration increases the sales amount of insurance policies also increases, hence the higher % of commission per sale.

2.1.10 Data Encoding

For prediction models the data to pass should be in numeric/categorical format only. The object variables in our dataset need to be converted to integer format, for this we are using one-hot encoding.

After encoding, this is how the variables appear in the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 16 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Age                        3000 non-null   int64
 1   Commision                  3000 non-null   float64
 2   Duration                   3000 non-null   int64
 3   Sales                      3000 non-null   float64
 4   Agency_Code_CWT            3000 non-null   uint8
 5   Agency_Code_EPX            3000 non-null   uint8
 6   Agency_Code_JZI            3000 non-null   uint8
 7   Type_Travel Agency         3000 non-null   uint8
 8   Claimed_Yes                3000 non-null   uint8
 9   Channel_Online             3000 non-null   uint8
 10  Product Name_Cancellation  3000 non-null   uint8
 11  Product Name_Customised    3000 non-null   uint8
 12  Product Name_Gold          3000 non-null   uint8
 13  Product Name_Silver        3000 non-null   uint8
 14  Destination_Americas       3000 non-null   uint8
 15  Destination_EUROPE         3000 non-null   uint8
dtypes: float64(2), int64(2), uint8(12)
```

- All of 'object' variables got separated into different variables with datatype uint8 (integer).
- Now we have 16 variables in our encoded dataset.

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

The target variable in our encoded dataset is 'Claimed_Yes', where 0 = No and 1 = Yes. Here is the proportion of values in the target variable:

```
Percentage of "No" in target variable: 69.2 %
Percentage of "Yes" in target variable: 30.8 %
```

The proportion seems to be good enough to move forward with models building.

The data has been first divided in to independent and dependent (target) variables, x and y respectively.

The data is now split into training and testing set with both sets having 70% and 30% of the data, respectively. Here is the proportion of target variable in both the sets:

```
Percentage of "No" in target variable in Training set: 69.1 %
Percentage of "Yes" in target variable in Training set: 30.9 %


Percentage of "No" in target variable in Testing set: 69.44 %
Percentage of "Yes" in target variable in Testing set: 30.56 %
```

2.2.1 Classification Model – CART / Decision Tree:

In the first instance, we will allow the decision tree to be completely built using default parameters; criterion = 'gini' and random_state = 2. After observing performance of the model, we will decide the pruning parameters to better fit the model.

Below is the decision tree built using default parameters. We can see that it is overgrown, unreadable and needs to be pruned.

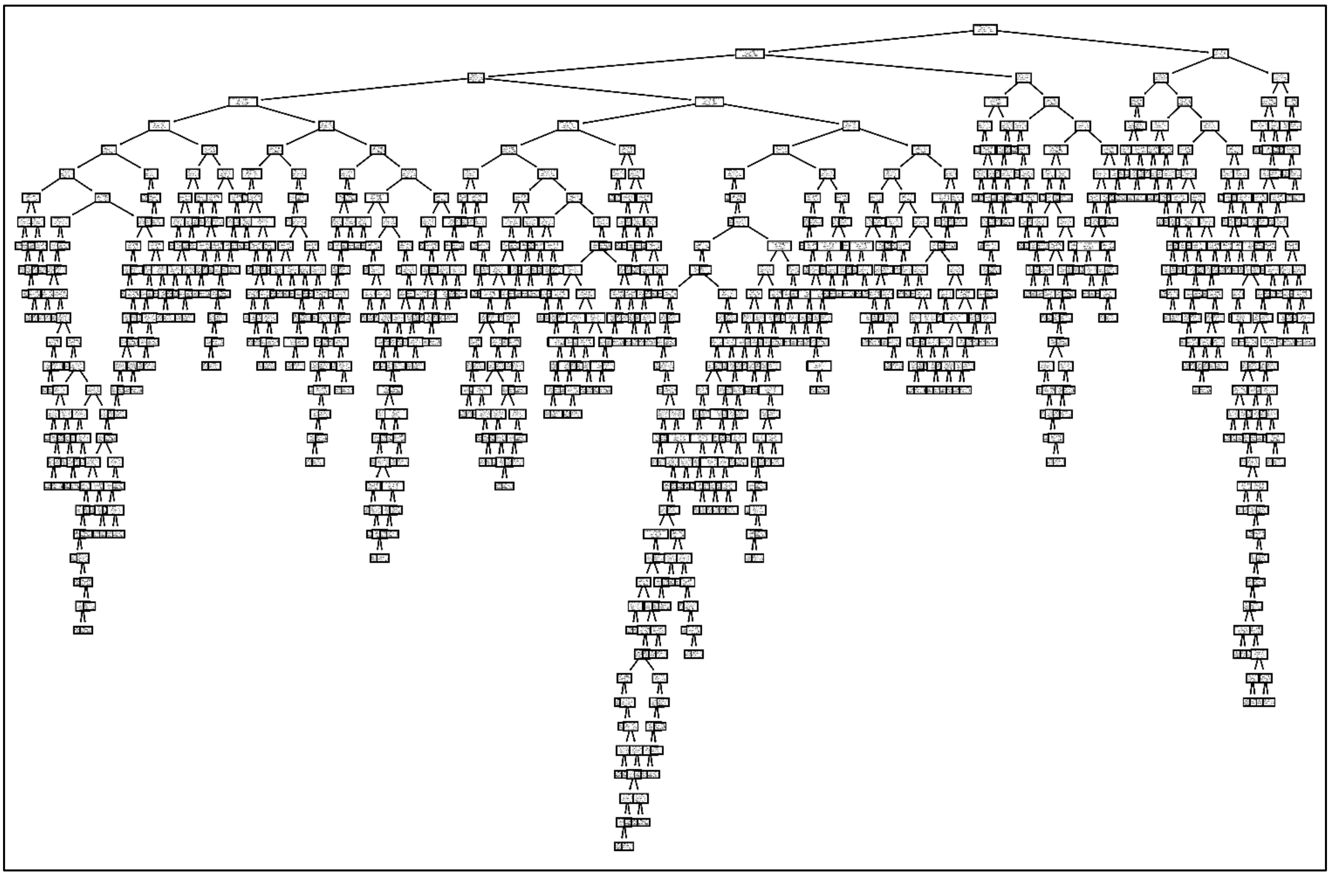


Figure 2. 9: Decision Tree 1

<u>Confusion matrix and classification report:</u>

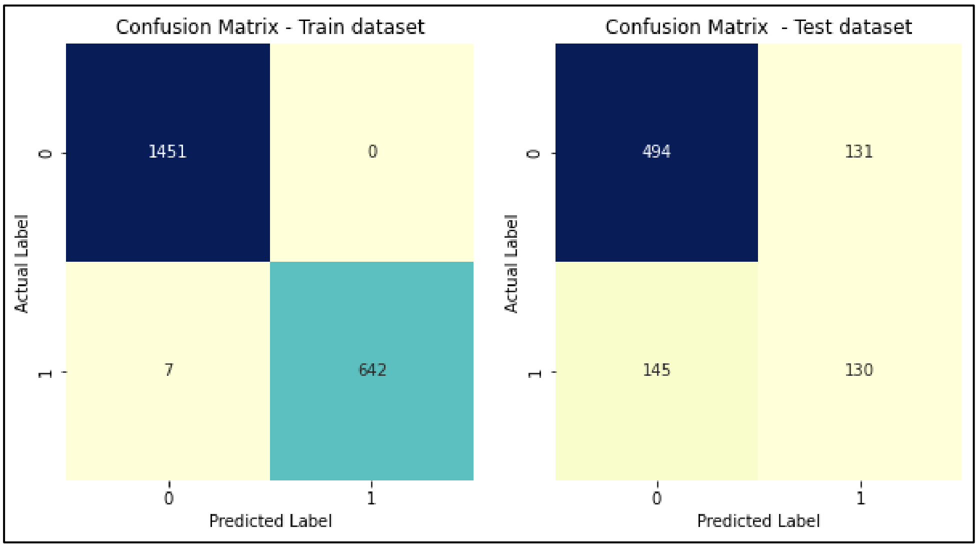


Figure 2. 10: CART Confusion Matrix 1

Classification Report - Train dataset

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1451
1	1.00	0.99	0.99	649
accuracy			1.00	2100
macro avg	1.00	0.99	1.00	2100
weighted avg	1.00	1.00	1.00	2100

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.77	0.79	0.78	625
1	0.50	0.47	0.49	275
accuracy			0.69	900
macro avg	0.64	0.63	0.63	900
weighted avg	0.69	0.69	0.69	900

Figure 2. 11: CART Classification Report 1

As we can see that the decision tree model with default parameters is clearly overfit, accuracy of train set is 1 and for test set it is ~70.

We performed GridSearch crossvalidation for this model, by passing multiple combination of values for the parameters, to find out the best parameters to build a model that performs well.

- max_depth - The maximum depth of the tree.
- min_samples_split - The minimum number of samples required to split an internal node.
- min_samples_leaf - The minimum number of samples required to be at a leaf node.
- criterion – The function to measure the quality of a split.

After running GridSearch cross validation, here are the observations:

- Best parameters: 'criterion': 'gini', 'max_depth': 7, 'min_samples_leaf': 5, 'min_samples_split': 55
- Feature importance: the below plot shows the relative importance of features used in building the model, starting from the highest importance –

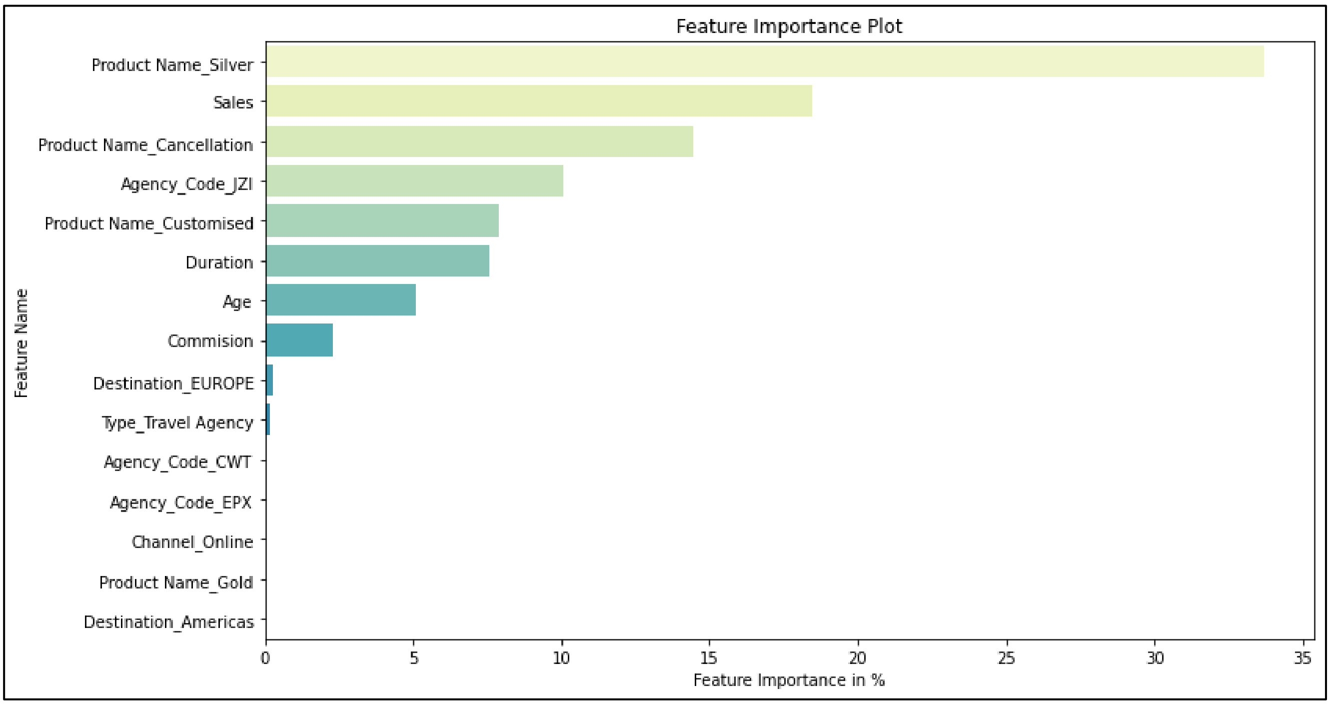


Figure 2. 12: Feature Importance

- Decision tree plotted using plot_tree function:

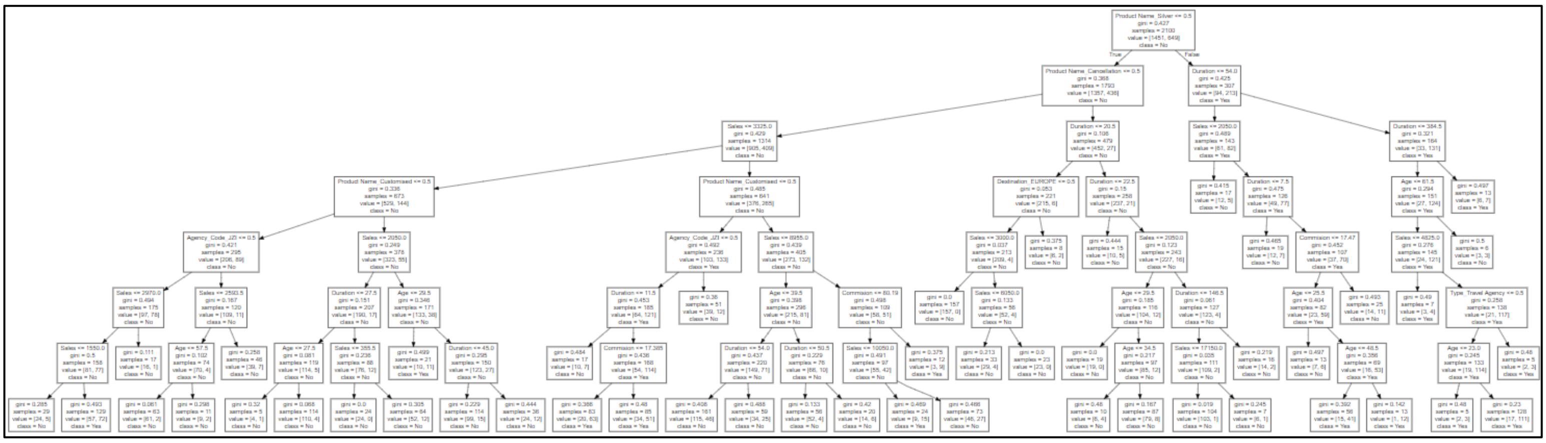


Figure 2. 13: Decision Tree 2

2.2.2 Classification Model – Random Forest:

In the first instance, we will built the model using default parameters; n_estimators = 100, criterion = 'gini', random_state = 2, oob_score = True. After observing performance of the model, we will decide the best parameters to better fit the model.

- Out-of-bag (oob_score) tell the accuracy of the model. In this case, the oob_score is ~0.75, which means there is ~25% error rate in the model.

Confusion matrix and classification report:

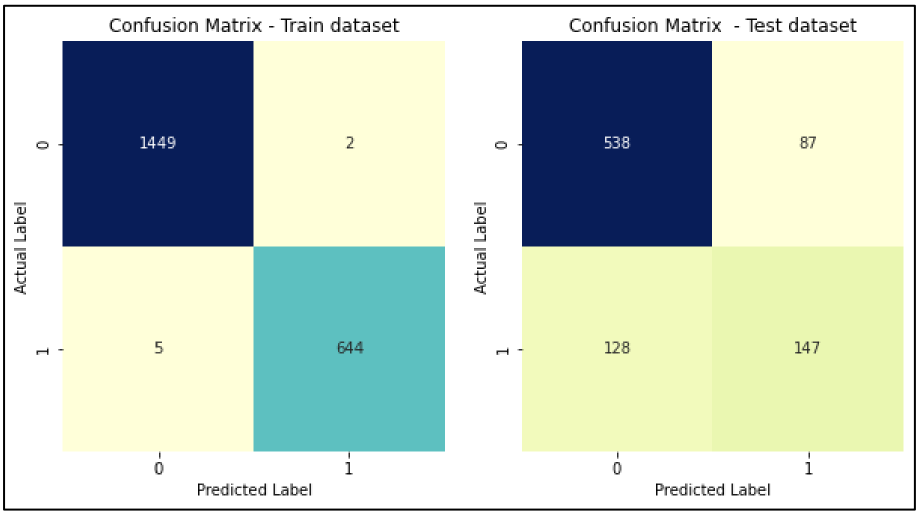


Figure 2. 14: RF Confusion Matrix 1

Classification Report - Train dataset

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1451
1	1.00	0.99	0.99	649
accuracy			1.00	2100
macro avg	1.00	1.00	1.00	2100
weighted avg	1.00	1.00	1.00	2100

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.81	0.86	0.83	625
1	0.63	0.53	0.58	275
accuracy			0.76	900
macro avg	0.72	0.70	0.71	900
weighted avg	0.75	0.76	0.76	900

Figure 2. 15: RF Classification Report 1

As we can see that the random forest model with default parameters is clearly overfitted, accuracy of train set is 1 and for test set it is ~75.

We performed GridSearch crossvalidation for this model, by passing multiple combination of values for the parameters, to find out the best parameters to build a model that performs well.

- n_estimators - The number of trees in the forest.
- criterion - The function to measure the quality of a split.
- max_depth - The maximum depth of the tree.
- min_samples_split - The minimum number of samples required to split an internal node.
- min_samples_leaf - The minimum number of samples required to be at a leaf node.
- max_features - The number of features to consider when looking for the best split.

After running GridSearch cross validation, here are the observations:

- Best parameters: 'criterion': 'gini', 'max_depth': 8, 'max_features': 4, 'min_samples_leaf': 4, 'min_samples_split': 40, 'n_estimators': 650.
- Feature importance: the below plot shows the relative importance of features used in building the model, starting from the highest importance –

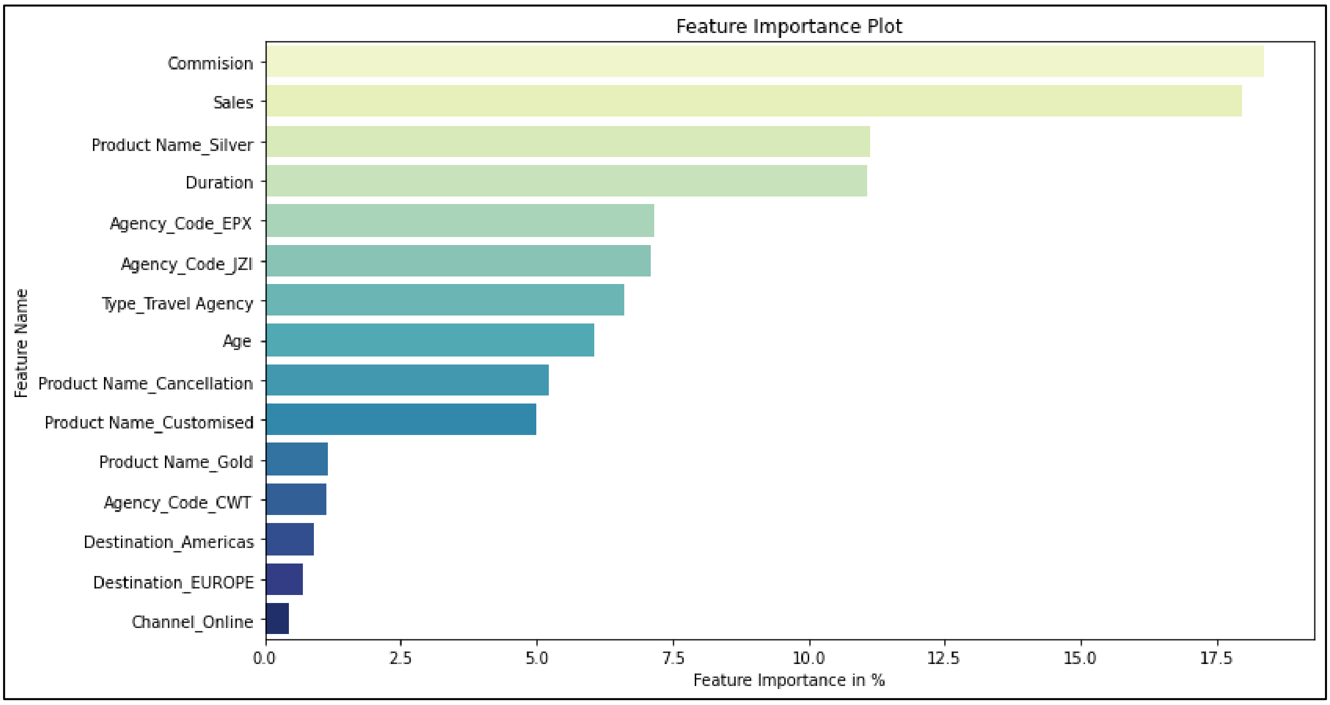


Figure 2. 16: RF Feature Importance

2.2.3 Classification Model – Artificial Neural Network:

In the first instance, we will built the model using default parameters (hidden_layer_sizes=100, activation='relu', random_state = 2). After observing performance of the model, we will decide the best parameters to better fit the model.

It is important that we pass the scaled data through Neural Network model otherwise the model will get biased towards the variables with higher magnitude.

<u>Confusion matrix and classification report:</u>

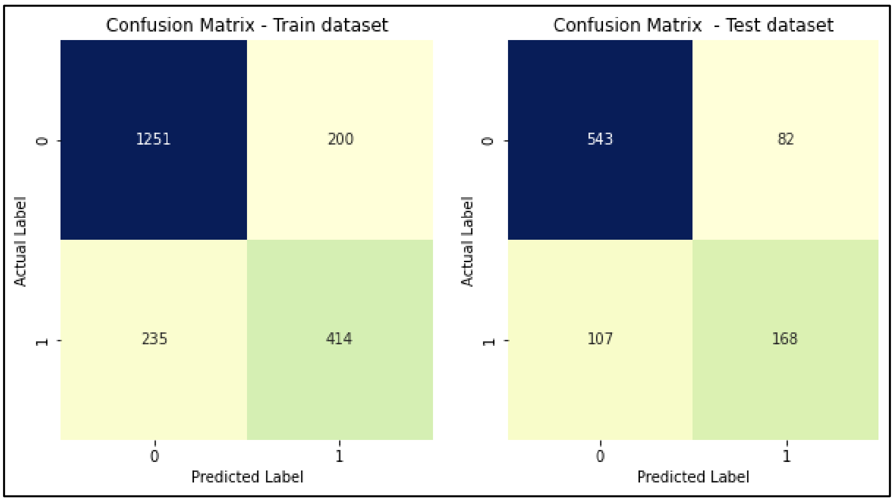


Figure 2. 17: ANN Confusion Matrix 1

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.84	0.86	0.85	1451
1	0.67	0.64	0.66	649
accuracy			0.79	2100
macro avg	0.76	0.75	0.75	2100
weighted avg	0.79	0.79	0.79	2100

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.84	0.87	0.85	625
1	0.67	0.61	0.64	275
accuracy			0.79	900
macro avg	0.75	0.74	0.75	900
weighted avg	0.79	0.79	0.79	900

Figure 2. 18: ANN Classification Report 1

As we can observe that the default parameters have performed considerably well. Let's try with different parameters to see if the results can be improved.

We performed GridSearch crossvalidation for this model, by passing multiple combination of values for the parameters, to find out the best parameters to build a model that performs well.

- hidden_layer_sizes - The ith element represents the number of neurons in the ith hidden layer.
- activation - Activation function for the hidden layer.
- solver - The solver for weight optimization.
- max_iter - Maximum number of iterations. The solver iterates until convergence (determined by 'tol') or this number of iterations. For stochastic solvers ('sgd', 'adam'), note that this determines the number of epochs (how many times each data point will be used), not the number of gradient steps.
- tol - Tolerance for the optimization.

After running GridSearch cross validation, here are the observations:

- Best parameters: 'activation': 'relu', 'hidden_layer_sizes': 100, 'max_iter': 10000, 'random_state': 2, 'solver': 'adam', 'tol': 0.01.

Feature importance cannot be obtained for Artificial Neural Network model.

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

2.3.1 Classification Model – CART / Decision Tree:

- Confusion matrix:

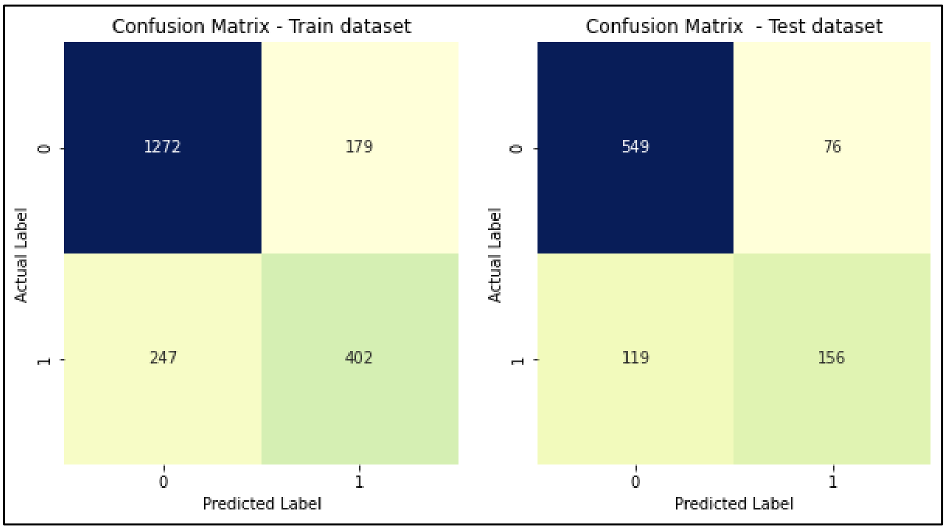


Figure 2. 19: CART Confusion Matrix 2

- Classification report:

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.84	0.88	0.86	1451
1	0.69	0.62	0.65	649
accuracy			0.80	2100
macro avg	0.76	0.75	0.76	2100
weighted avg	0.79	0.80	0.79	2100

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.82	0.88	0.85	625
1	0.67	0.57	0.62	275
accuracy			0.78	900
macro avg	0.75	0.72	0.73	900
weighted avg	0.78	0.78	0.78	900

Figure 2. 20: CART Classification Report 2

- Above results indicate that we have reduced the overfitting of the Decision Tree model, and now the accuracy and F1-score of train and test set is very close.

```
ROC - AUC score for training set is 0.86
ROC - AUC score for testing set is 0.81
```

- The ROC-AUC score for the testing set is less than that of the training set, hence we can say that the testing sample is not performing as well as the training sample.
- Most important predictors of the decision tree model are:
  - Product_Name_Silver Plan
  - Sales
  - Product_Name_Cancellation Plan
  - Agency_Code_JZI
  - Product_Name_Customized Plan
  - Duration
  - Age

  - Commision
  - Destinatioin_Europe
  - Type_Travel_Agency

- ROC Curve:

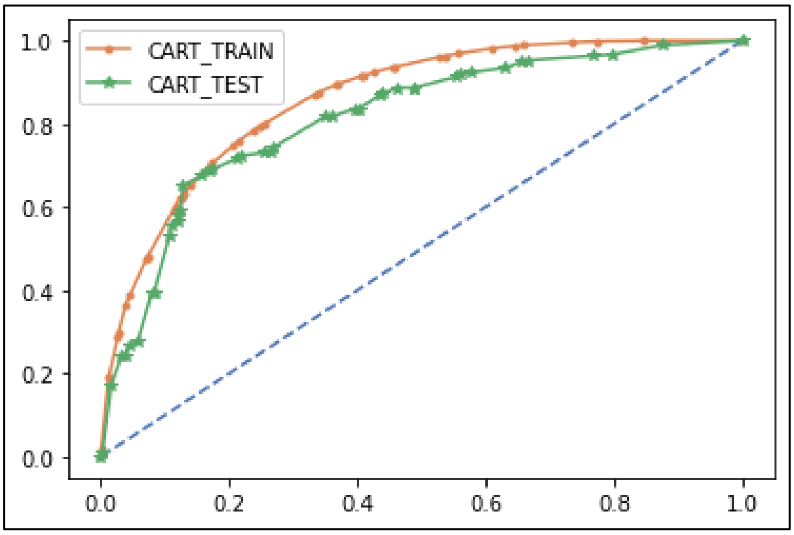


Figure 2. 21: CART ROC Curve

- Looking at the ROC curve, we can interpret that test set is not performing as good as the train set.

2.3.2 Classification Model – Random Forest:

- Confusion matrix:

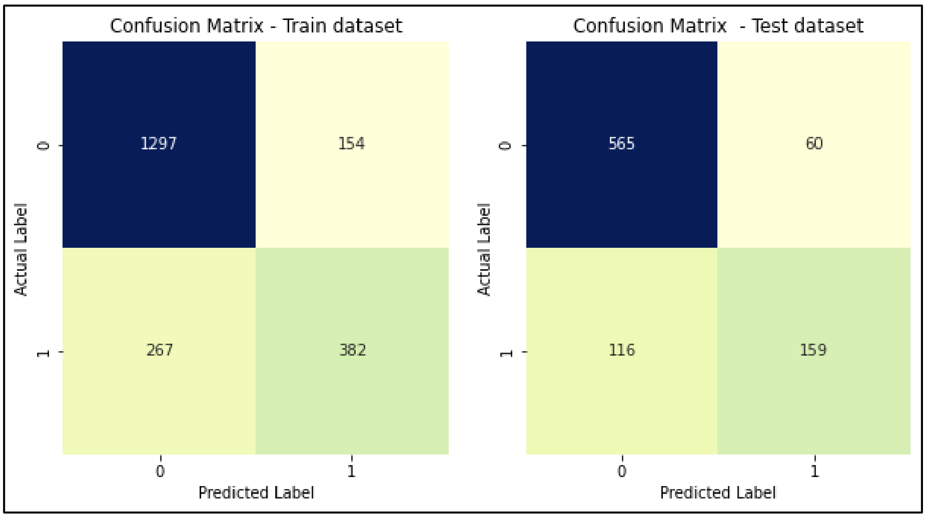


Figure 2. 22: RF Confusion Matrix 2

- Classification report:

```
Classification Report  - Train dataset
              precision    recall  f1-score   support

           0       0.83      0.89      0.86      1451
           1       0.71      0.59      0.64       649

    accuracy                           0.80      2100
   macro avg       0.77      0.74      0.75      2100
weighted avg       0.79      0.80      0.79      2100
```

```
Classification Report  - Test dataset
              precision    recall  f1-score   support

           0       0.83      0.90      0.87       625
           1       0.73      0.58      0.64       275

    accuracy                           0.80       900
   macro avg       0.78      0.74      0.75       900
weighted avg       0.80      0.80      0.80       900
```

Figure 2. 23: RF Classification Report 2

- Above results indicate that we have reduced the overfitting of the Random Forest model, and now the accuracy and F1-score of train and test are same.

```
ROC - AUC score for training set is 0.86
ROC - AUC score for testing set is 0.83
```

- The ROC-AUC score for the testing set is less than that of the training set. Based on this observation, we can say that the testing sample is not performing exactly as well as the training sample.
- Most important predictors of the decision tree model are:
  - Commision
  - Sales
  - Product_Name_Silver Plan
  - Duration
  - Agency_Code_EPX
  - Agency_Code_JZI
  - Type_Travel_Agency
  - Age
  - Product_Name_Cancellation Plan
  - Product_Name_Customized Plan

- Product_Name_Gold Plan
- Agency_Code_CWT
- Destination_Americas
- Destination_Europe
- Channel_Online

- ROC Curve:

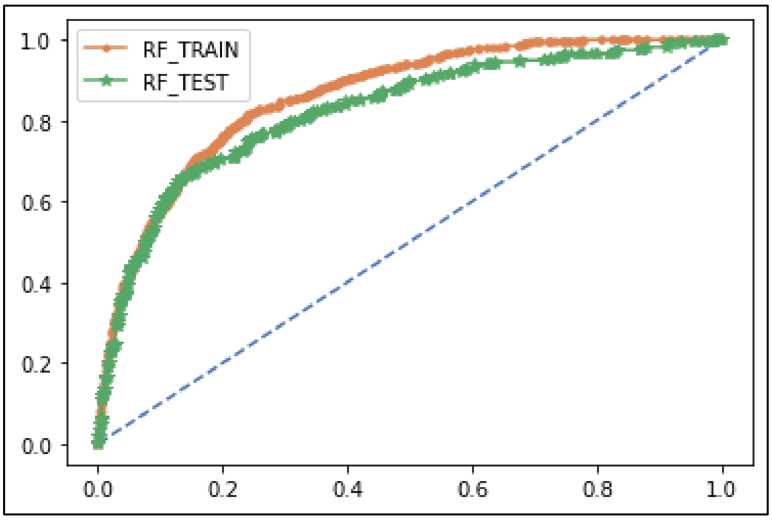


Figure 2. 24: RF ROC Curve

- Looking at all the outputs from the model, we can say that Random Forest model has better precision than CART model, and the Random Forest model turned out to be well trained as compared to the CART model.

2.3.3 Classification Model – Artificial Neural Network:

- Confusion matrix:

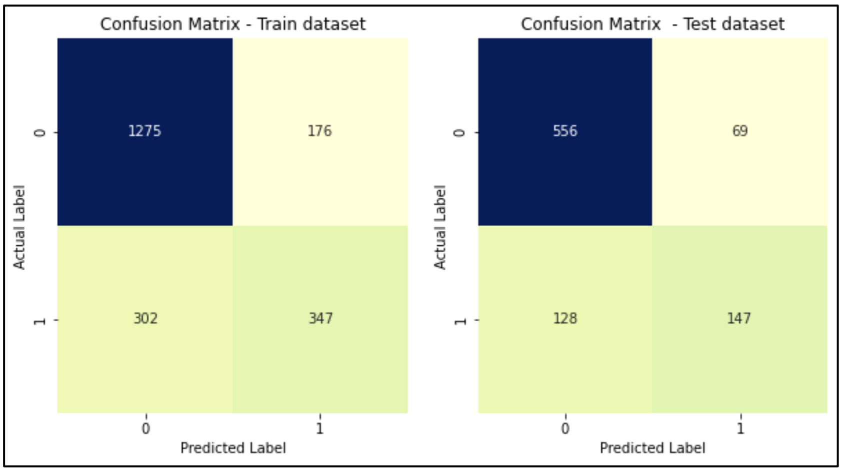


Figure 2. 25: ANN Confusion Matrix 2

- Classification report:

Classification Report - Train dataset

	precision	recall	f1-score	support
0	0.81	0.88	0.84	1451
1	0.66	0.53	0.59	649
accuracy			0.77	2100
macro avg	0.74	0.71	0.72	2100
weighted avg	0.76	0.77	0.76	2100

Classification Report - Test dataset

	precision	recall	f1-score	support
0	0.81	0.89	0.85	625
1	0.68	0.53	0.60	275
accuracy			0.78	900
macro avg	0.75	0.71	0.72	900
weighted avg	0.77	0.78	0.77	900

Figure 2. 26: ANN Classification Report 2

- As we can see that the tuned model performance has not improved as compared to the default parameters, but test set seems to be performing slightly better than the train set.

```
ROC - AUC score for training set is 0.82
ROC - AUC score for testing set is 0.82
```

- The ROC-AUC score for training and testing set is same, hence, we can say that the testing sample is performing as well as the training sample.

- ROC Curve:

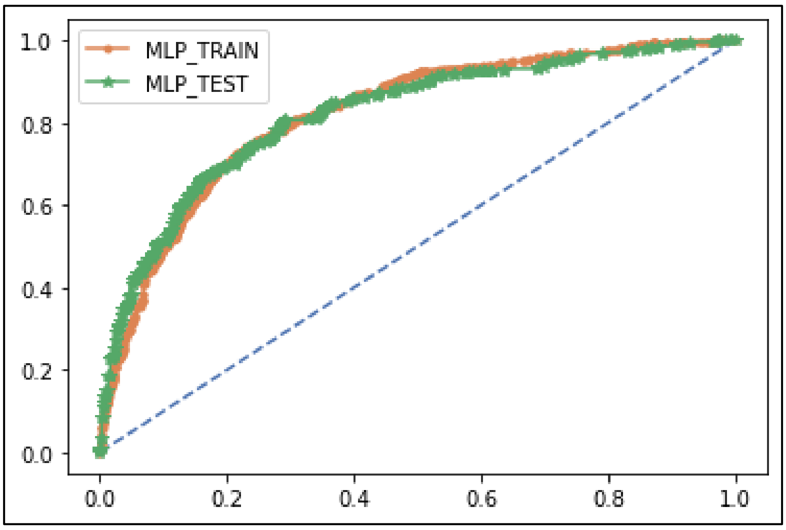


Figure 2. 27: ANN ROC Curve

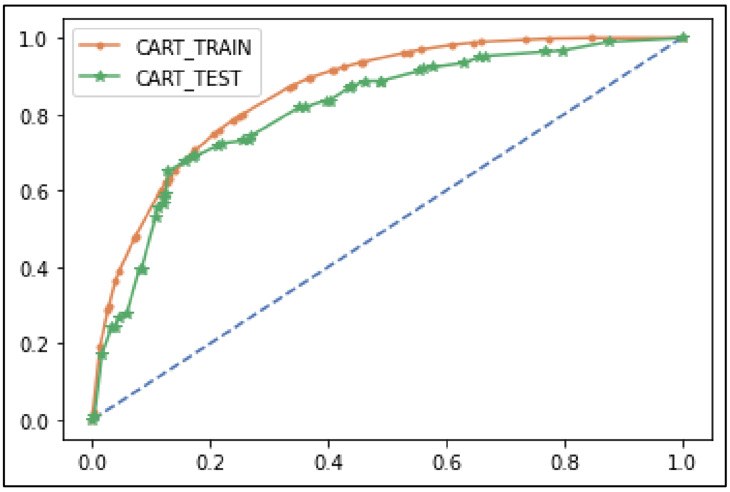
- Looking at all the outputs from the model, we can say that Artificial Neural Network model is also performing really well as compared to the CART model. But Random Forest model is giving slightly better results.

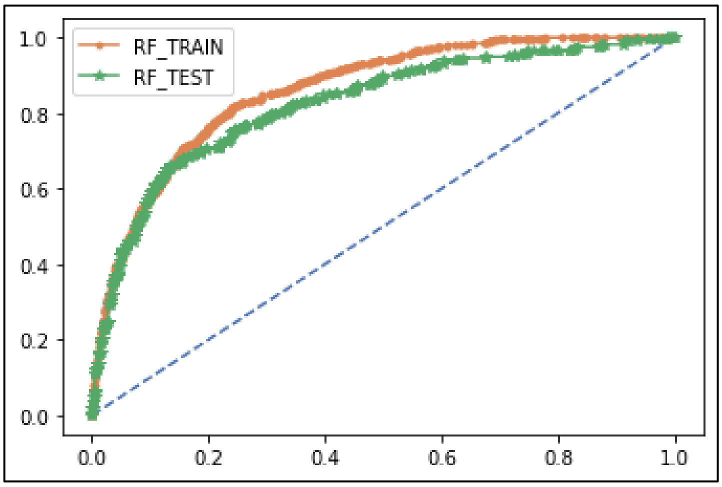
2.4 Final Model: Compare all the models and write an inference which model is best/optimized.

	CART Train	CART Test	RF Train	RF Test	ANN Train	ANN Test
Accuracy	0.80	0.78	0.80	0.80	0.77	0.78
Recall	0.62	0.57	0.59	0.58	0.53	0.53
AUC	0.86	0.81	0.86	0.83	0.82	0.82
Precision	0.69	0.67	0.71	0.73	0.66	0.68
F1 score	0.65	0.62	0.64	0.64	0.59	0.60

Table 2. 6: Models Comparision

<u>ROC Curve Comparision</u>:





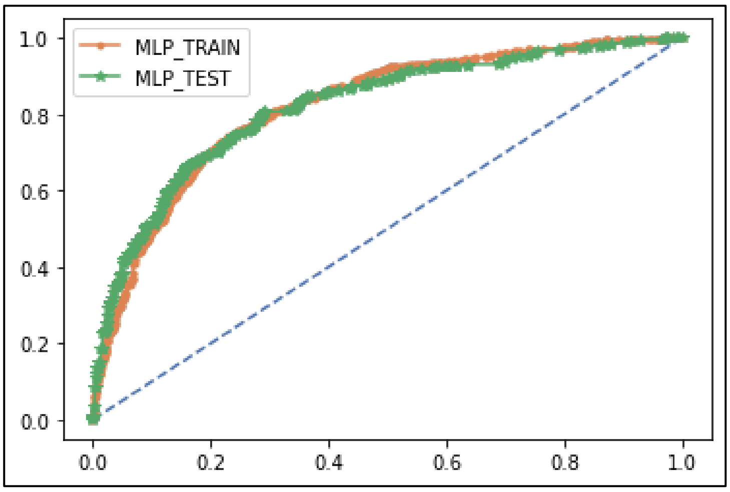


Figure 2. 28: ROC Curve Comparision

- As we can observe from the above data, the difference in the accuracy, ROC-AUC score, precision, recall and F1 score of train and test set in ANN and Random Forest models is very less than the CART model.
- We can clearly drop the idea to go forward with the CART model, as it is not performing well enough and the data is not well trained

- Looking at the above table values, both ANN and Random Forest models have performed well in terms of its Accuracy, recall and precision. The precision of testing set is even slightly higher than the training set for both models. But the values are higher for Random Forest model.
- On the other hand, he ROC curve is best fit for the ANN model, where testing set performing as good as the training set, which is not the case with Random Forest model.
- The scenario of False Negative, where “prediction is that policy was not claimed but actually policy was claimed” need to be of main focus for the business. As such, the Accracy and Recall score is very important for this case study. ANN model has provided a better Accuracy and Recall score with the best trained data.
- After evaluating all above factors, we can conclude that ANN model is best optimized for this business problem.

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

- The median values of the commision, duration and sales are higher for the customers who have made the claim. Hence, we can say that the customers who brings in higher sales and commission and travels for long duration tend to claim their insurance policy.
- C2B insurance agency faces the highest number of claims, among all the other agencies.
- Airlines type of insurance firms have almost equal amount of customers who claim and don't claim. Although, Travel Agency firms have gathered more customers and their claim frequency is comparatively very low.
- More number of customers among who opted for Silver and Gold plans claimed for insurance.
- We can put our main focus on the C2B agency which faces the highest number of claims and also belongs to Airlines industry.
- Sales, Commision, Duration, Age, Agency_Code_JZI, Product_Name_Customized Plan, Product_Name_Cancellation Plan are among the top predictors from CART and Random Forest models.
- ANN model is best optimized to be use to predict outcomes with an Accuracy of 0.78 and Recall of 0.53.