DM Project Report

Deepti Yadav
April’22
Date : 03/07/2022

Table of Contents

List of Figures

List of Tables

Problem 1 : Clustering

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

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).

Table 1- Dataset Description

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

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
205	13.89	14.02	0.8880	5.439	3.199	3.986	4.738
206	16.77	15.62	0.8638	5.927	3.438	4.920	5.795
207	14.03	14.16	0.8796	5.438	3.201	1.717	5.001
208	16.12	15.00	0.9000	5.709	3.485	2.270	5.443
209	15.57	15.15	0.8527	5.920	3.231	2.640	5.879

Table 2 - Dataset Information

```
<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 3 - Missing values Check

```
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
```

Observation:

7 variables and 210 records.

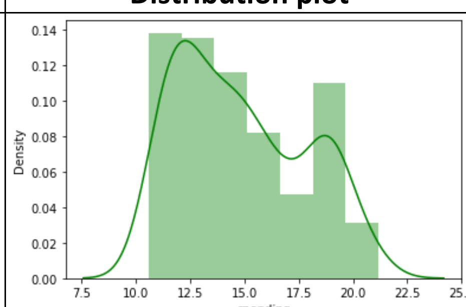
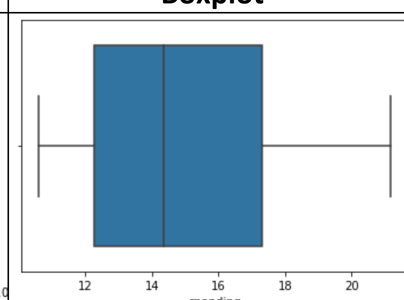
No missing record based on initial analysis.

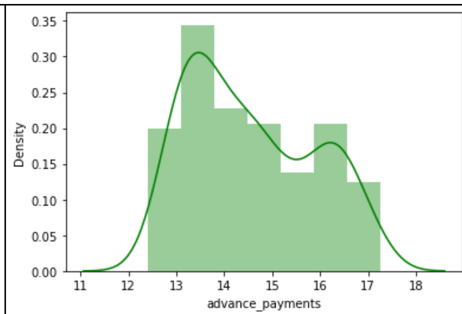
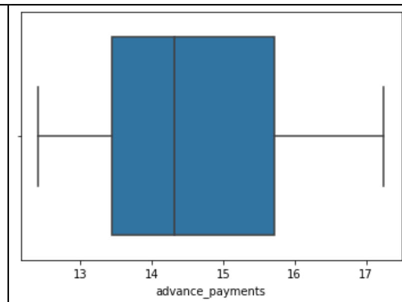
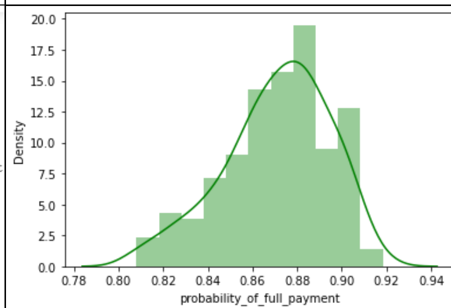
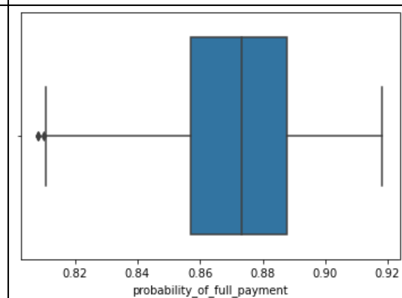
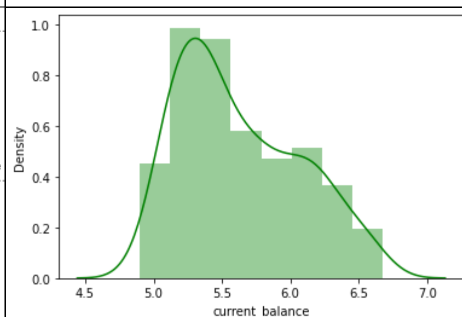
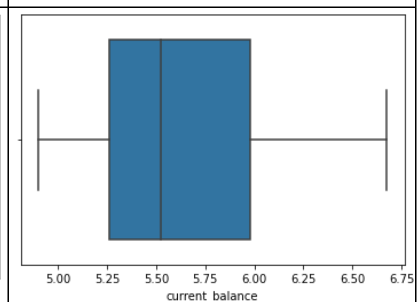
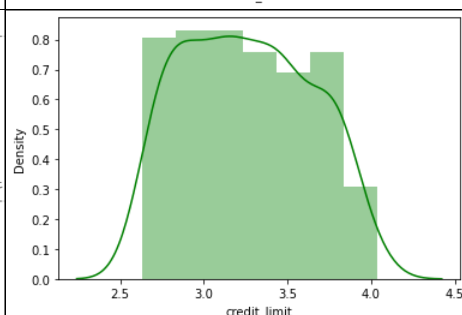
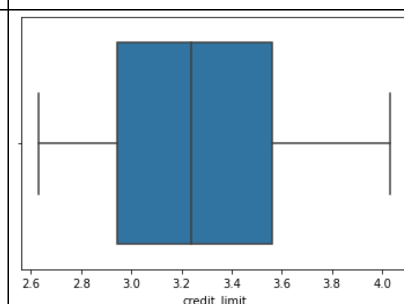
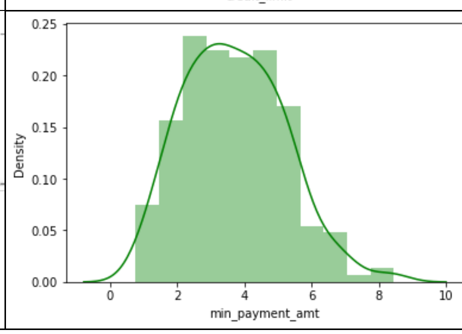
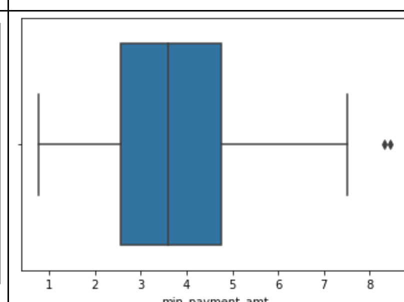
All the variables numeric type.

No duplicate rows found

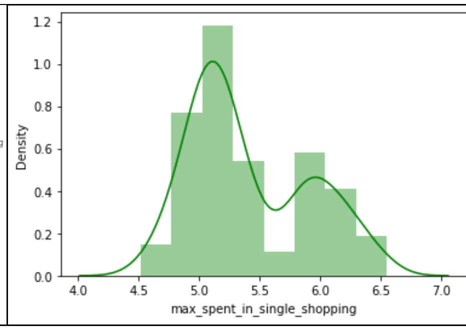
Univariate Analysis :

Table 4 – Univariate Analysis

Description	Distribution plot	Boxplot
Description of spending -- count 210.000000 mean 14.847524 std 2.909699 min 10.590000 25% 12.270000 50% 14.355000 75% 17.305000 max 21.180000 Name: spending, dtype: float64 Distribution of spending		

Description of advance_payments --- count 210.000000 mean 14.559286 std 1.305959 min 12.410000 25% 13.450000 50% 14.320000 75% 15.715000 max 17.250000 Name: advance_payments, dtype: float64 Distribution of advance_payments ---		
Description of probability_of_full_payment --- count 210.000000 mean 0.870999 std 0.023629 min 0.808100 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 ---		
Description of current_balance --- count 210.000000 mean 5.628533 std 0.443063 min 4.899000 25% 5.262250 50% 5.523500 75% 5.979750 max 6.675000 Name: current_balance, dtype: float64 Distribution of current_balance ---		
Description of credit_limit --- count 210.000000 mean 3.258605 std 0.377714 min 2.630000 25% 2.944000 50% 3.237000 75% 3.561750 max 4.033000 Name: credit_limit, dtype: float64 Distribution of credit_limit ---		
Description of min_payment_amt --- count 210.000000 mean 3.700201 std 1.503557 min 0.765100 25% 2.561500 50% 3.599000 75% 4.768750 max 8.456000 Name: min_payment_amt, dtype: float64 Distribution of min_payment_amt ---		

```
Description of max_spent_in_single_shopping
----------------------------------------------------------------------------
count    210.000000
mean       5.408071
std        0.491480
min        4.519000
25%        5.045000
50%        5.223000
75%        5.877000
max        6.550000
Name: max_spent_in_single_shopping, dtype: float64 Distribution of max_spent_in_single_shopping
----------------------------------------------------------------------------
```



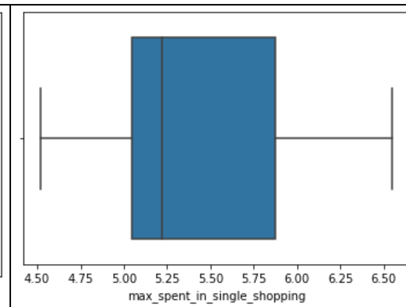


Table 5 – Skewness Analysis

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

Observations:

1) Outliers present in "probability_of_full_payment" & "min_payment_amt"

2) "probability_of_full_payment" is left skewed.

3) Other variables are right skewed.

Multivariate Analysis :

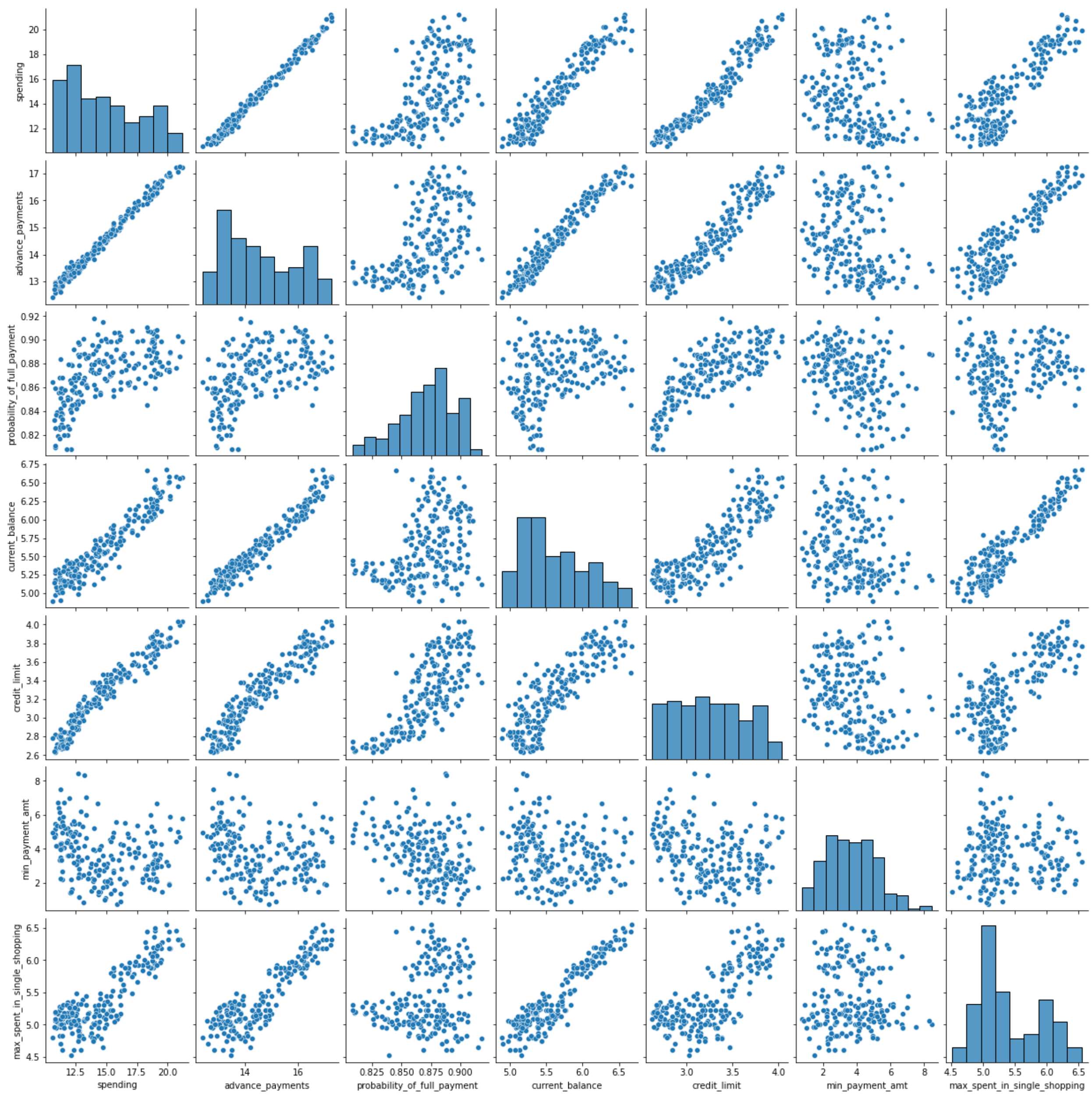


Figure 1 - Pairplot

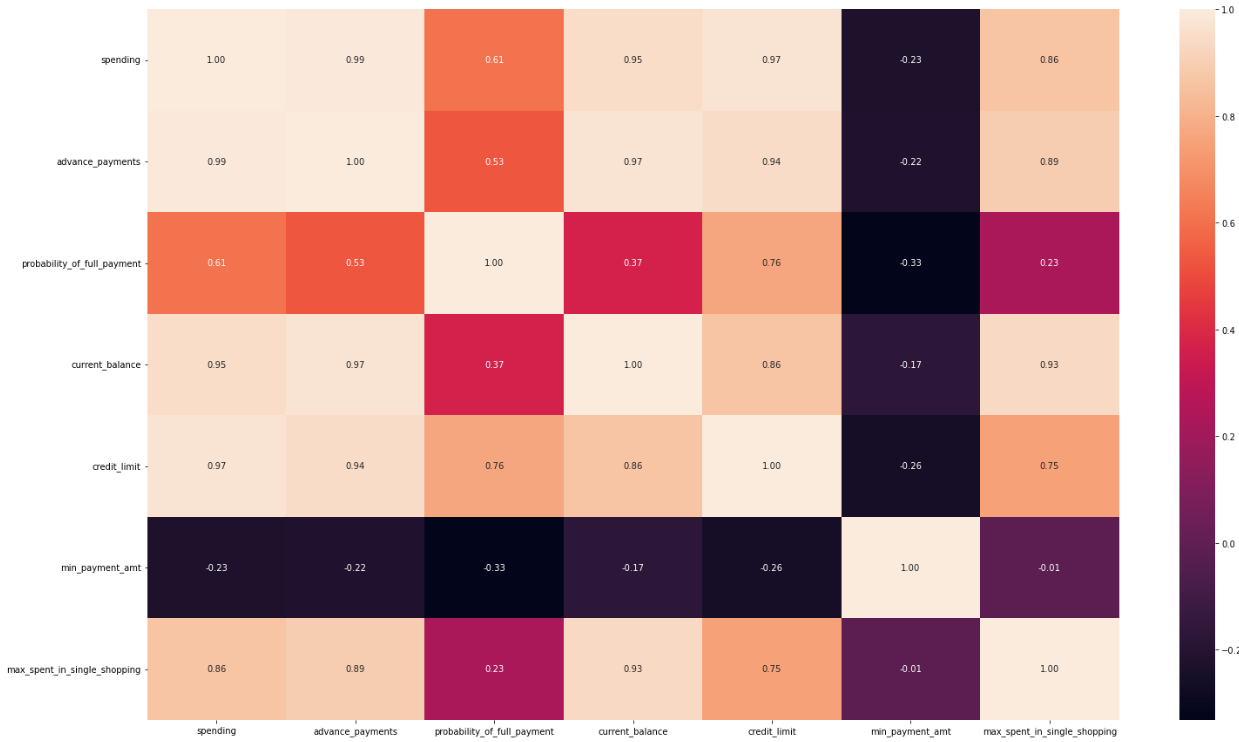


Figure 2 – Heat Map

Table 6 – Correlation Values

		correlation
advance_payments	**spending**	0.994341
	current_balance	0.972422
credit_limit	**spending**	0.970771
current_balance	**spending**	0.949985
advance_payments	**credit_limit**	0.944829
max_spent_in_single_shopping	**current_balance**	0.932806
advance_payments	**max_spent_in_single_shopping**	0.890784
spending	**max_spent_in_single_shopping**	0.863693
current_balance	**credit_limit**	0.860415
probability_of_full_payment	**credit_limit**	0.761635
max_spent_in_single_shopping	**credit_limit**	0.749131
spending	**probability_of_full_payment**	0.608288
advance_payments	**probability_of_full_payment**	0.529244

Observation

Strong positive correlation between
- advance_payments & spending,
- advance_payments & current_balance,
- credit_limit & spending
- current_balance & spending
- advance_payments & credit_limit
- max_spent_in_single_shopping & current_balance

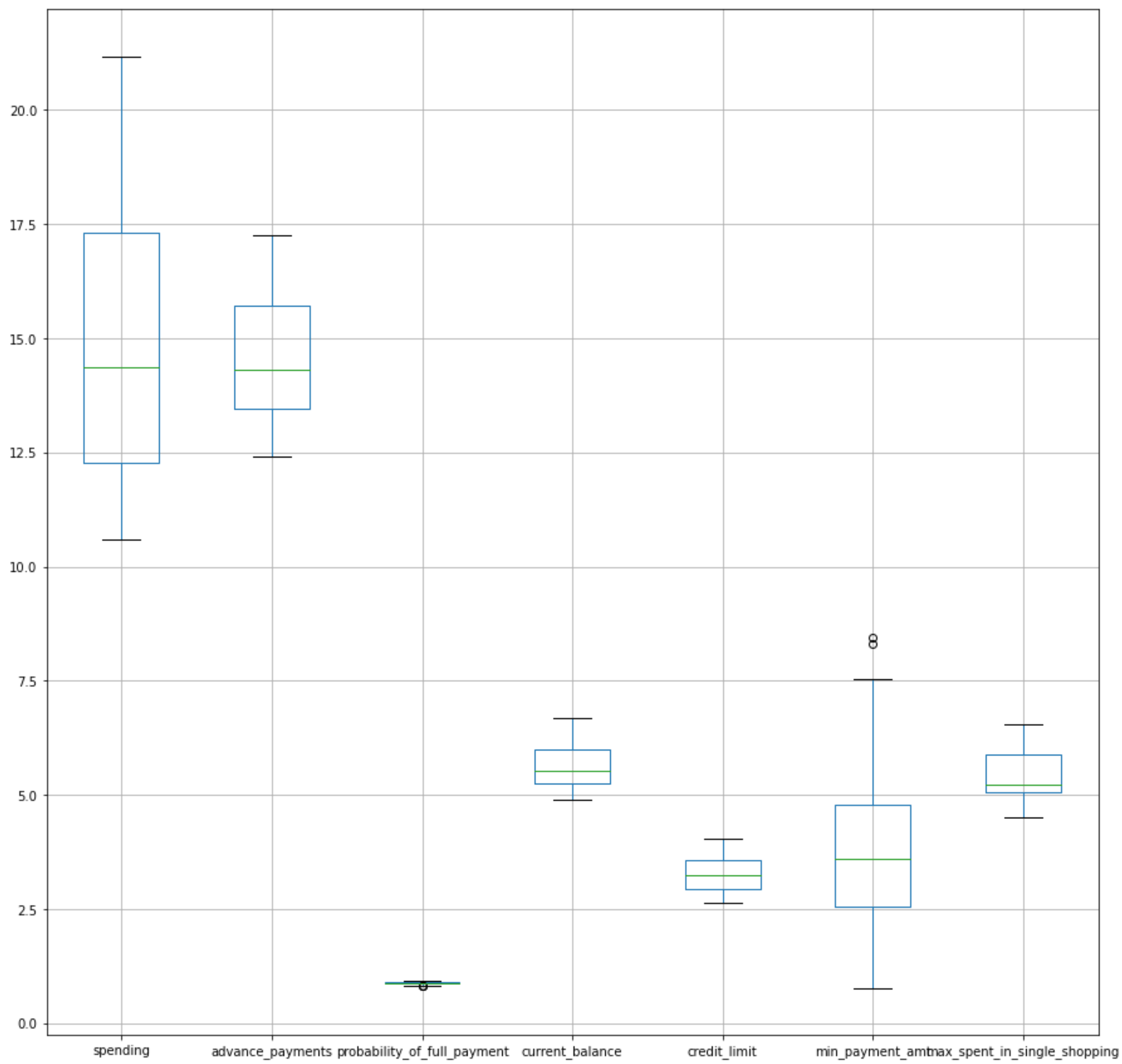


Figure 3 – Boxplot with Outliers

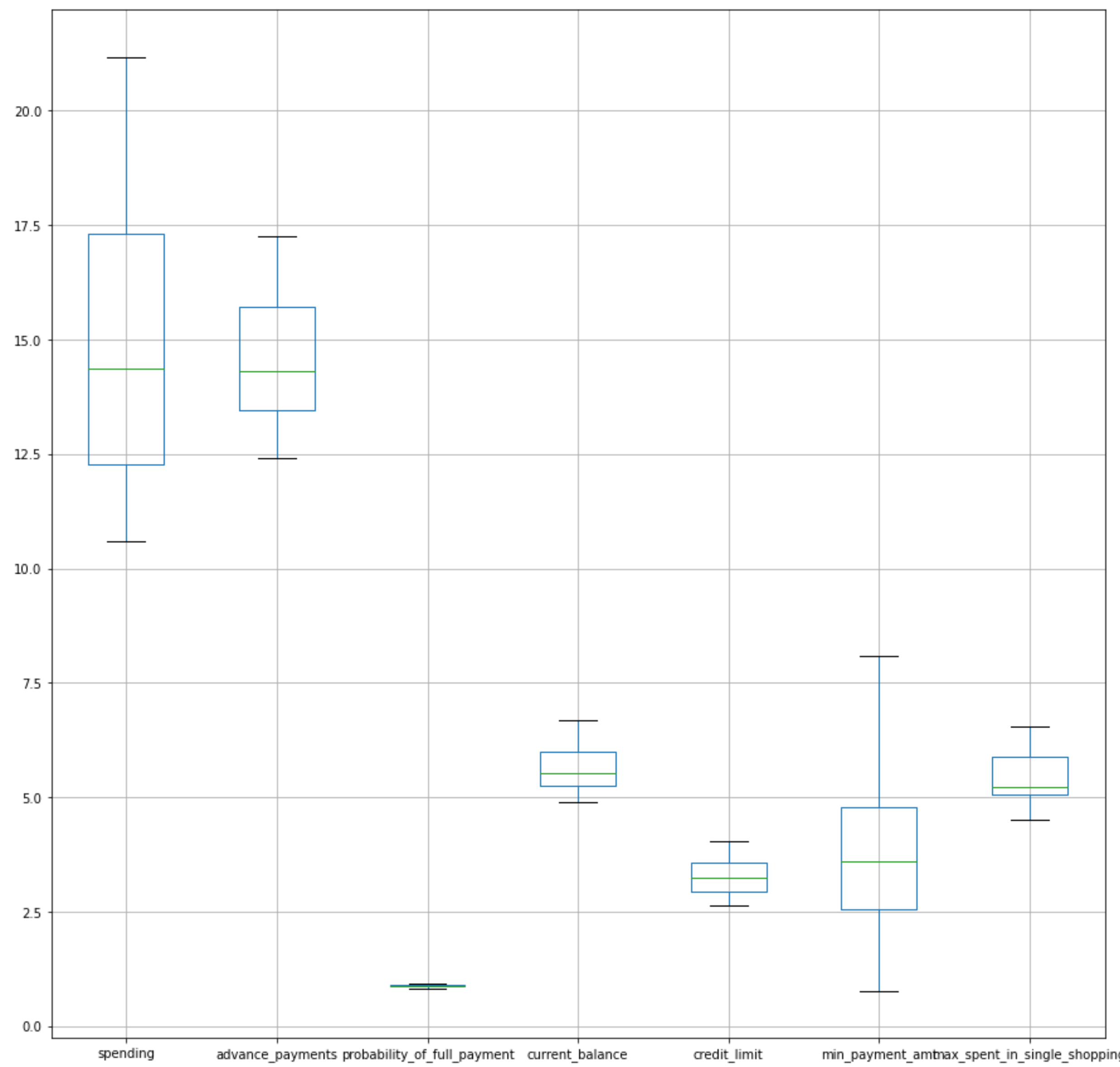


Figure 4 – Boxplot without Outliers

There are no outliers after treating them

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

Standardization or scaling is an important aspect of data pre-processing. Since, the range of values of data may vary widely, it becomes a necessary step in data preprocessing while using machine learning algorithms. All machine learning

algorithms are dependent on the scaling of data. for clustering too, scaling is

usually applied. In this case we can see that variables are in 100s, 1000s and 10000s.Since the data in these variables are of different scales, it is tough to compare these variables. In this method, we convert variables with different scales of measurements into a single scale. Scaling normalizes the data using the formula (x-mean)/standard deviation. Standard deviation becomes 1 and mean becomes zero.

StandardScaler is used for scaling and data is given below.

Table 7 – Data after Scaling

	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
...	...	...	...	...	...	...	...
205	-0.329866	-0.413929	0.721222	-0.428801	-0.158181	0.190536	-1.366631
206	0.662292	0.814152	-0.305372	0.675253	0.476084	0.813214	0.789153
207	-0.281636	-0.306472	0.364883	-0.431064	-0.152873	-1.322158	-0.830235
208	0.438367	0.338271	1.230277	0.182048	0.600814	-0.953484	0.071238
209	0.248893	0.453403	-0.776248	0.659416	-0.073258	-0.706813	0.960473

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

Choosing average linkage method and creating Dendogram

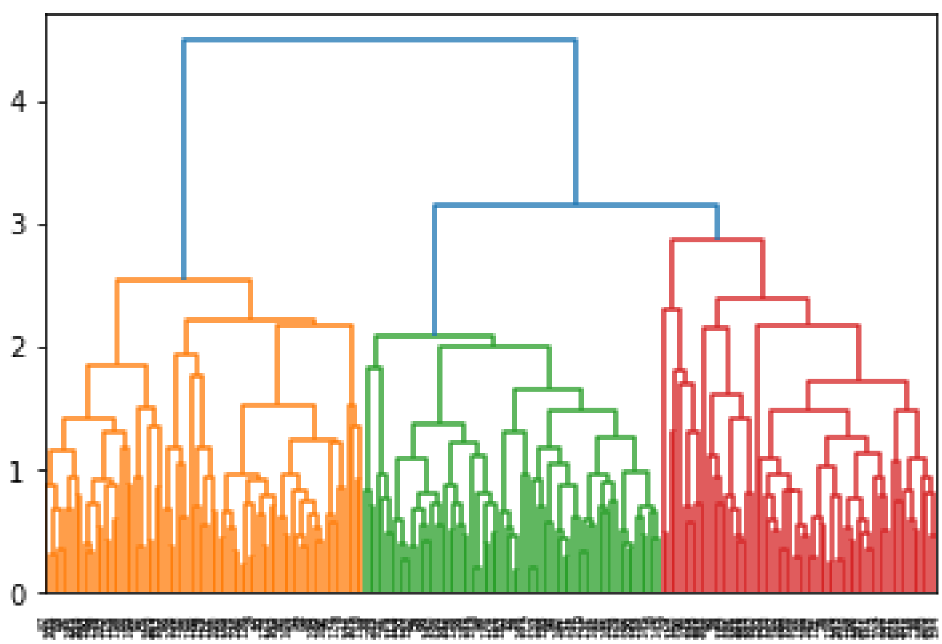


Figure 5 – Dendogram

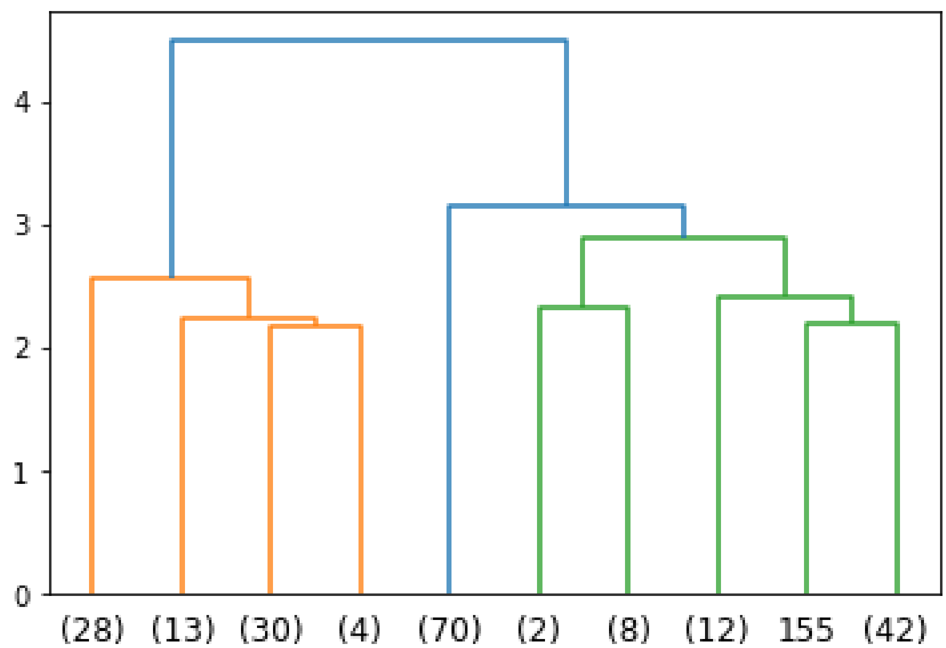


Figure 6 – Truncated Dendogram

Importing fcluster module to create clusters

```
array([1, 3, 1, 2, 1, 3, 2, 2, 1, 2, 1, 1, 2, 1, 3, 3, 3, 2, 2, 2, 2, 2,
       1, 2, 3, 1, 3, 2, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 2, 1, 1, 3, 1, 1,
       2, 2, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 3, 2, 2, 1, 3, 1,
       1, 3, 1, 2, 3, 2, 1, 1, 2, 1, 3, 2, 1, 3, 3, 3, 3, 1, 2, 1, 1, 1,
       1, 3, 3, 1, 3, 2, 2, 1, 1, 1, 2, 1, 3, 1, 3, 1, 3, 1, 1, 2, 3, 1,
       1, 3, 1, 2, 2, 1, 3, 3, 2, 1, 3, 2, 2, 2, 3, 3, 1, 2, 3, 3, 2, 3,
       3, 1, 2, 1, 1, 2, 1, 3, 3, 3, 2, 2, 2, 2, 1, 2, 3, 2, 3, 2, 3, 1,
       3, 3, 2, 2, 3, 1, 1, 2, 1, 1, 1, 2, 1, 3, 3, 2, 3, 2, 3, 1, 1, 1,
       3, 2, 3, 2, 3, 2, 3, 3, 1, 1, 3, 1, 3, 2, 3, 3, 2, 1, 3, 1, 1, 2,
       1, 2, 3, 3, 3, 2, 1, 3, 1, 3, 3, 1], dtype=int32)
```

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.

Calculating WSS for other values of K - Elbow Method

Clusters with K = 1 : wss - 1469.9999999999998,
Clusters with K = 2 : wss - 659.1474009548499,
Clusters with K = 3 : wss - 430.298481751223,
Clusters with K = 4 : wss - 371.221763926848,
Clusters with K = 5 : wss - 325.944677114075,
Clusters with K = 6 : wss - 289.7657733967166,

Clusters with K = 7 : wss - 262.22500296635945,
Clusters with K = 8 : wss - 239.71459430002525,
Clusters with K = 9 : wss - 222.40017089869343,
Clusters with K = 10 : wss - 208.48821050568935

WSS reduces as K keeps increasing

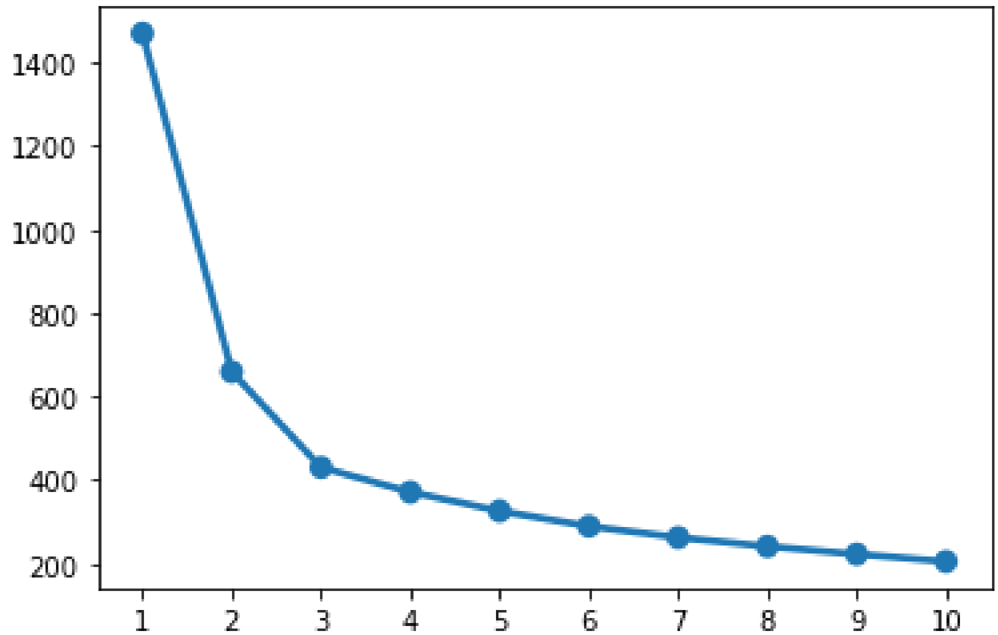


Figure 7 – Elbow curve

From the above curve there is a sharp dip at K=3.

Also silhouette score is better for 3 clusters (0.40) than for 4 clusters (0.32).

So selecting K=3 for further evaluation.

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

Table 8 – Cluster profile for hierarchical clustering

clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
1	18.129200	16.058000	0.881595	6.135747	3.648120	3.650200	5.987040	75
2	11.916857	13.291000	0.846845	5.258300	2.846000	4.619000	5.115071	70
3	14.217077	14.195846	0.884869	5.442000	3.253508	2.759007	5.055569	65

Table 9 – Cluster profile for K-Means clustering

Clus_kmeans	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	sil_width	freq
0	11.856944	13.247778	0.848330	5.231750	2.849542	4.733892	5.101722	0.399556	72
1	18.495373	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701	0.468077	67
2	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803	0.338593	71

Recommendation for different promotional strategies for different clusters

Cluster 0 for Kmeans & Cluster 1 for hierarchical / : High Spending Group

- Giving any reward points might increase their purchases.
- maximum max_spent_in_single_shopping is high for this group, so can be offered discount/offer on next transactions upon full payment
- Increase there credit limit and
- Increase spending habits
- Give loan against the credit card, as they are customers with good repayment record.
- Tie up with luxary brands, which will drive more one_time_maximun spending

Cluster 2 for Kmeans & Cluster 3 for hierarchical: Moderate Spending Group

- They are potential target customers who are paying bills and doing purchases and maintaining comparatively good credit score. So we can increase credit limit or can lower down interest rate.
- Promote premium cards/loyality cars to increase transcations.
- Increase spending habits by trying with premium ecommerce sites, travel portal, travel airlines/hotel, as this will encourge them to spend more

Cluster 1 for Kmeans & Cluster 2 for hierarchical : Low Spending Group

- customers should be given remainders for payments. Offers can be provided on early payments to improve their payment rate.
- Increase there spending habits by tieing up with grocery stores, utlities (electircity, phone, gas, others)

Problem 2 : CART- RF - ANN

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.

Attribute Information:

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

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

Table 10- Dataset Description

	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

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product_Name	Destination
2995	28	CWT	Travel Agency	Yes	166.53	Online	364	256.20	Gold Plan	Americas
2996	35	C2B	Airlines	No	13.50	Online	5	54.00	Gold Plan	ASIA
2997	36	EPX	Travel Agency	No	0.00	Online	54	28.00	Customised Plan	ASIA
2998	34	C2B	Airlines	Yes	7.64	Online	39	30.55	Bronze Plan	ASIA
2999	47	JZI	Airlines	No	11.55	Online	15	33.00	Bronze Plan	ASIA

Table 11 - Dataset Information

```
<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 12 - Missing values Check

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

Observation

- 10 variables are present
- Age, Commision, Duration, Sales are numeric variable & rest are object/categorial variables
- 3000 records, no missing one
- 9 independant variable and one target variable - Clamied


```
<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 13 - Geting unique counts of all Objects

```
Agency_Code
 EPX    1365
C2B     924
CWT     472
JZI     239
Name: Agency_Code, dtype: int64


Type
 Travel Agency    1837
Airlines         1163
Name: Type, dtype: int64


Claimed
 No     2076
Yes     924
Name: Claimed, dtype: int64


Channel
 Online     2954
Offline      46
Name: Channel, dtype: int64


Destination
 ASIA        2465
Americas     320
EUROPE       215
Name: Destination, dtype: int64


Product Name
 Customised Plan      1136
Cancellation Plan     678
Bronze Plan           650
Silver Plan           427
Gold Plan             109
Name: Product_Name, dtype: int64
```

Table 14 – Data Five Point summary

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	3000.0	NaN	NaN	NaN	38.091	10.463518	8.0	32.0	36.0	42.0	84.0
Agency_Code	3000	4	EPX	1365	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Type	3000	2	Travel Agency	1837	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Claimed	3000	2	No	2076	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Commision	3000.0	NaN	NaN	NaN	14.529203	25.481455	0.0	0.0	4.63	17.235	210.21
Channel	3000	2	Online	2954	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Duration	3000.0	NaN	NaN	NaN	70.001333	134.053313	-1.0	11.0	26.5	63.0	4580.0
Sales	3000.0	NaN	NaN	NaN	60.249913	70.733954	0.0	20.0	33.0	69.0	539.0
Product_Name	3000	5	Customised Plan	1136	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Destination	3000	3	ASIA	2465	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Observation

- duration has negative value, it is not possible. Wrong entry.
- Commision & Sales- mean and median varies significantly

Replacing Duration of tour with minimum possible value that is 1.

Table 15 – Data Five Point summary (modified)

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	3000.0	NaN	NaN	NaN	38.091	10.463518	8.0	32.0	36.0	42.0	84.0
Agency_Code	3000	4	EPX	1365	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Type	3000	2	Travel Agency	1837	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Claimed	3000	2	No	2076	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Commision	3000.0	NaN	NaN	NaN	14.529203	25.481455	0.0	0.0	4.63	17.235	210.21
Channel	3000	2	Online	2954	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Duration	3000.0	NaN	NaN	NaN	70.002667	134.052619	1.0	11.0	26.5	63.0	4580.0
Sales	3000.0	NaN	NaN	NaN	60.249913	70.733954	0.0	20.0	33.0	69.0	539.0
Product_Name	3000	5	Customised Plan	1136	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Destination	3000	3	ASIA	2465	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 16 – Checking for Duplicates

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

139 rows × 10 columns

As the customer ID are not available, whether the duplicates are really duplicates cannot be verified. Also removing duplicate is resulting in overfitting of model. Therefore decision is to keep duplicates.

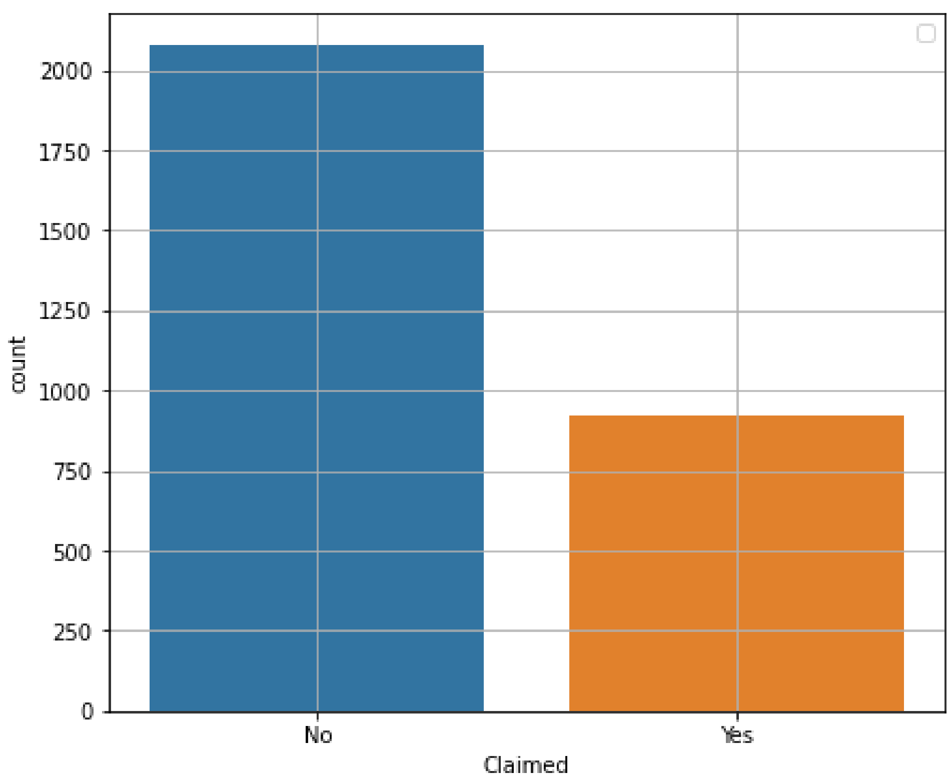


Figure 8 – Proportion of observations in Target class

```
No      0.692
Yes     0.308
Name: Claimed, dtype: float64
```

The target variables are unbalanced type as we almost 50% yes compared to No

Univariate Analysis :

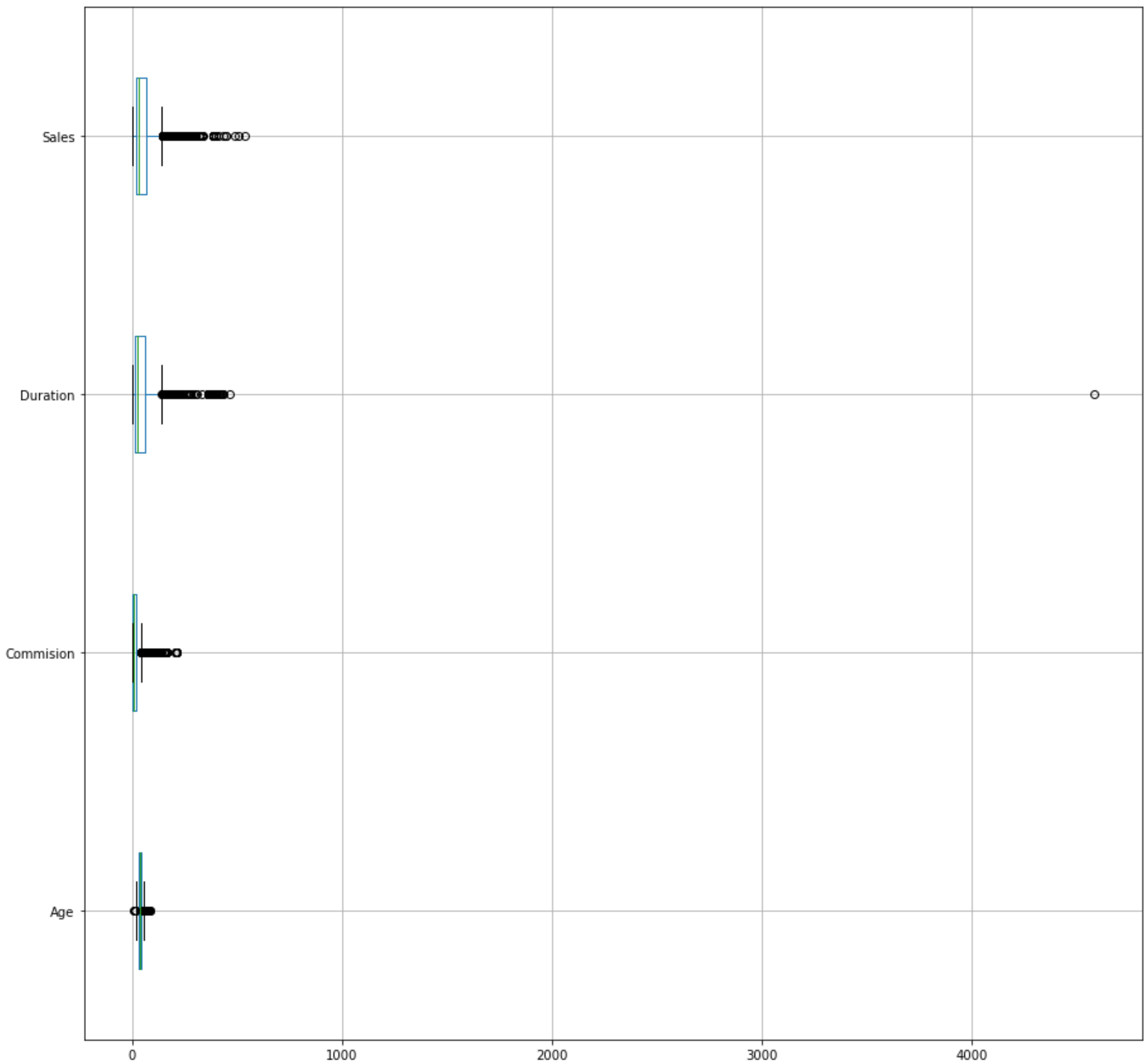


Figure 9 – Boxplot for continuous variables

Outliers exists for every variable, and also has many outliers**.**
Outliers must be treated.

Outlier treatment:

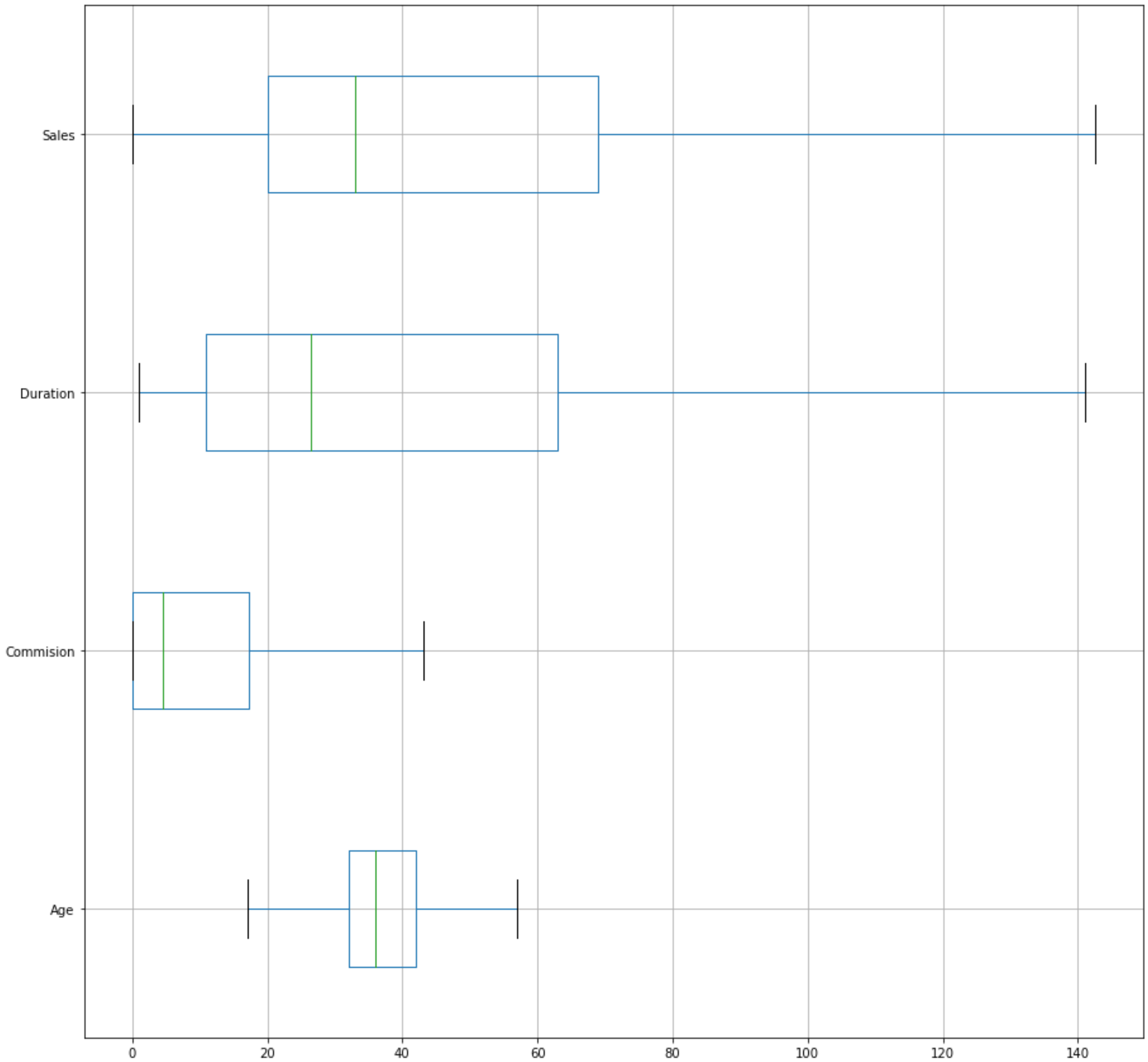


Figure 10 – Boxplot for continuous variables (without outliers)

There are no outliers after treating them

Multivariate Analysis :

Checking pairwise distribution of the continuous variables

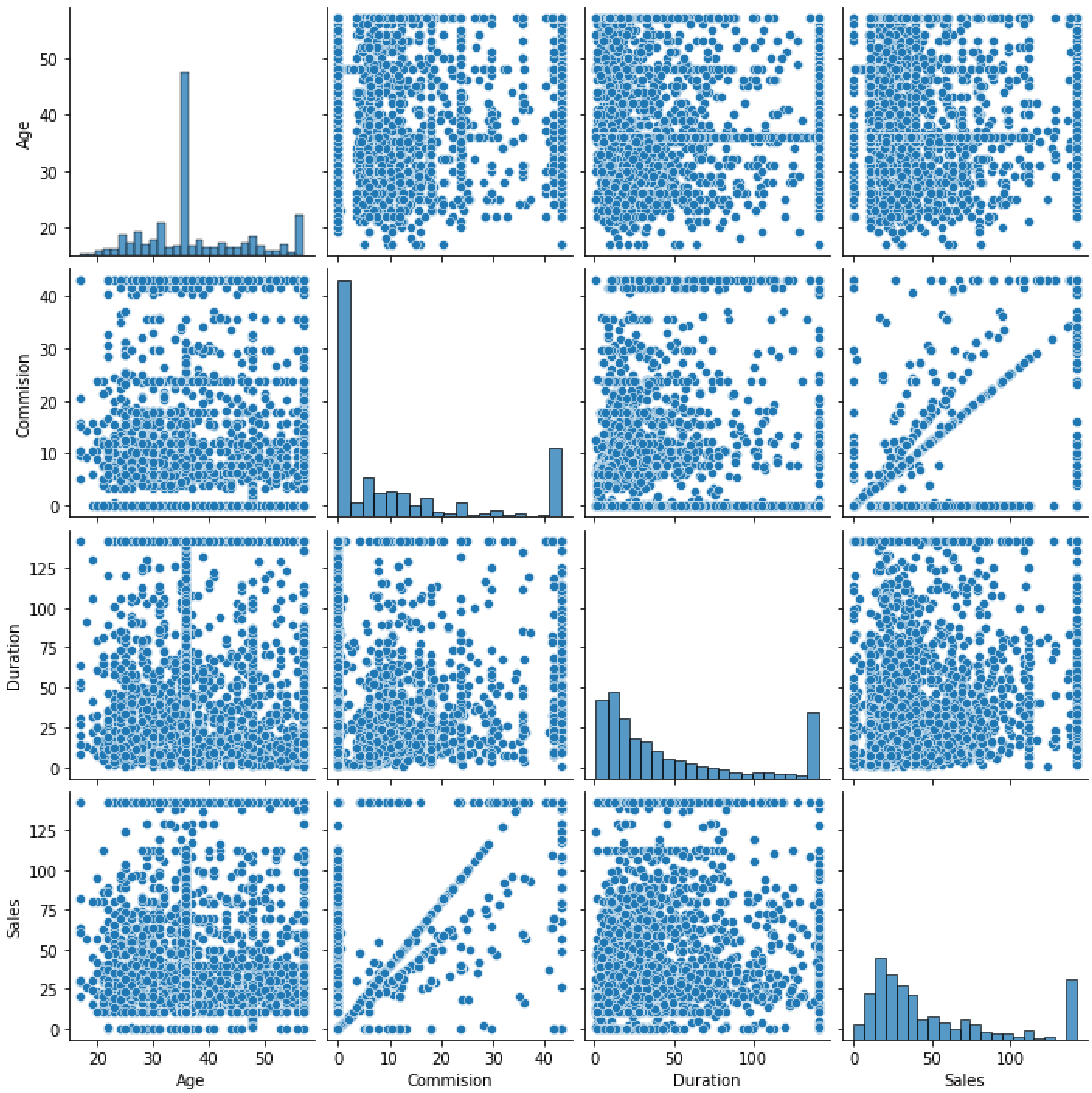


Figure 11 - Pairplot

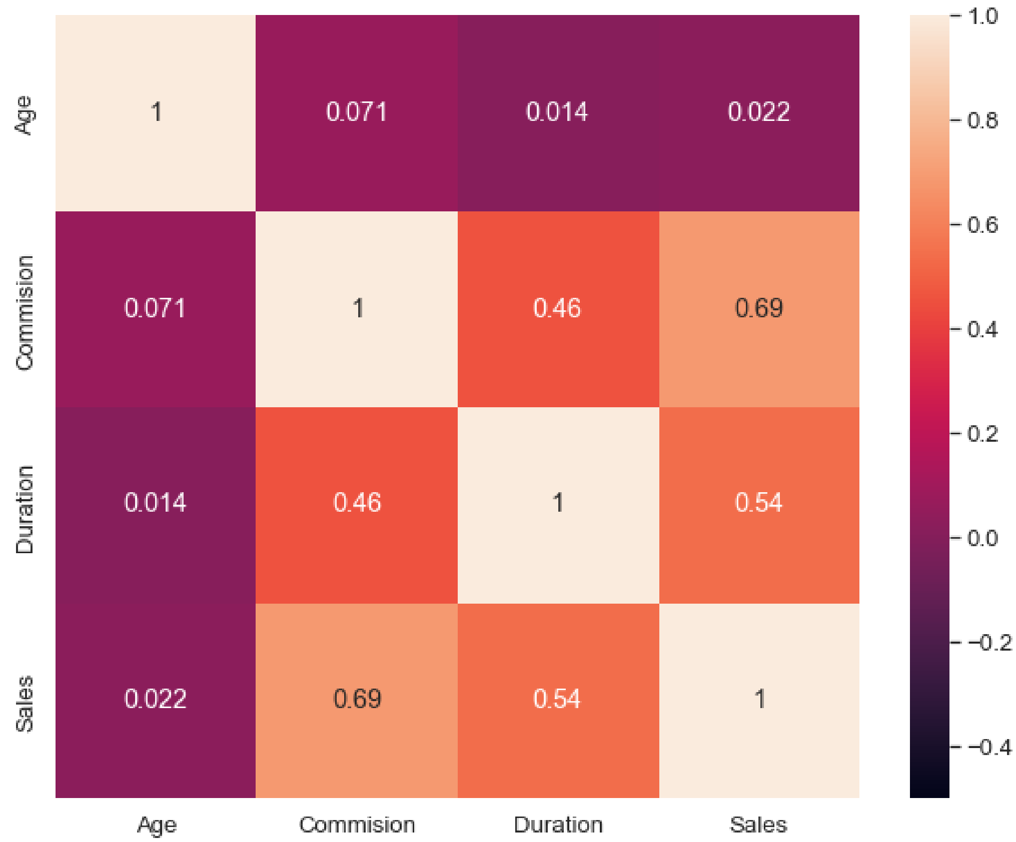


Figure 12 – Heat Map

Table 17 – Correlation Coefficients

		correlation
Sales	**Commision**	0.686219
Type	**Agency_Code**	0.552247
Sales	**Duration**	0.542824

Observation

There seems to be a clear correlation between Sales and commission.

Correlation also exists between

- Commission and Duration

- Sales and Duration

Decision tree in Python can take only numerical / categorical columns. It cannot take string / object types.

Table 18 – Dataset (All categorical/numerical)

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product_Name	Destination
0	48.0	0	0	0	0.70	1	7.0	2.51	2	0
1	36.0	2	1	0	0.00	1	34.0	20.00	2	0
2	39.0	1	1	0	5.94	1	3.0	9.90	2	1
3	36.0	2	1	0	0.00	1	4.0	26.00	1	0
4	33.0	3	0	0	6.30	1	53.0	18.00	0	0

Table 19 – Dataset information (All categorical/numerical)

```
<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   float64
 1   Agency_Code   3000 non-null   int8
 2   Type          3000 non-null   int8
 3   Claimed       3000 non-null   int8
 4   Commision     3000 non-null   float64
 5   Channel       3000 non-null   int8
 6   Duration      3000 non-null   float64
 7   Sales         3000 non-null   float64
 8   Product_Name  3000 non-null   int8
 9   Destination   3000 non-null   int8
dtypes: float64(4), int8(6)
memory usage: 111.5 KB
```

Table 20 – Variables unique code

```
Agency_Code
 2    1365
0     924
1     472
3     239
Name: Agency_Code, dtype: int64


Type
 1    1837
0    1163
Name: Type, dtype: int64


Claimed
 0    2076
1     924
Name: Claimed, dtype: int64


Channel
 1    2954
0      46
Name: Channel, dtype: int64


Destination
 0    2465
1     320
2     215
Name: Destination, dtype: int64


Product Name
 2    1136
1     678
0     650
4     427
3     109
Name: Product Name, dtype: int64
```

Label Encoding has been done and all columns are converted to number

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

Splitting data into training and test set in 30% test data

```
X_train (2100, 9)
X_test (900, 9)
train_labels (2100,)
test_labels (900,)
Total Obs 3000
```

Building classification model CART

```
param_grid = {
    'max_depth': [8,9,10],
    'min_samples_leaf': [15,20,25],
    'min_samples_split': [45,60,75]
}

dt_model = DecisionTreeClassifier()

grid_search = GridSearchCV(estimator = dt_model, param_grid = param_grid, cv = 3)
```

Best paramters

```
DecisionTreeClassifier

DecisionTreeClassifier(max_depth=8, min_samples_leaf=25, min_samples_split=60,
                       random_state=1)
```

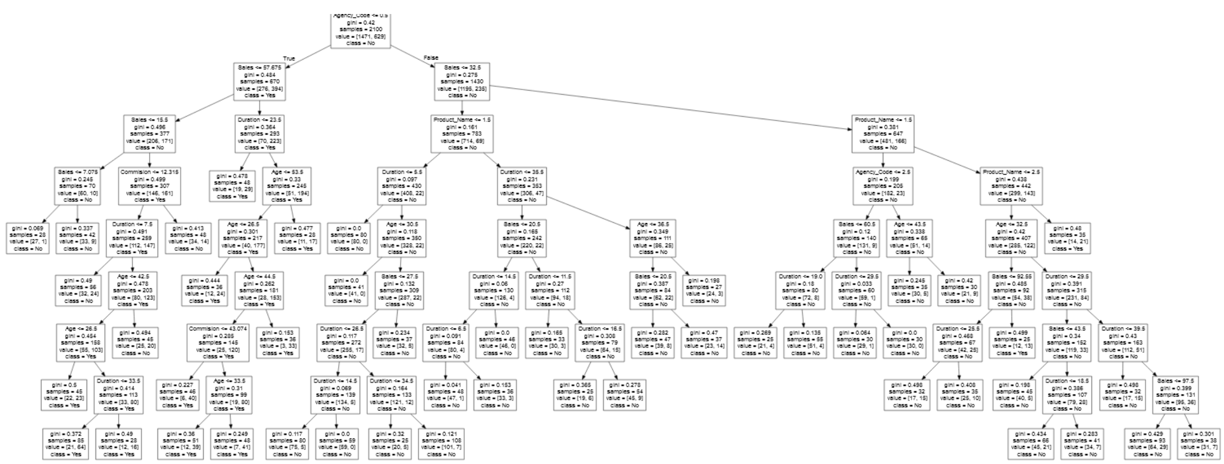


Figure 13 – Decision - CART

Feature importance with tuning hyper parameters

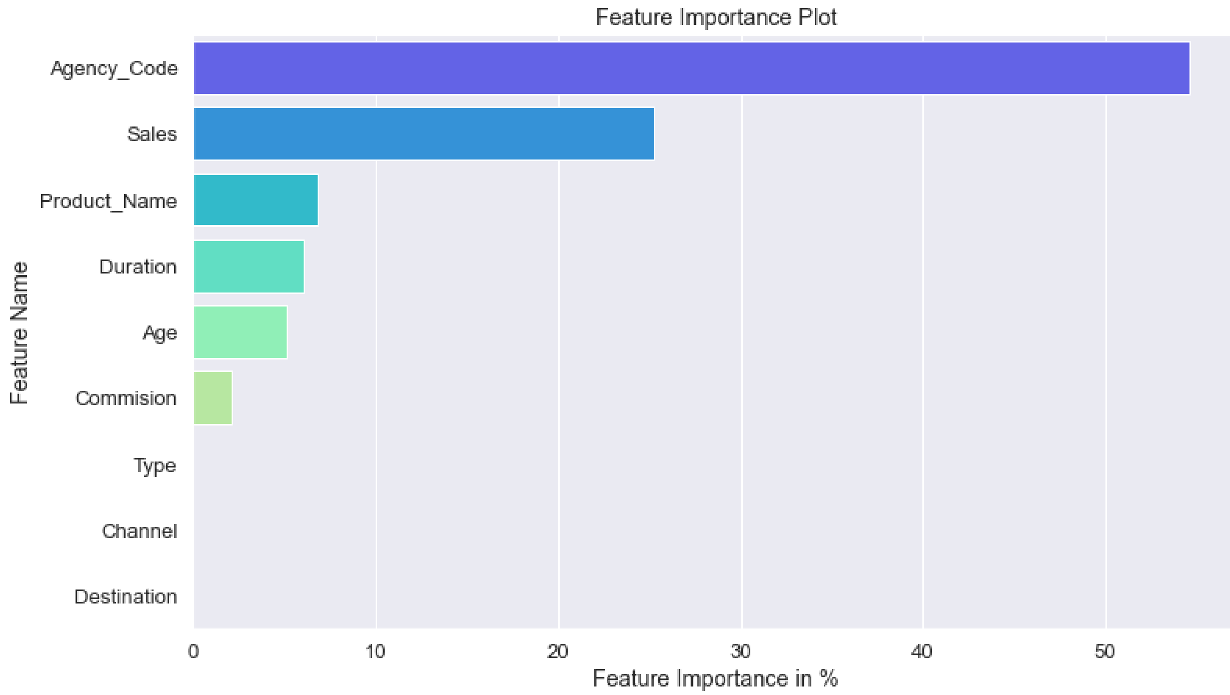


Figure 14 – Feature importance - CART

	Imp
Agency_Code	0.545688
Sales	0.252360
Product_Name	0.068850
Duration	0.060773
Age	0.051264
Commision	0.021065
Type	0.000000
Channel	0.000000
Destination	0.000000

Getting the Predicted Probabilities

Table 21 – Predicted Probability - CART

	0	1
0	0.966667	0.033333
1	0.555556	0.444444
2	0.247059	0.752941
3	0.130435	0.869565
4	0.935185	0.064815

Builiding Random Forest Classifier

Model with tuning hyper parameters

```
GridSearchCV
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [5, 10, 15], 'max_features': [4, 5, 6, 7],
                         'min_samples_leaf': [10, 50, 70],
                         'min_samples_split': [30, 50, 70],
                         'n_estimators': [200, 250, 300]})
estimator: RandomForestClassifier
RandomForestClassifier
```

```
RandomForestClassifier
RandomForestClassifier(max_depth=10, max_features=5, min_samples_leaf=10,
                       min_samples_split=50, n_estimators=250)
```

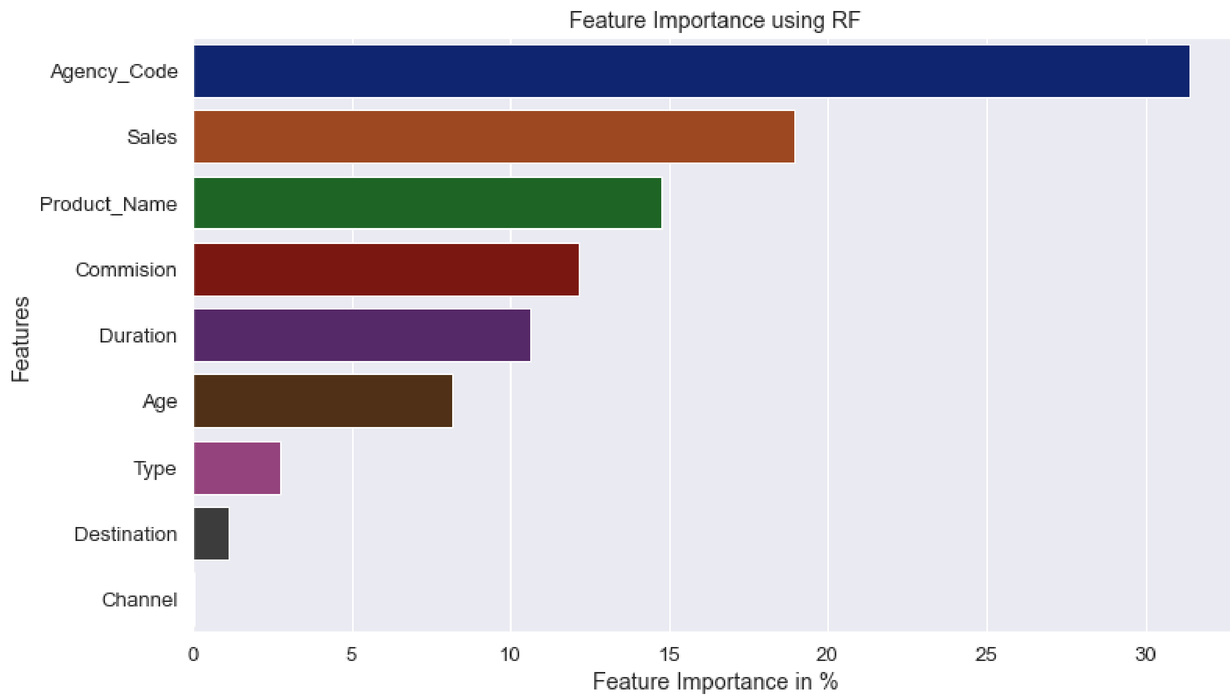


Figure 15 – Feature importance - RF

	Imp
Agency_Code	0.313772
Sales	0.189457
Product_Name	0.147803
Commision	0.121517
Duration	0.106087
Age	0.081551
Type	0.027649
Destination	0.011496
Channel	0.000669

Builiding ANN

Model with tuning hyper parameters

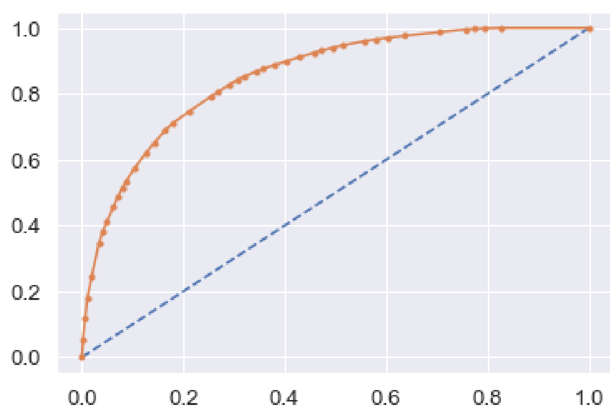
```
▾ GridSearchCV
GridSearchCV(cv=3, estimator=MLPClassifier(),
             param_grid={'hidden_layer_sizes': [(50, 100, 200)],
                         'max_iter': [2500, 3000, 4000], 'solver': ['adam'],
                         'tol': [0.01]})
  ▸ estimator: MLPClassifier
    ▸ MLPClassifier
```

```
{'hidden_layer_sizes': (50, 100, 200),
 'max_iter': 4000,
 'solver': 'adam',
 'tol': 0.01}
```

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.

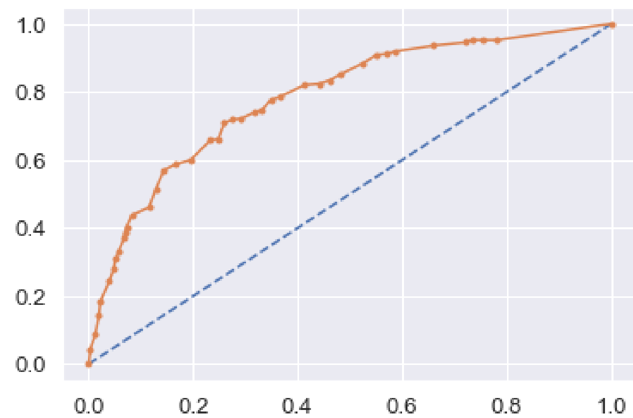
Classification model - CART

Table 22 – AUC and ROC for the training data (CART)



AUC: 0.855

Table 23 – AUC and ROC for the test data (CART)



AUC: 0.785

Table 24 – Confusion Matrix for the training data (CART)

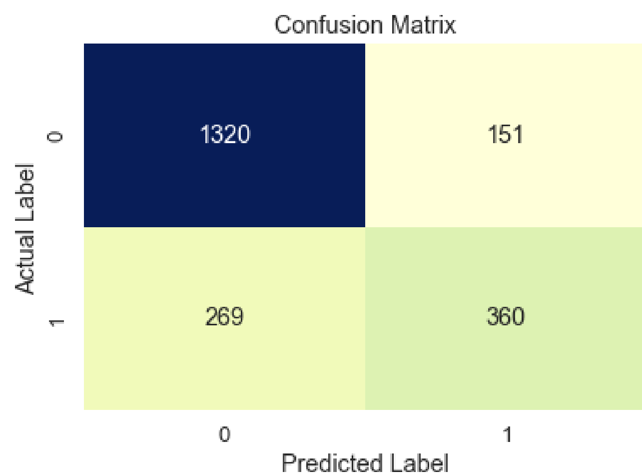


Table 25 – Confusion Matrix for test data (CART)

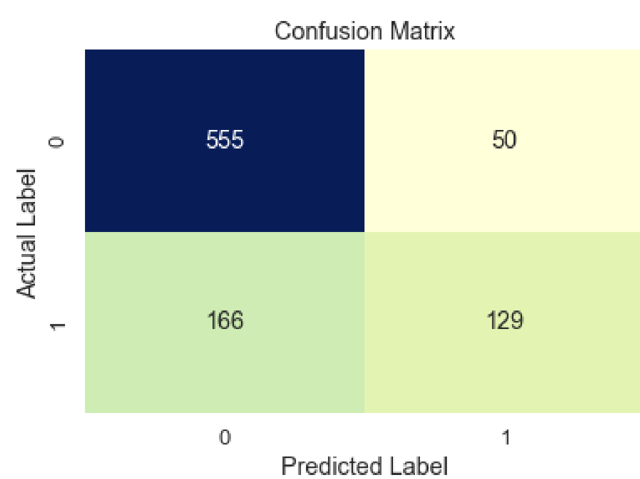


Table 26 – Classification report for training data (CART)

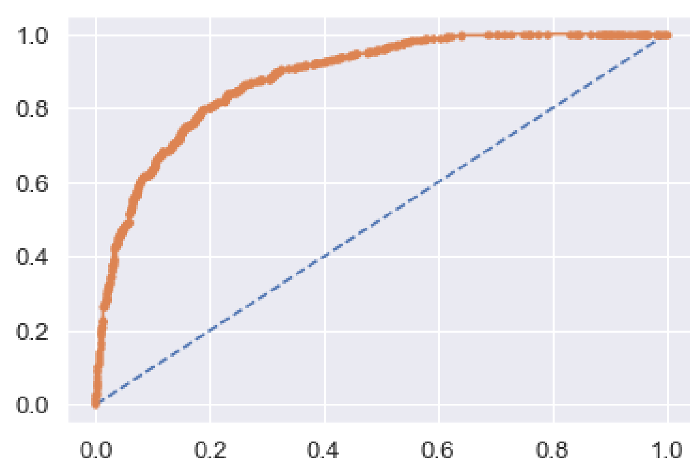
	precision	recall	f1-score	support
0	0.83	0.90	0.86	1471
1	0.70	0.57	0.63	629
accuracy			0.80	2100
macro avg	0.77	0.73	0.75	2100
weighted avg	0.79	0.80	0.79	2100

Table 27 – Classification report for test data (CART)

	precision	recall	f1-score	support
0	0.77	0.92	0.84	605
1	0.72	0.44	0.54	295
accuracy			0.76	900
macro avg	0.75	0.68	0.69	900
weighted avg	0.75	0.76	0.74	900

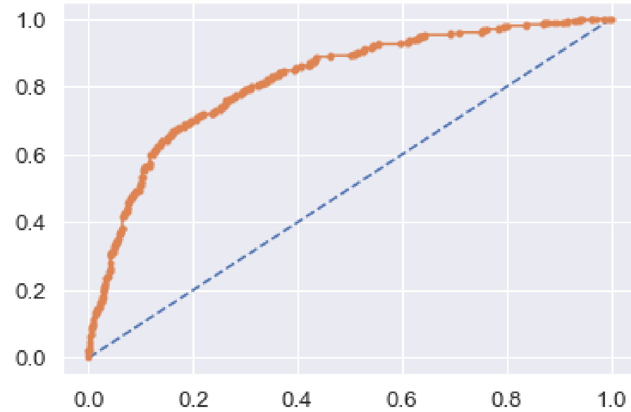
Random forest classification model

Table 28 – AUC and ROC for the training data (RF)



AUC: 0.885

Table 29 – AUC and ROC for the test data (RF)



AUC: 0.820

Table 30 – Confusion Matrix for the training data (RF)

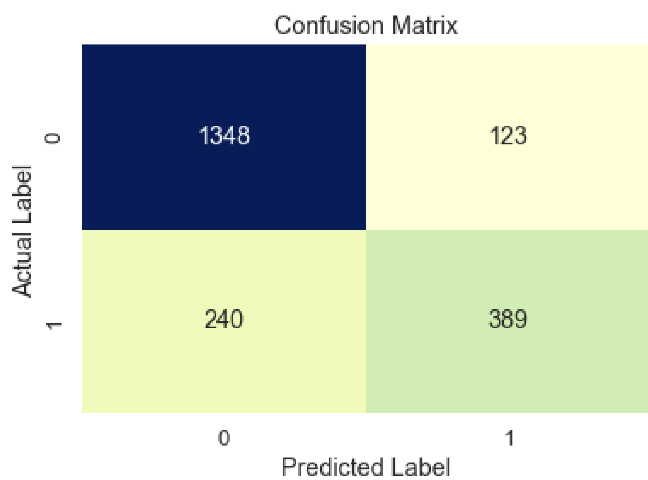


Table 31 – Confusion Matrix for test data (RF)

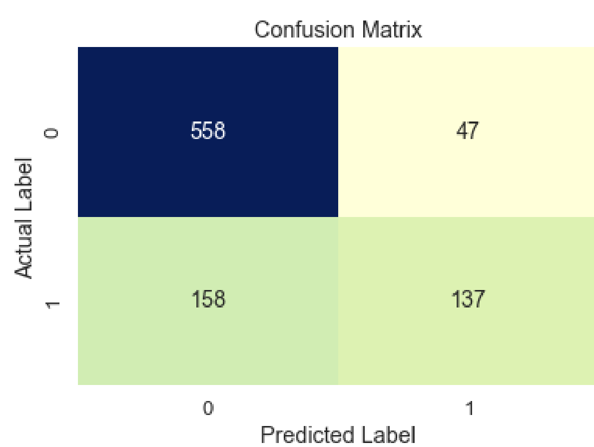


Table 32 – Classification report for training data (RF)

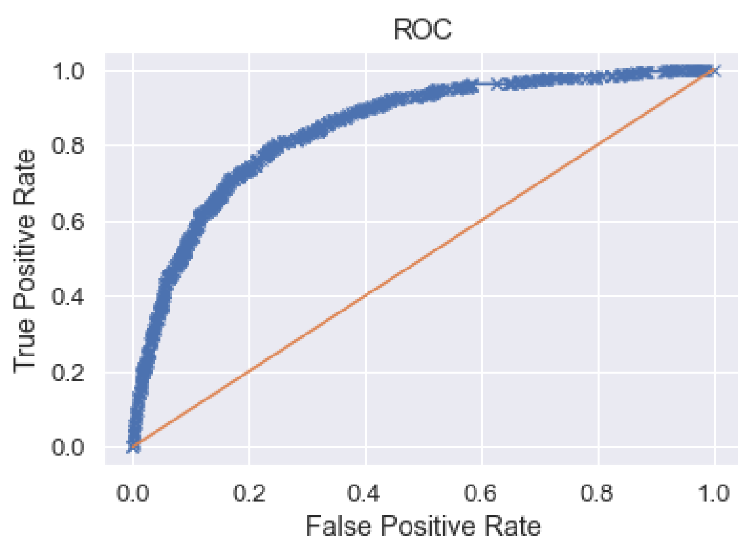
	precision	recall	f1-score	support
0	0.85	0.92	0.88	1471
1	0.76	0.62	0.68	629
accuracy			0.83	2100
macro avg	0.80	0.77	0.78	2100
weighted avg	0.82	0.83	0.82	2100

Table 33 – Classification report for test data (RF)

	precision	recall	f1-score	support
0	0.78	0.92	0.84	605
1	0.74	0.46	0.57	295
accuracy			0.77	900
macro avg	0.76	0.69	0.71	900
weighted avg	0.77	0.77	0.76	900

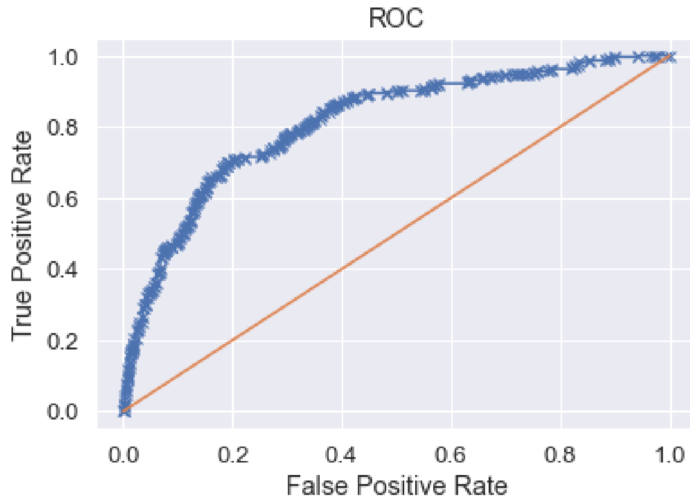
ANN Model

Table 34 – AUC and ROC for the training data (ANN)



AUC: 0.847

Table 35 – AUC and ROC for the test data (ANN)



AUC: 0.814

Table 36 – Confusion Matrix for the training data (ANN)

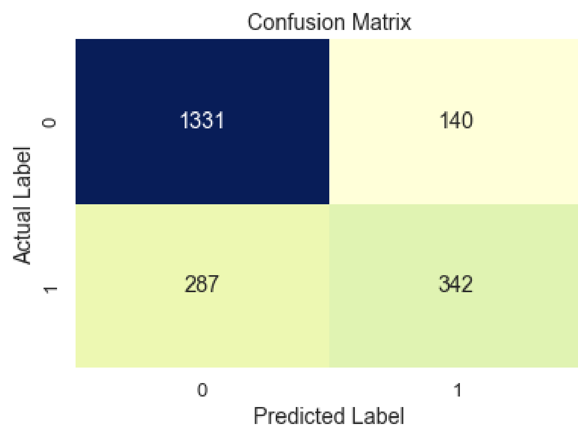


Table 37 – Confusion Matrix for test data (ANN)

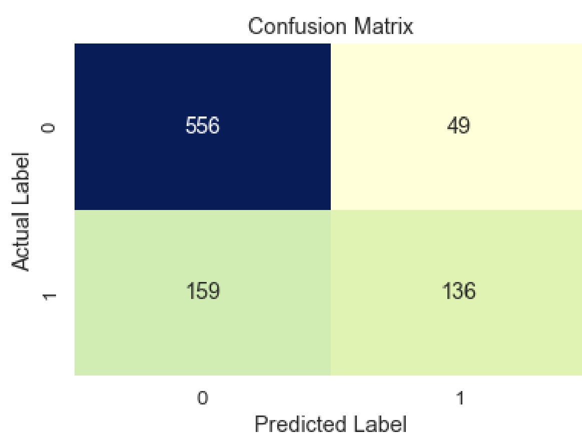


Table 38 – Classification report for training data (ANN)

	precision	recall	f1-score	support
0	0.82	0.90	0.86	1471
1	0.71	0.54	0.62	629
accuracy			0.80	2100
macro avg	0.77	0.72	0.74	2100
weighted avg	0.79	0.80	0.79	2100

Table 39 – Classification report for test data (ANN)

	precision	recall	f1-score	support
0	0.78	0.92	0.84	605
1	0.74	0.46	0.57	295
accuracy			0.77	900
macro avg	0.76	0.69	0.70	900
weighted avg	0.76	0.77	0.75	900

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

Table 40 – Comparison of all model

	CART Train	CART Test	Random Forest Train	Random Forest Test	ANN Train	ANN Test
Accuracy	0.80	0.76	0.83	0.77	0.80	0.77
AUC	0.855	0.785	0.885	0.820	0.847	0.814
Recall	0.57	0.44	0.62	0.46	0.54	0.46
Precision	0.70	0.72	0.76	0.74	0.71	0.74
F1 Score	0.63	0.54	0.68	0.57	0.62	0.57

Out of 3 models, Random forest is selected due to best Accuracy, AUC, Precision and F1 score and Recall.

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

For the business problem of Insurance providing firm, three model were analysed i.e. CART, Random forest and ANN for the predictions. These three models were evaluated on trainig and testing datasets and model performance were analysed.

The Accuracy, Precision and F1 score was computed using classification report. The confusion matrix, AUC_ROC score and ROC plot was computed and compared for different models.

All the models have peroformed well but to increase our accuracy in predictions, we can choose Random forest which creates multiple trees for decision making.

Recommendation & Insights:

- More real time unstructured data and past data should be collected in order to have balanced data.
- As per the data 90% of insurance is done by online channel. Almost all the offline business has a claimed associated, need to find why?
- Need to train the JZI agency resources to pick up sales as they are in bottom, need to run promotional marketing campaign or evaluate if we need to tie up with alternate agency
- Also based on the model we are getting 80%accuracy, so we need customer books airline tickets or plans, cross sell the insurance based on the claim data pattern.
- Other interesting fact is more sales happen via Agency than Airlines and the trend shows the claim are processed more at Airline. So we may need to deep dive into the process to understand the workflow and why?

Key performance indicators (KPI) The KPI's of insurance claims are:

- Reduce claims cycle time
- Increase customer satisfaction
- Combat fraud
- Optimize claims recovery

• Reduce claim handling costs Insights gained from data and AI-powered analytics could expand the boundaries of insurability, extend existing products, and give rise to new risk transfer solutions in areas like a non-damage business interruption and reputational damage.