DATA MINING PROJECT

Submitted by,

JIYA JACOB

PGP-DSBA ONLINE

JULY-B 2021

DATE:24/11/2021

BANK MARKETTING DATA

CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

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.

INTRODUCTION

The purpose of this whole exercise is to perform clustering techniques of the given data set using the two major techniques available mainly- kmeans clustering and hierarchical clustering. The values are clustered into different groups based on their inter cluster distances and finally these clusters are added to the existing data frame. The data is scaled before clustering to ensure efficient clustering.

DATA DESCRIPTION

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)

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

a. Sample 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: Dataset Sample

The data consists of details of 210 individuals along with their spending nature arranged in 7 columns such as spending, advance_payments ,probability_of_full_payment etc.

b. Shape of the dataset

```
df.shape
```

```
(210, 7)
```

There are total 210 rows and 7 columns in the given dataset.

c. Checking for the missing values in the dataset.

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

Table 2. Checking for the missing values in the dataset

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

d. Checking for the null values in the given 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
```

Table 3. Checking for the null values in the dataset

e. Checking for the datatype of variables present in the dataset.

```
spending                      float64
advance_payments              float64
probability_of_full_payment   float64
current_balance               float64
credit_limit                  float64
min_payment_amt               float64
max_spent_in_single_shopping  float64
dtype: object
```

Table 4. Checking the data type of variables in the data set

From the above table we can see that the data type of all the 7 variables in the given data set is float in nature.

f. Checking for the number of duplicated values

```
df.duplicated().sum()
```

```
0
```

From the above output we can see that there are no duplicated values in the given dataset.

g. Checking for the summary of the given dataset.

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

TABLE 5: SUMMARY OF THE DATA

From the summary of the dataset, we can see that mean of the spending and advance_payments are almost similar while the mean for probability_of_full_payment is least among all. The variables credit_limit and min_payment_amt share almost the similar mean whereas the current_balance and max_spent_in_single_shoppinng is also having the similar means. Among all variables, it is the spending that has the highest mean value, maximum value, standard deviation and lowest minimum value. probability_of_full_payment has the least value for all the statistical measurements.

Graphical Representation of univariate and bivariate analysis for continuous columns

1.Spending

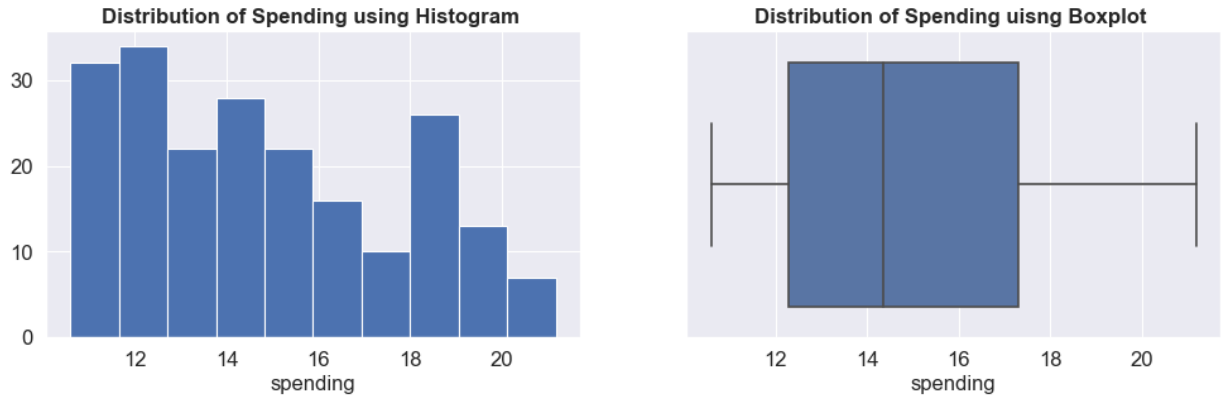

Fig 1: The box plot and the histogram showing the distribution of data of 'Spending' variable

From the univariate analysis using histogram, we see that people who have spent between the range of 12000-14000 are the most, approximately 1500, followed by the people who have spent in 5000-12000, approximately 500 and the least people spend in the amount range of 20000-21500, which is less than 10. From the bivariate analysis using boxplot, there are no presence of outliers. The second quartile(Q2) or median for the Spending variable is about 14.355. The lower or first quartile(Q1) is about 12.27 and the upper or the third quartile(Q3) is about 17.305. The inter quartile range (IQR) for the above boxplot is 5.035

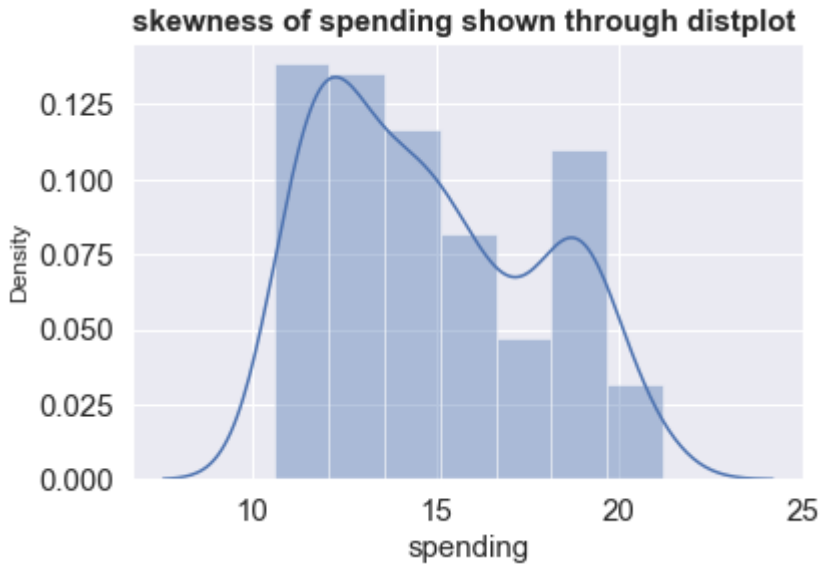

Fig 2: The distplot showing the skewness of data of 'Spending' variable

From the above distplot , we can see that Spending is positively skewed or right skewed, with median=14.355,mode=0 and 11.23,mean= 14.847

2.Advance_payments

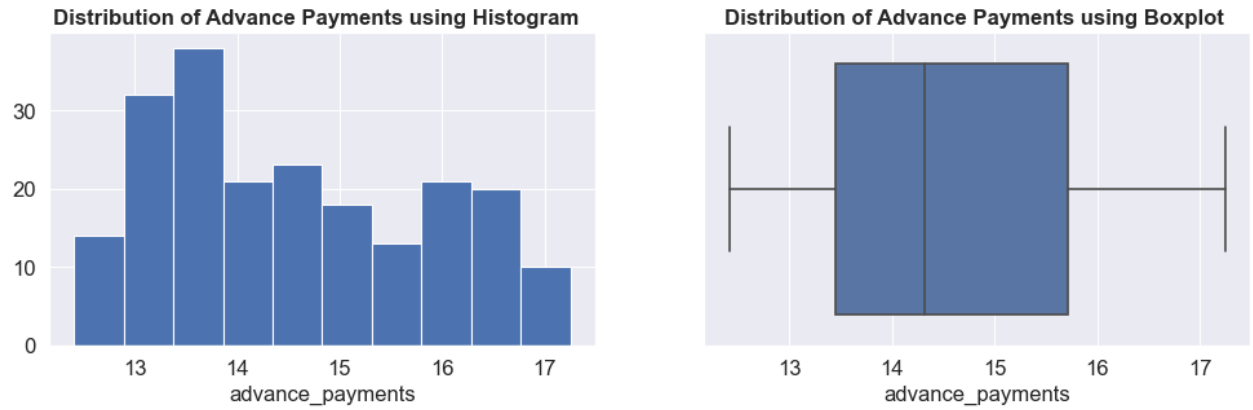

Fig 3: The box plot and the histogram showing the distribution of data of ''Advance Payments" variable

From the univariate analysis using histogram, we see that people who have done advance payment is between the range of 13500-14000 are the most, approximately 38, followed by the people who have spent in 13000-13500, approximately 32 and the least people spend in the amount range of 17000-17500, which is 10. From the bivariate analysis using boxplot, there are no presence of outliers. The second quartile(Q2) or median for the Advance_payments variable is about 14.32. The lower or first quartile(Q1) is about 13.45 and the upper or the third quartile(Q3) is about 15.715. The inter quartile range (IQR) for the above boxplot is 2.265

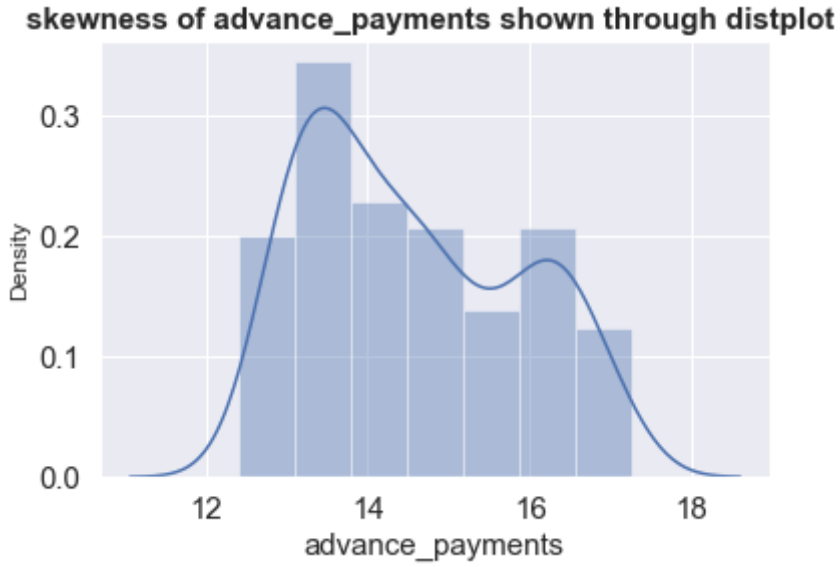

Fig 4: The distplot showing the skewness of data of 'Advance Payments' variable

From the above distplot , we can see that Advance Payments is positively skewed or right skewed, with median=14.32,mode=0 and 13.47,mean=14.559

3.Probability_of_full_payment

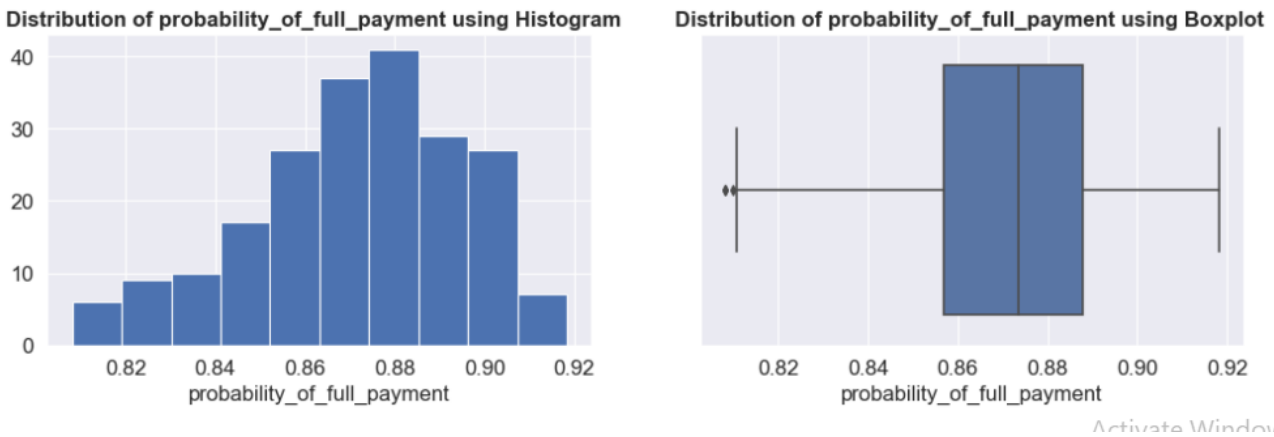

Fig 5: The box plot and the histogram showing the distribution of data of "Probability_of_full_payment" variable

From the univariate analysis using histogram, we see that people who have been in the Probability_of_full_payment range the maximum is between 0.87-0.88 (approximately 40), followed by the people whose Probability_of_full_payment is in the range of 0.862-0.87 , (approximately 38) and the least people in the probability range of 0.91-0.92 where the people are having the high probability of full payment(approximately 8). From the bivariate analysis using boxplot, there are no presence of outliers. The second quartile(Q2) or median for the Probability_of_full_payment variable is about 0.873. The lower or first quartile(Q1) is about 0.8569 and the upper or the third quartile(Q3) is about 0.887775. The inter quartile range (IQR) for the above boxplot is 0.030874

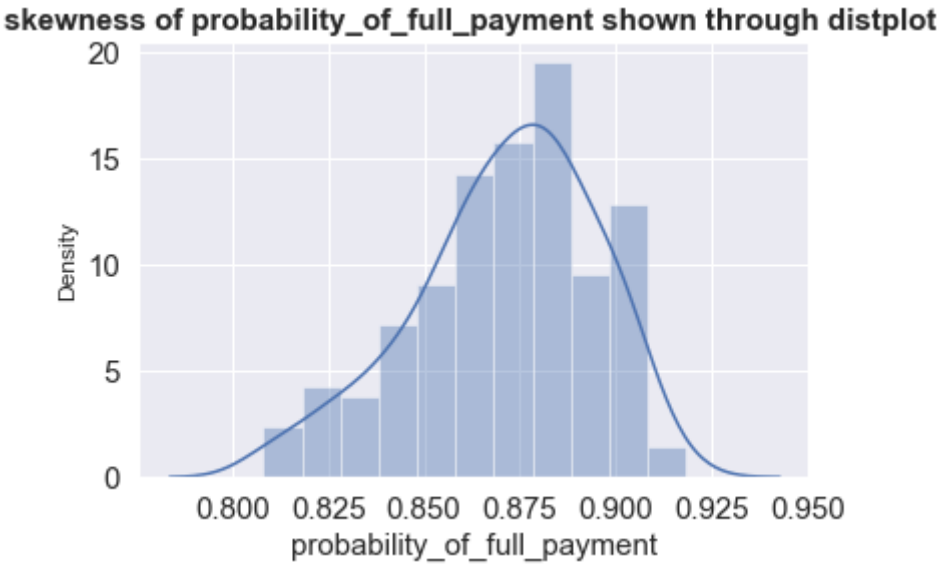

Fig 6: The distplot showing the skewness of data of 'Probability_of_full_payment' variable

From the above distplot , we can see that Probability_of_full_payment' is negatively skewed or left skewed, with median=0.87345,mode= 0 and 0.8823,mean=0.8709

4.Current_balance

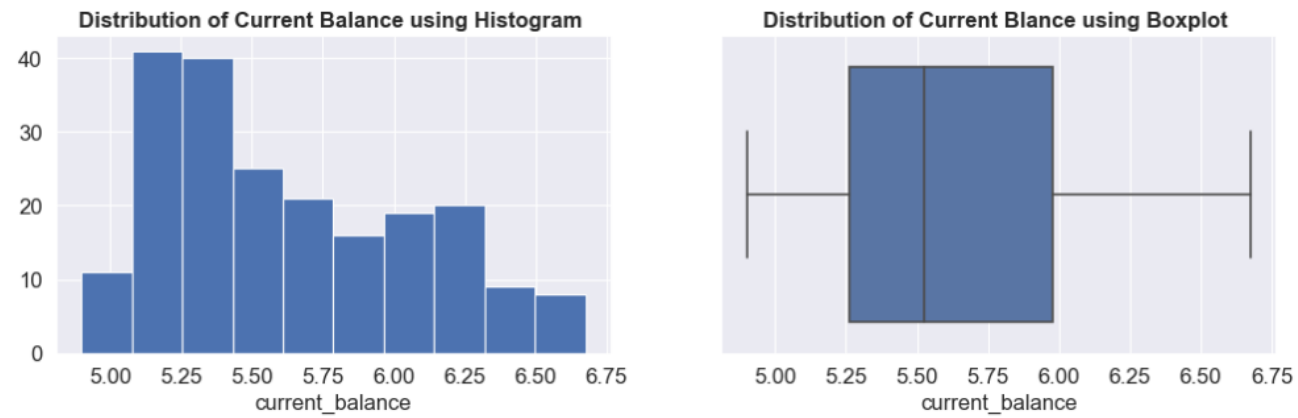

Fig 7: The box plot and the histogram showing the distribution of data of 'Current_balance' variable

From the univariate analysis using histogram, we see that the 'Current_balance' amount (in 1000s) left in the account for the people to do purchases is plotted along the x-axis. Maximum number of people have the current balance in their account in the range 5.05-5.25(approximately 41 people) The second most people have their balance in the range 5.25-5.45 (40 people). The least people have their current balance in the range 6.50-6.70(8 people). From the bivariate analysis using boxplot, there are no presence of outliers. The second quartile(Q2) or median for the Current balance variable is about 5.5235. The lower or first quartile(Q1) is about 5.262 and the upper or the third quartile(Q3) is about 5.979. The inter quartile range (IQR) for the above boxplot is 0.7175

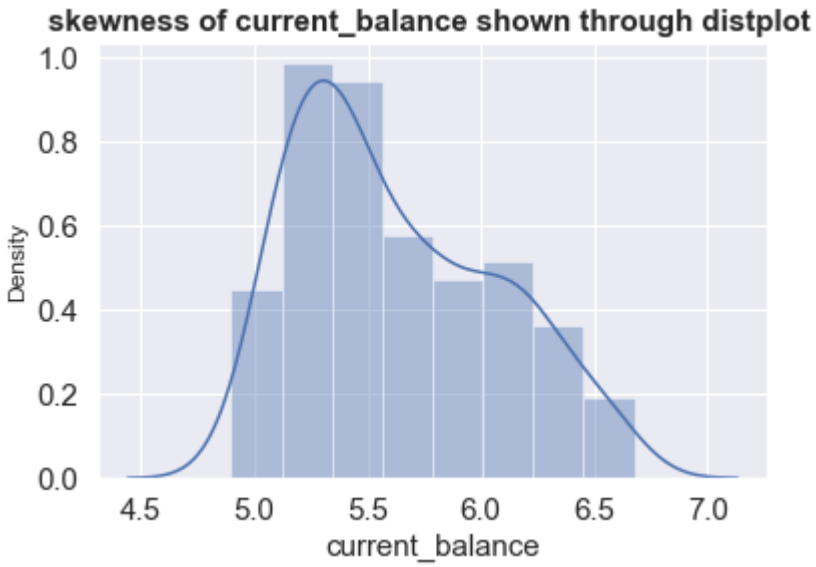

Fig 8: The distplot showing the skewness of data of 'Current_balance' variable

From the above distplot , we can see that Current_balance is positively skewed or right skewed, with median=5.5235,mode=0 and 5.236,mean=5.6285

5.Credit_limit

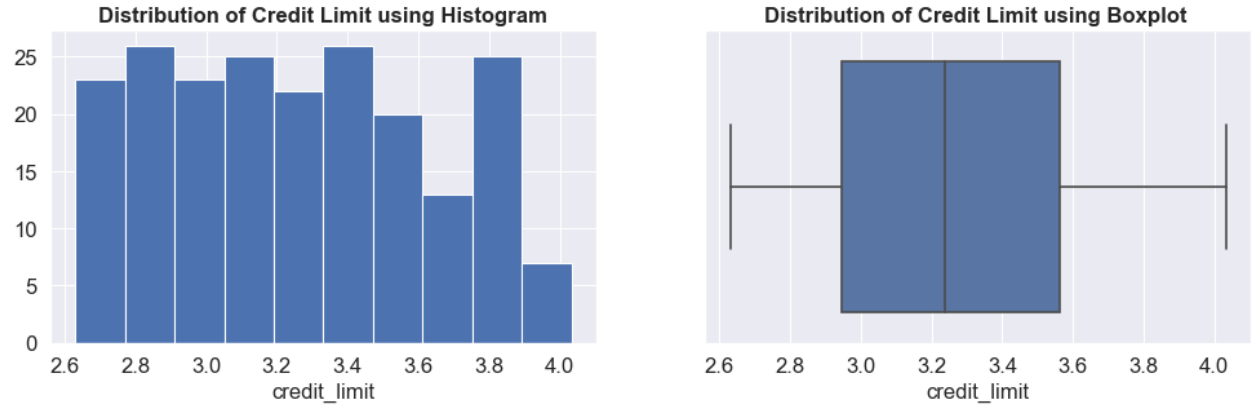

Fig 9: The box plot and the histogram showing the distribution of data of 'Credit_limit ' variable

From the univariate analysis using histogram, we see that the Credit_limit has been plotted in the x-axis. It shows the limit of amount in the people's credit card (in 10000s). Most people have the balance in the range of either 2.8-2.9 or 3.3-3.4(approximately 26 people).The second most highest balance people have in their account is in the range of either 3.1-3.2 or 3.8-3.9(approximately 25 people) From the bivariate analysis using boxplot, there are no presence of outliers. The second quartile(Q2) or median for the Credit limit variable is about 3.237 The lower or first quartile(Q1) is about 2.944 and the upper or the third quartile(Q3) is about 3.561 The inter quartile range (IQR) for the above boxplot is 0.617

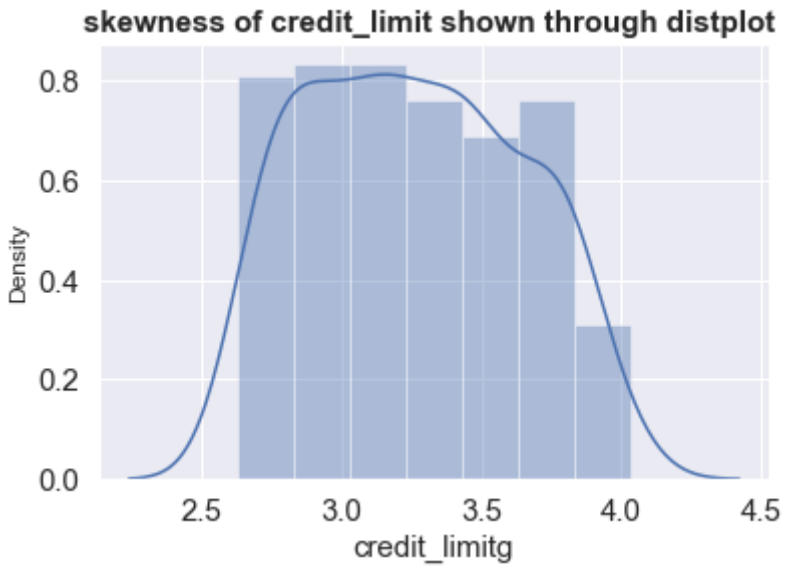

Fig 10: The distplot showing the skewness of data of 'Credit_limit' variable

From the above distplot , we can see that Credit_limit is slightly positively skewed or right skewed, with median=3.237,mode= 0 and 3.026,mean=3.2586

6.Min_payment_amt

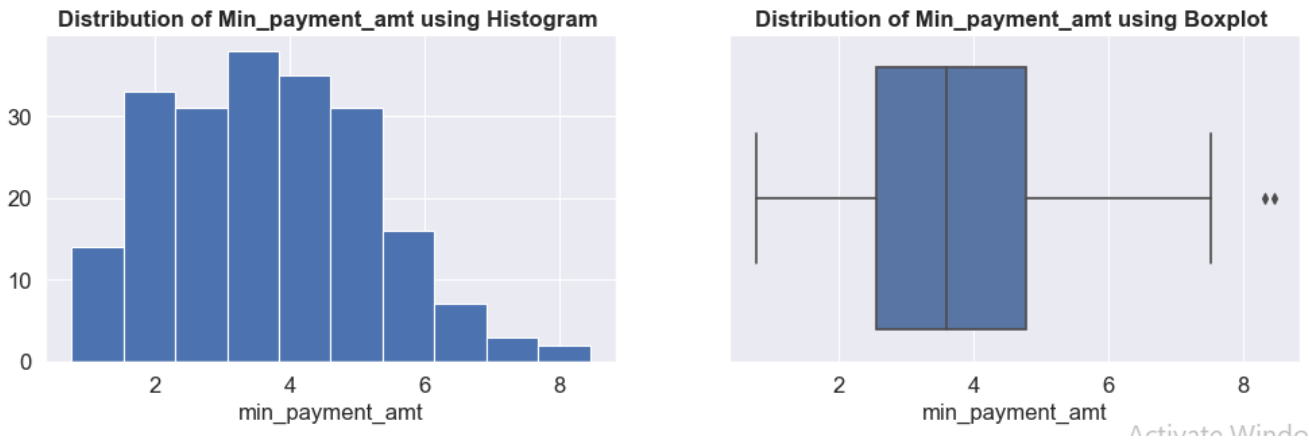

Fig 11: The box plot and the histogram showing the distribution of data of 'Min_payment_amt' variable

From the univariate analysis using histogram, we see that minimum amount paid by customers for monthly purchases are plotted in x-axis as Min_payment_amt'(in 100s).The maximum number of people's minimum payment is in the range of 3.5-3.9(approximately 38 people),followed by the payment in the range of 3.9-4.2(approximately 35 people).The least minimum payment are in the range of 7.5-8(approximately 3 people). From the bivariate analysis using boxplot, there are no presence of outliers. The second quartile(Q2) or median for the Min_payment_amt variable is about 3.599 The lower or first quartile(Q1) is about 2.5615 and the upper or the third quartile(Q3) is about 4.76875 The inter quartile range (IQR) for the above boxplot is 2.2072

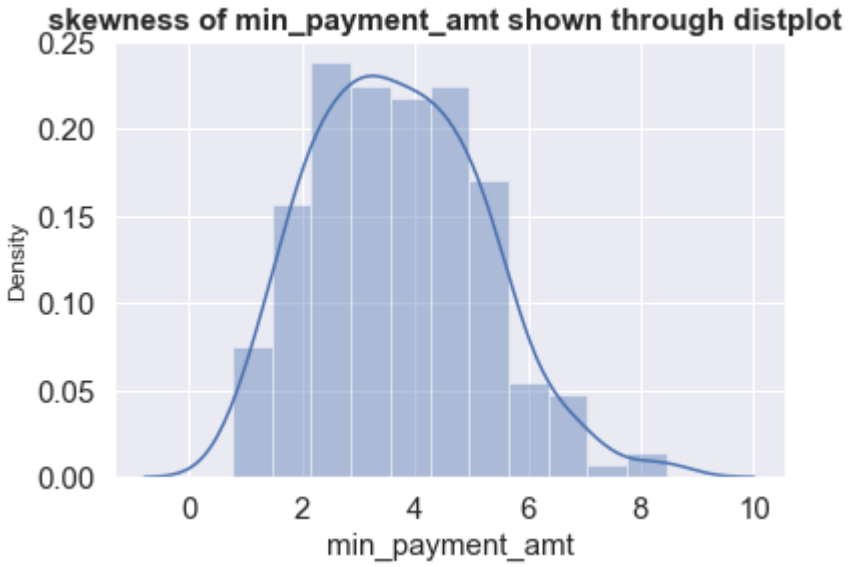

Fig 12: The distplot showing the skewness of data of 'Min_payment_amt' variable

From the above distplot, we can see that Min_payment_amt is positively or right skewed, median=3.599, mode=0 and 2.129, mean=3.7002

7.Max_spent_in_single_shopping

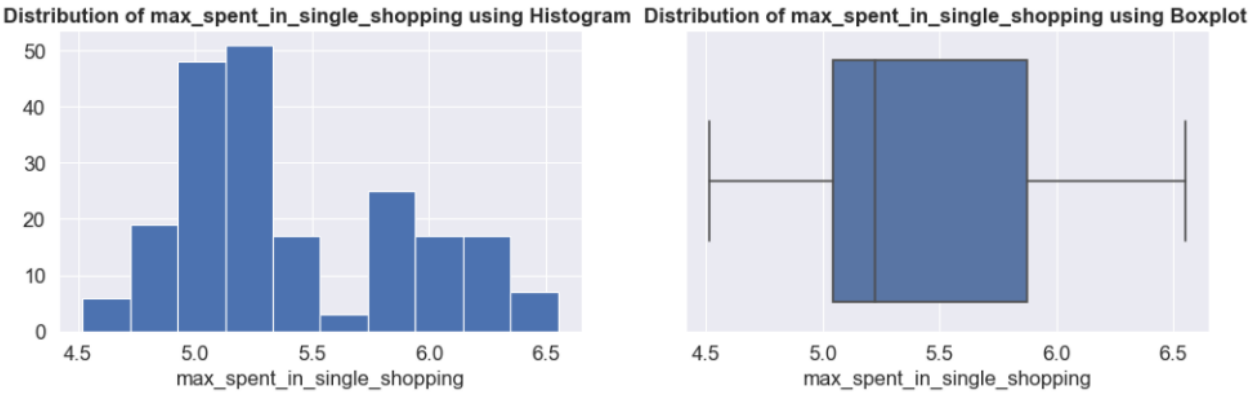

Fig 13: The box plot and the histogram showing the distribution of data of ''Max_spent_in_single_shopping' variable

From the univariate analysis using histogram, we see that the maximum amount spent in one shopping is plotted as Max_spent_in_single_shopping' (in 1000s) along the x-axis. Most people maximum spending is in the range of 5.1-5.4(approximately 50 people), followed by the range of 4.59-5.1(approximately 49 people). The least spending of the people is in the range of 5.5-5.7(approximately 4 people). From the bivariate analysis using boxplot, there are no presence of outliers. The second quartile(Q2) or median for the Max_spent_in_single_shopping variable is about 5.223. The lower or first quartile(Q1) is about 5.045 and the upper or the third quartile(Q3) is about 5.877. The inter quartile range (IQR) for the above boxplot is 0.8319

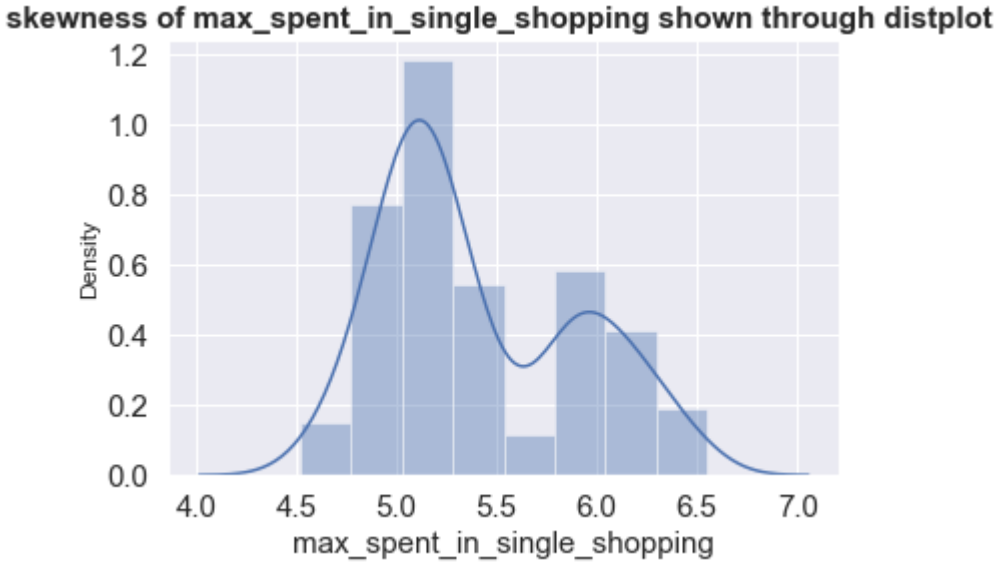

Fig 14: The distplot showing the skewness of data of 'Max_spent_in_single_shopping ' variable

From the above distplot , we can see that 'Max_spent_in_single_shopping is positively skewed or right skewed, with median=5.223,mode=0 and 5.001,mean=5.4080

i. Multivariate analysis

i.1.Heat Map

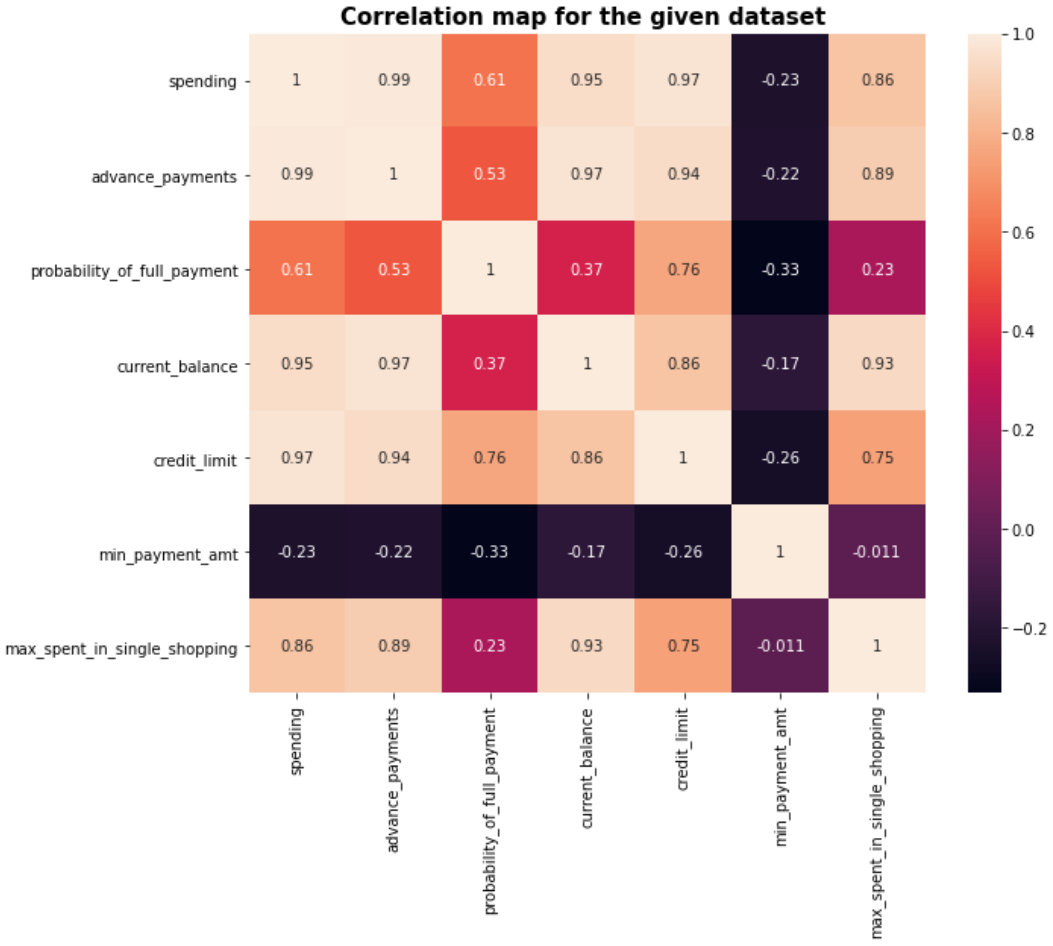

Fig 15: Heat map is portrayed for the multi variate analysis of the data.

From the pair plot it is seen that there exists high positive correlation between many components, among which advance_payments and spending has the highest positive correlation followed by advance_payments and current_balance while there are many components with negative correlations among which min_payment_amt and probability_of_full_payment has the least negative correlation.

j.2. Pair Plot

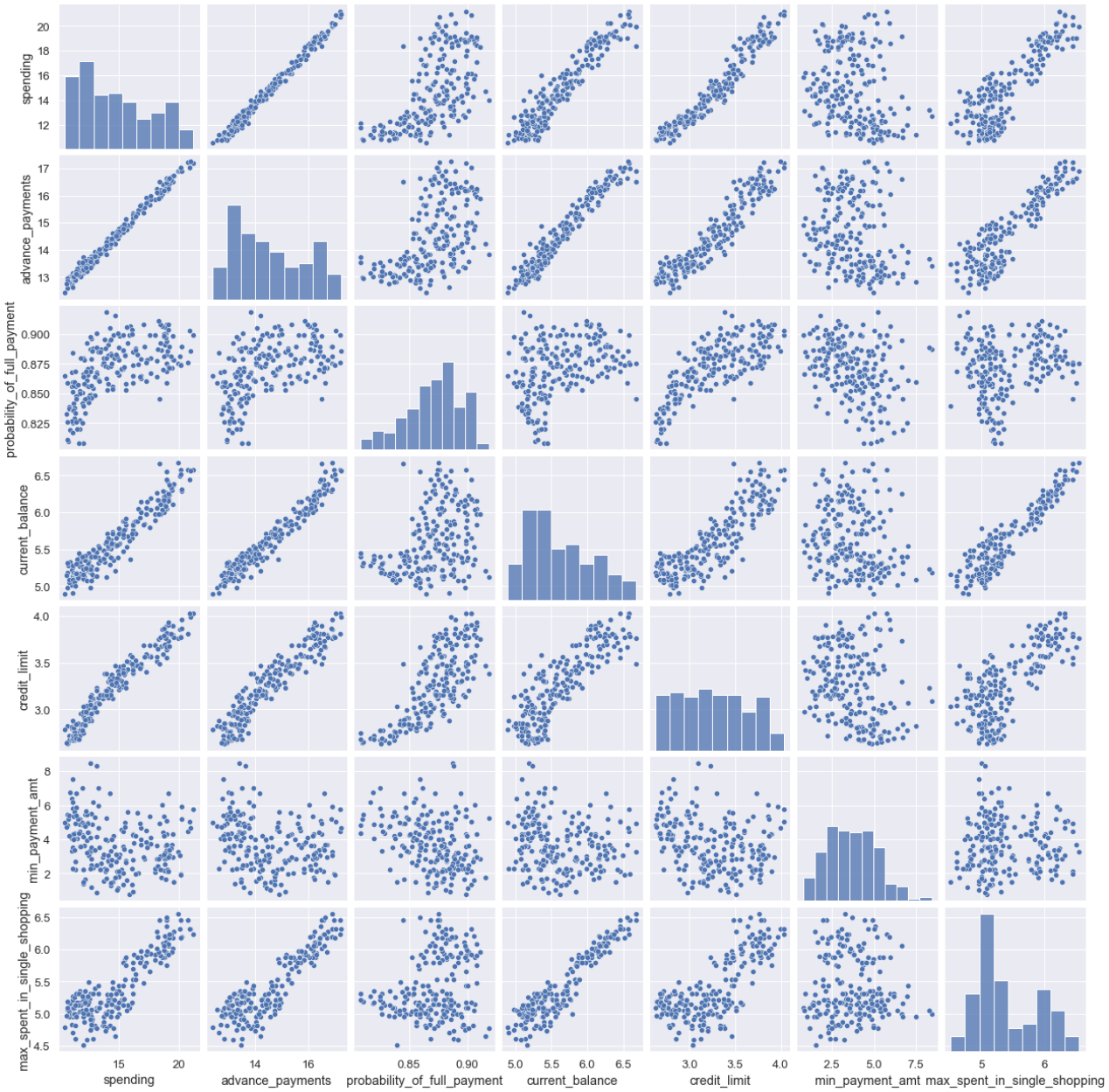

Fig 16: The pair plot showing the multivariate analysis

k. Skewness of the variables

```
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
```

Table 6: Skewness of all the variables

From the above table, we can see that probability_of_full_payment has negative correlation while max_spent_in_single_shopping has the highest skewness followed by current balance.

I. Outlier proportion

1.Spending

```
Range of values:  10.59
Minimum spending:  10.59
Maximum spending:  21.18
Mean value:  14.847523809523818
Median value:  14.355
Mode value:  0    11.23
1    14.11
2    15.38
dtype: float64
Standard deviation:  2.909699430687361
Null values:  False
spending - 1st Quartile (Q1) is:  12.27
spending - 3st Quartile (Q3) is:  17.305
Interquartile range (IQR) of spending is  5.035
Lower outliers in spending:  4.717499999999999
Upper outliers in spending:  24.8575
Number of outliers in spending upper :  0
Number of outliers in spending lower :  0
% of Outlier in spending upper:  0 %
% of Outlier in spending lower:  0 %
```

2. Advance_payments

```
Range of values:  4.84
Minimum advance_payments:  12.41
Maximum advance_payments:  17.25
Mean value:  14.559285714285727
Median value:  14.32
Mode value:  0    13.47
dtype: float64
Standard deviation:  1.305958726564022
Null values:  False
advance_payments - 1st Quartile (Q1) is:  13.45
advance_payments - 3st Quartile (Q3) is:  15.715
Interquartile range (IQR) of advance_payments is  2.2650000000000006
Lower outliers in advance_payments:  10.052499999999998
Upper outliers in advance_payments:  19.1125
Number of outliers in advance_payments upper :  0
Number of outliers in advance_payments lower :  0
% of Outlier in advance_payments upper:  0 %
% of Outlier in advance_payments lower:  0 %
```

3. Probability_of_full_payment

```
Range of values:  0.11019999999999996
Minimum probability_of_full_payment  0.8081
Maximum probability_of_full_payment:  0.9183
Mean value:  0.8709985714285714
Median value:  0.8734500000000001
Mode value:  0    0.8823
dtype: float64
Standard deviation:  0.0236294165838465
Null values:  False
probability_of_full_payment - 1st Quartile (Q1) is:  0.8569
probability_of_full_payment - 3st Quartile (Q3) is:  0.887775
Interquartile range (IQR) of probability_of_full_payment is  0.030874999999999986
Lower outliers in probability_of_full_payment:  0.8105875
Upper outliers in probability_of_full_payment:  0.9340875
Number of outliers in probability_of_full_payment upper :  0
Number of outliers in probability_of_full_payment lower :  3
% of Outlier in probability_of_full_payment upper:  0 %
% of Outlier in probability_of_full_payment lower:  1 %
```

4. Current_balance

```
Range of values:  1.7759999999999998
Minimum current_balance:  4.899
Maximum current_balance:  6.675
Mean value:  5.628533333333335
Median value:  5.5235
Mode value:  0    5.236
1    5.395
dtype: float64
Standard deviation:  0.44306347772644944
Null values:  False
current_balance - 1st Quartile (Q1) is:  5.26225
current_balance - 3st Quartile (Q3) is:  5.97975
Interquartile range (IQR) of current_balance is  0.7175000000000002
Lower outliers in current_balance:  4.186
Upper outliers in current_balance:  7.056000000000001
Number of outliers in current_balance upper :  0
Number of outliers in current_balance lower :  0
% of Outlier in current_balance upper:  0 %
% of Outlier in current_balance lower:  0 %
```

5.Credit_limit

```
Range of values:  1.4030000000000005
Minimum credit_limit:  2.63
Maximum credit_limit:  4.033
Mean value:  3.258604761904763
Median value:  3.237
Mode value:  0    3.026
dtype: float64
Standard deviation:  0.37771444490658734
Null values:  False
credit_limit - 1st Quartile (Q1) is:  2.944
credit_limit - 3st Quartile (Q3) is:  3.56175
Interquartile range (IQR) of credit_limit is  0.61775
Lower outliers in credit_limit:  2.017375
Upper outliers in credit_limit:  4.488375
Number of outliers in credit_limit upper :  0
Number of outliers in credit_limit lower :  0
% of Outlier in credit_limit upper:  0 %
% of Outlier in credit_limit lower:  0 %
```

6.Min_payment_amt

```
Range of values:  7.690899999999999
Minimum min_payment_amt:  0.7651
Maximum min_payment_amt:  8.456
Mean value:  3.7002009523809503
Median value:  3.599
Mode value:  0    2.129
1    2.221
2    2.700
dtype: float64
Standard deviation:  1.5035571308217792
Null values:  False
min_payment_amt - 1st Quartile (Q1) is:  2.5615
min_payment_amt - 3st Quartile (Q3) is:  4.76875
Interquartile range (IQR) of min_payment_amt is  2.2072499999999997
Lower outliers in min_payment_amt:  -0.7493749999999992
Upper outliers in min_payment_amt:  8.079625
Number of outliers in min_payment_amt upper :  2
Number of outliers in min_payment_amt lower :  0
% of Outlier in min_payment_amt upper:  1 %
% of Outlier in min_payment_amt lower:  0 %
```

7.Max_spent_in_single_shopping

```
Range of values:  2.0309999999999997
Minimum max_spent_in_single_shopping:  4.519
Maximum max_spent_in_single_shoppings:  6.55
Mean value:  5.408071428571429
Median value:  5.223000000000001
Mode value:  0    5.001
dtype: float64
Standard deviation:  0.49148049910240543
Null values:  False
max_spent_in_single_shopping - 1st Quartile (Q1) is:  5.045
max_spent_in_single_shopping - 3st Quartile (Q3) is:  5.877
Interquartile range (IQR) of max_spent_in_single_shopping is  0.8319999999999999
max_spent_in_single_shopping - 1st Quartile (Q1) is:  5.045
max_spent_in_single_shopping - 3st Quartile (Q3) is:  5.877
Interquartile range (IQR) of max_spent_in_single_shopping is  0.8319999999999999
Number of outliers in max_spent_in_single_shopping upper :  0
Number of outliers in max_spent_in_single_shopping lower :  0
% of Outlier in max_spent_in_single_shopping upper:  0 %
% of Outlier in max_spent_in_single_shopping lower:  0 %
```

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

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

Table 7: Summary of the dataset before scaling

From the above summary of the dataset, we see that the standard deviation of all the variables are not similar and their mean values are also quite different. Among the variables, Spending, current_balance, max_spent_in_single_shopping is given in the scale of 1000s while advance_payments and min_payment_amt are in the scale of 100s.The value of credit_limit is recorded in 10000s.These different scales will influence the clustering process and the k-means or hierarchical clustering technic will definitely get biased to the larger value.

Standard scaler uses simple standardisation or normalisation. Standard scaler uses the z-score computation. It computes the z-score of all the values in each and every column with mean 0 and std deviation 1. It basically centers all the values around 0.

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

Table 8: Head of the data set after performing the scaling.

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

Clustering means grouping observations so that the observations belonging in the same group are similar, whereas observations in different groups are dissimilar.

- In hierarchical clustering, records are sequentially grouped to create clusters, based on distances between records and distances between clusters.
- Hierarchical clustering also produces a useful graphical display of the clustering process and results, called a dendrogram.

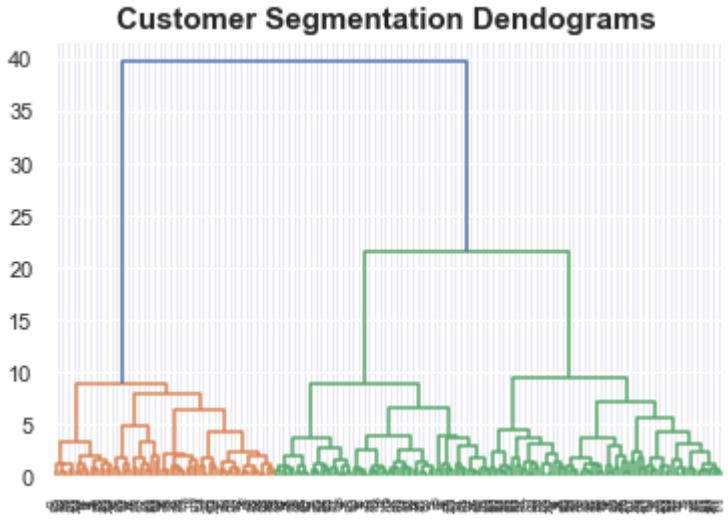

Fig 17: The above figure shows the customer segmentation dendrograms.

The above figure shown the dendrograms and the linkage (ward's link method) that has been drawn initially using hierarchical clustering.

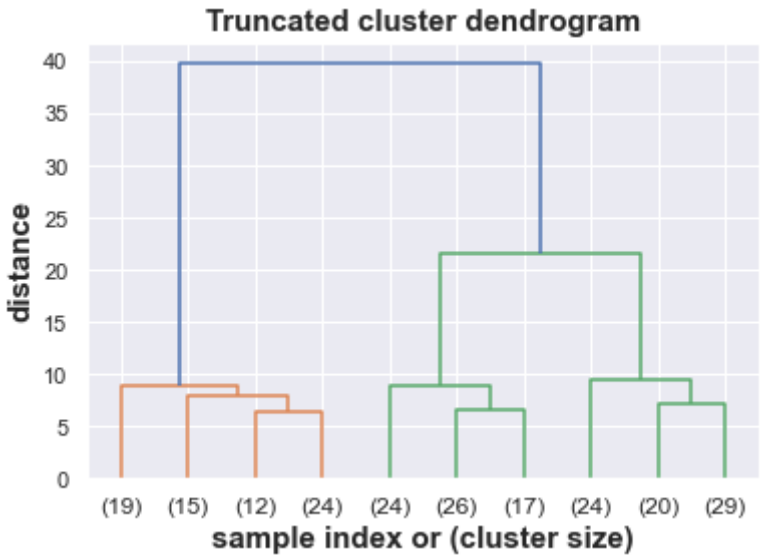

Fig 18: The above figure shows the truncated cluster dendrogram.

The above figure shows the more neatly formed diagram where the we have given the truncate_mode as lastp , where we have to asked to display only the last 10 linkages.

Every horizontal line in the above figure denotes a merge.Total number of observations in the red cluster =19+15+12+24=7. Total number of observations in the green cluster=24+26+17+24+20+29=140.Green cluster has maximum data(customers) into it

After getting the last 10 linkages, we have then used the fcluster technic to form the cluster where we formed the cluster using two criteria namely, maxclust and distance method.

- In maxclust method, we have given the number of maxclust=3 which means the maximum number of clusters we would like to form are 3 and so the data is grouped into either of the 3 clusters.
- In distance method, we have given the distance=20, where a horizontal line is drawn at the point= 20 in the above shown plot and then count how many vertical lines arise from it which in this case happens to be 3. Any desired number of clusters can be obtained by 'cutting' the dendrogram at the proper level.

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

The above chart shows the array of the clusters formed for each entry in the data frame.

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

Table 9: The head of the dataset after attaching the newly formed clusters (hierarchy clustering) along with the data frame.

If we calculate the mean of all the variables in cluster 1 individually, we get the centroid of the cluster 1. Similarly, we can find the centroid of all the 3 newly formed clusters.

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

From the above output, we can see that there are 70 variables in cluster 1, 67 variables in cluster 2 and 73 variables in cluster 3.

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

k-means clustering is the most used non-hierarchical clustering technique. It aims to partition n observations into k clusters in which each observation belongs to the cluster whose mean (centroid) is nearest to it, serving as a prototype of the cluster. It minimizes within-cluster variances (squared Euclidean distances).

K-means inertia for the dataset:

k_means.inertia_ for 1 cluster =1469.9999999999995
k_means.inertia_ for 2 clusters = 659.1717544870411
k_means.inertia_ for 3 clusters = 430.65897315130064
k_means.inertia_ for 4 clusters = 371.6531439995162

```
[1469.9999999999995,
 659.1717544870411,
 430.65897315130064,
 371.38509060801107,
 327.2127816566134,
 289.315995389595,
 262.98186570162267,
 241.8189465608603,
 223.91254221002728,
 206.3961218478669]
```

From the above wss plot we can conclude that the inertia after 3 is not decreasing significantly, so we can conclude with cluster number=3.

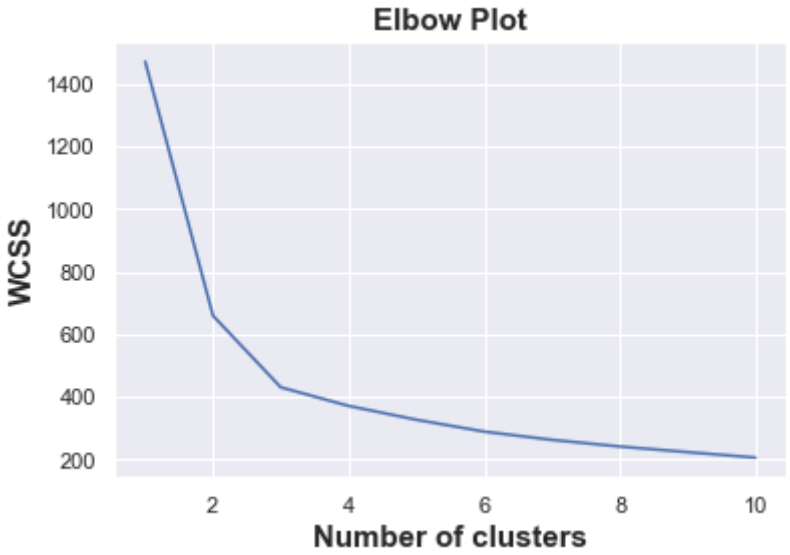

Fig 19: The above figure shows the elbow plot.

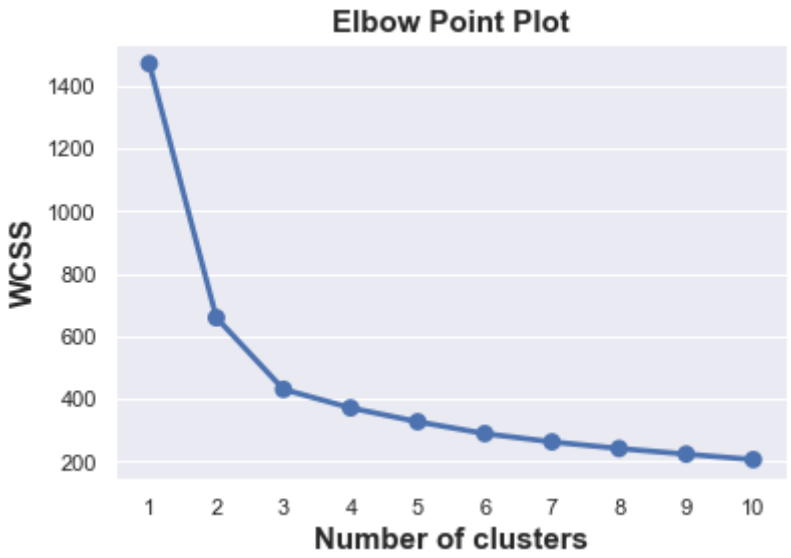

Fig 20: The above figure shows the elbow point plot.

Interpretation:

As the elbow curve doesn't drop significantly after the point 3, we can conclude that the optimum number of clusters =3.

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

From the above output, we can see that there are 71 variables in cluster 0, 72 variables in cluster 1 and 67 variables in cluster 2.

Silhouette Score

```
[0.46577247686580914,
 0.40072705527512986,
 0.3276547677266192,
 0.2827335237380383,
 0.2885980140325899,
 0.2819058746607507,
 0.26644334449887014,
 0.2583120167794957,
 0.25230419288400546]
```

The above array is an array containing the silhouette scores for different number of clusters. Silhouette Score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1. Here the silhouette score for cluster =3 is 0.4007270552751 which shows the clustering was highly efficient.

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

Table 10: The head of the dataset after attaching the newly formed clusters (k-means clustering) along with the data frame.

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

Cluster Group Profiles

Group 1: High Spending

Group 3: Medium Spending

Group 2: Low Spending

Group 1: High Spending Group

- Giving any reward or bonus points that might increase their purchases and can also t hey can be given attractive gifts for their high purchases.
- Since the maximum max_spent_in_single_shopping is high for this group, so they can be offered either discounts or other complementary products can be given to them fr ee of cost.
- These people can be offered attractive EMIs, as they are customers with good repay ment record.

Group 2: Low Spending Group

- Early payments should be rewarded with offer or special schemes to inhibit the early p ayment rate.
- Pursue to inhibit their spending habits by tying up with grocery stores, utilities (electri city, phone, gas, others)

Group 3: Medium Spending Group

- They are potential target customers who are paying bills and doing purchases and mai ntaining comparatively good credit score. Most of the people of this group seems to be middle class people. So, by lowering down interest rate, we can increase their spendin g habits.
- More transparency needs to felt with the transactions, so the customers increase the nu mber of transactions

INSURANCE DATA

CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

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.

INTRODUCTION

The purpose of this whole exercise is to do the exploratory data analysis and then implement different classification techniques such as CART model, Artificial Neural Network and Random Forest model after splitting that data into train and test data. This assignment should help the student in knowing the technique of selecting the best parameters using GridSearchCV technique which can be used to build the models with maximum accuracy and without any loose of data. The comparison of accuracy between the train and test data will clearly portray the efficiency of the model that has been built.

DATA DESCRIPTION

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

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

a. Sample 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 1: Dataset Sample

The above table shows the head of the given dataset, i.e. the first five entries to ensure that the dataset has been loaded without any issues.

a. Shape of the dataset

```
df2.shape
```

```
(3000, 10)
```

There are total 3000 rows and 10 columns in the dataset.

b. Checking for the missing values in the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Age           3000 non-null   int64
 1   Agency_Code   3000 non-null   object
 2   Type          3000 non-null   object
 3   Claimed       3000 non-null   object
 4   Commision     3000 non-null   float64
 5   Channel       3000 non-null   object
 6   Duration      3000 non-null   int64
 7   Sales         3000 non-null   float64
 8   Product Name  3000 non-null   object
 9   Destination   3000 non-null   object
dtypes: float64(2), int64(2), object(6)
```

Table 2. Checking for the missing values in the dataset

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

c. Checking for the null values in the given dataset.

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

Table 3. Checking for the null values in the dataset

From the above table, we can see that there are no null values in the given data set.

d. Checking for the datatypes of variables present in the dataset.

```
Age              int64
Agency_Code     object
Type            object
Claimed         object
Commision      float64
Channel         object
Duration         int64
Sales          float64
Product Name    object
Destination     object
```

Table 4. Checking the data type of variables in the data set

Out of 10, there are 2 variables of int data type,6 variables of object data type and two variables of float data type.

Checking for the number of duplicated values

```
df2.duplicated().sum()
```

```
139
```

Here, we can see that there are 139 duplicated values in the given dataset. Even though there are 139 duplicated values, since there aren't any unique identification ID for the customers, there are chances that these values can be of different customers. So, I am not dropping it.

e. Checking for the summary of the given dataset.

	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

Table 5: The above table shows the summary statistics of the given dataset.

h. Graphical Representation of univariate and bivariate analysis for continuous columns

h.1. Age

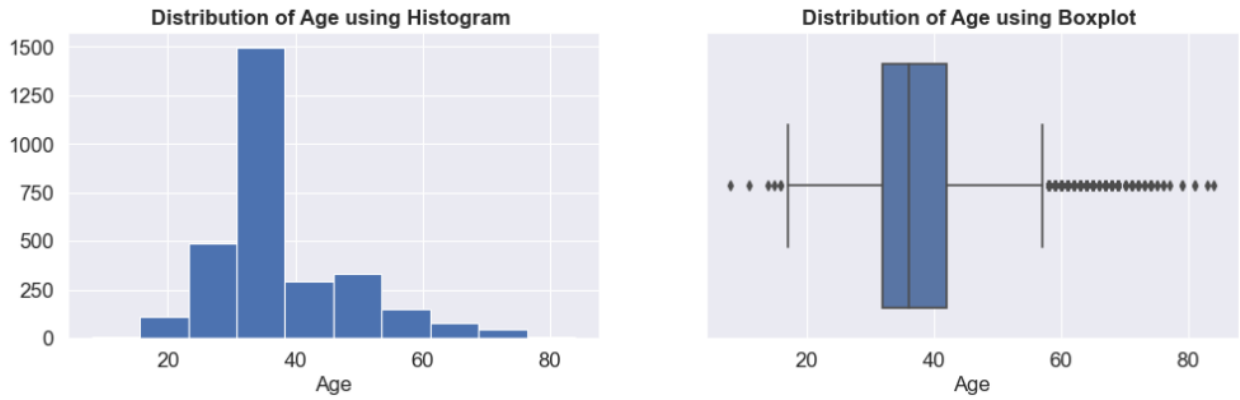

Fig 1: The box plot and the histogram showing the distribution of data of 'Age' variable

From the univariate analysis using histogram, we see that people who have insured comes in the age group of 30-40 the most, approximately 1500, followed by the age group of 20-30, approximately 500 and the least being the 70-80 age group which is less than 50. From the bivariate analysis using boxplot, there are presence of outliers. The second quartile(Q2) or median for the Age variable is about 36. The lower or first quartile(Q1) is about 32 and the upper or the third quartile(Q3) is about 42. The inter quartile range (IQR) for the above boxplot is 10.

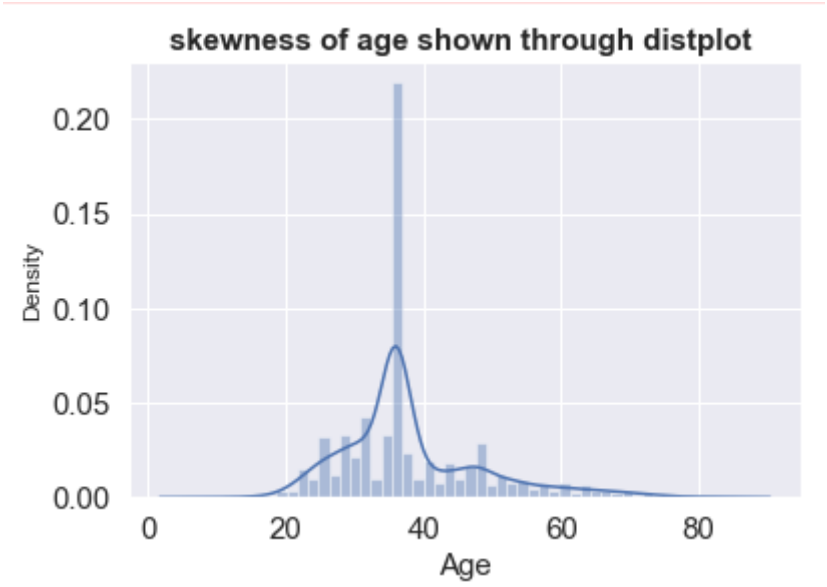

Fig 2: The distplot showing the skewness of data of 'Age' variable

From the above distplot, we can see that Age is positively skewed or right skewed, with median=36, mode=0 and 36 ; mean= 38.091. From this it is clear that Age variable is bimodal with two mode values.

h.2. Commision

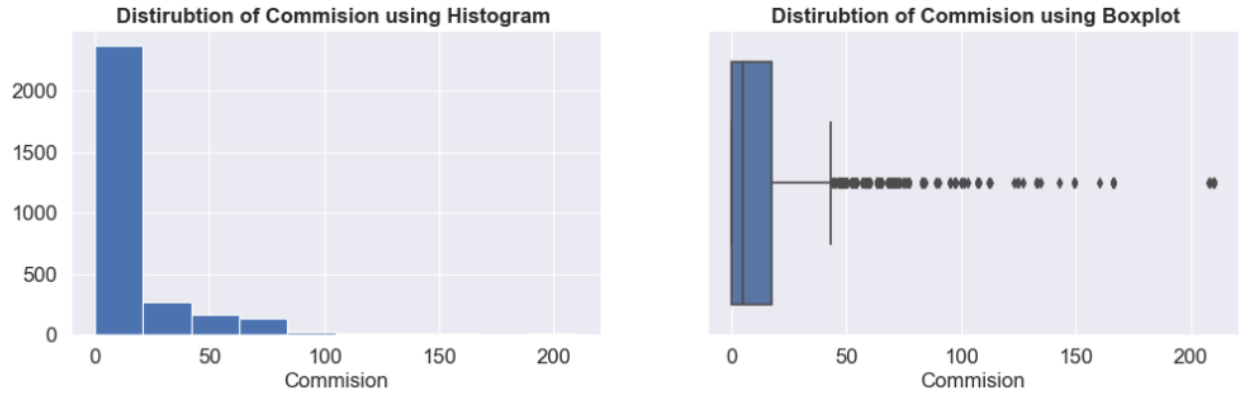

Fig 3: The box plot and the histogram showing the distribution of data of 'Commision' variable

From the univariate analysis using histogram, we see that the commission received by the tour insurance firm in the range 0-25(%) are the most, approximately 2400, followed by the those who received in the range 25-50(%), approximately 250 and the least being the 80-100(%) which is less than 10. From the bivariate analysis using boxplot, there are presence of outliers. The second quartile(Q2) or median for the Commision variable is about 4.63. The lower or first quartile(Q1) is 0 and the upper or the third quartile(Q3) is about 17.235. The inter quartile range (IQR) for the above boxplot is 17.235.

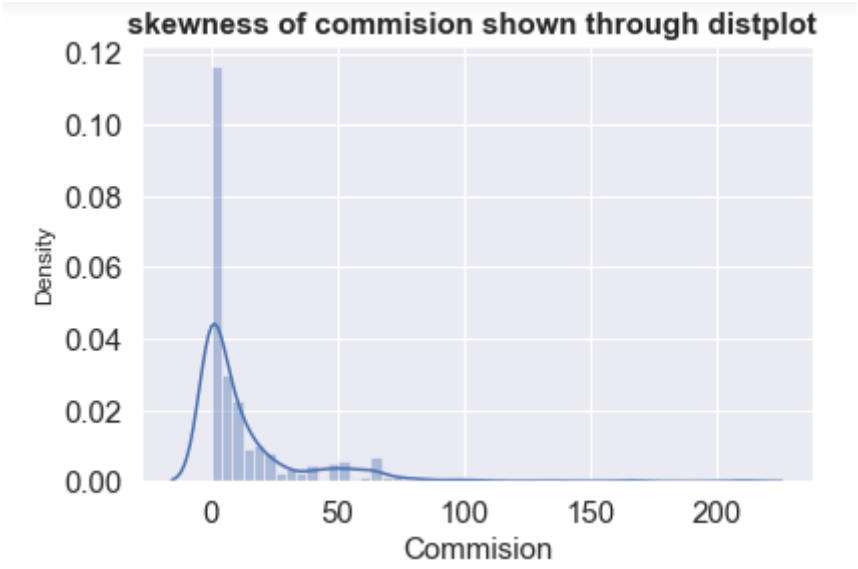

Fig 4: The distplot showing the skewness of data of 'Commision' variable

From the above distplot, we can see that commision is positively skewed or right skewed, with median=4.63, mode=0 and mean=14.529.

h.3. Duration

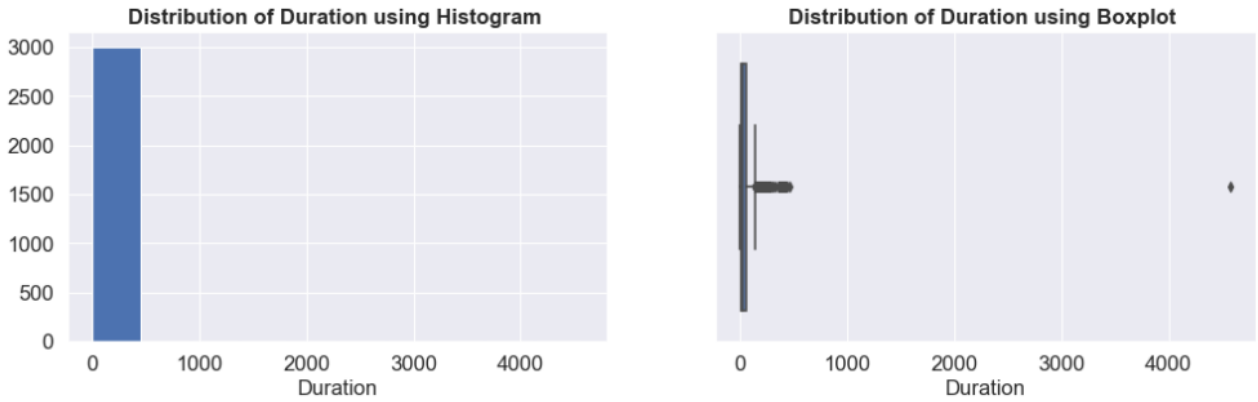

Fig 5: The box plot and the histogram showing the distribution of data of 'Duration' variable

From the univariate analysis using histogram, we see that the duration of the tour had just been 0-500 days. From the bivariate analysis using boxplot, there is a presence of outliers.

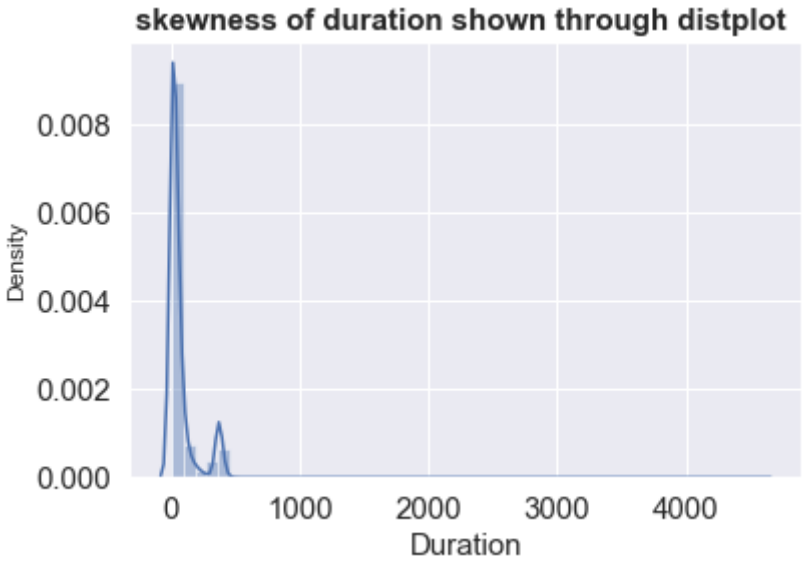

Fig 6: The distplot showing the skewness of data of 'Duration' variable

From the above distplot , we can see that Duration is positively skewed or right skewed, with median=26.5, mode=0 and 8; mean=70.001. Here also we can see that Duration is bimodal with two modes.

h.4. Sales

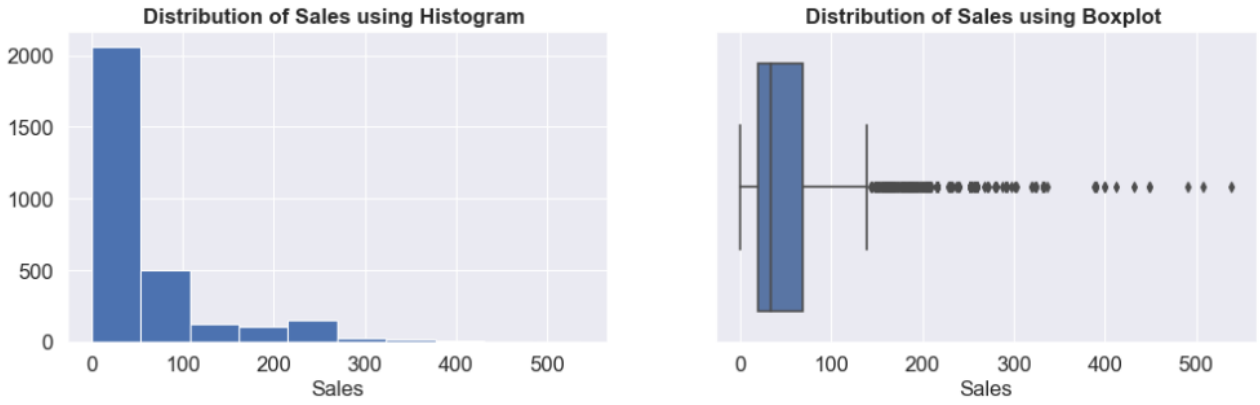

Fig 7: The box plot and the histogram showing the distribution of data of 'Sales' variable

From the univariate analysis using histogram, we see Amount worth of sales per customer in procuring tour insurance policies in rupees (in 100's) is the most for 0-5000 range, approximately 2000 customers have been charged amount in this range, followed by range of 5001-10000, approximately 500 customers have been charged and the least being the 32000-38000 range which is less than 10. From the bivariate analysis using boxplot, there are presence of outliers. The second quartile(Q2) or median for the Sales variable is about 33. The lower or first quartile(Q1) is about -53.5 and the upper or the third quartile(Q3) is about 142.5. The inter quartile range (IQR) for the above boxplot is 49.

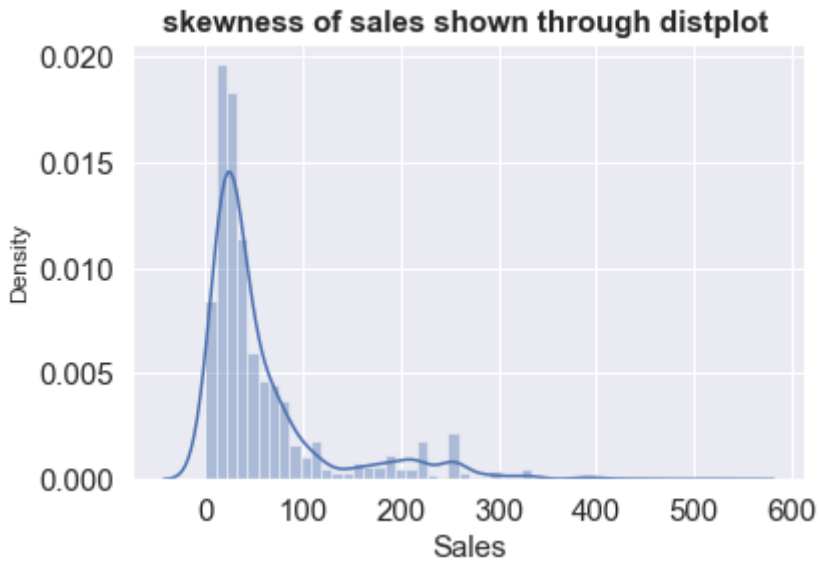

Fig 8: The distplot showing the skewness of data of 'Sales' variable

From the above distplot, we can see that sales is positively skewed or right skewed, with median=33, mode=0 and 20 and mean=60.2499; Here also we can see that sales is bimodal with two modes.

i. Graphical Representation of univariate and bivariate analysis for categorical columns

i.1. Type

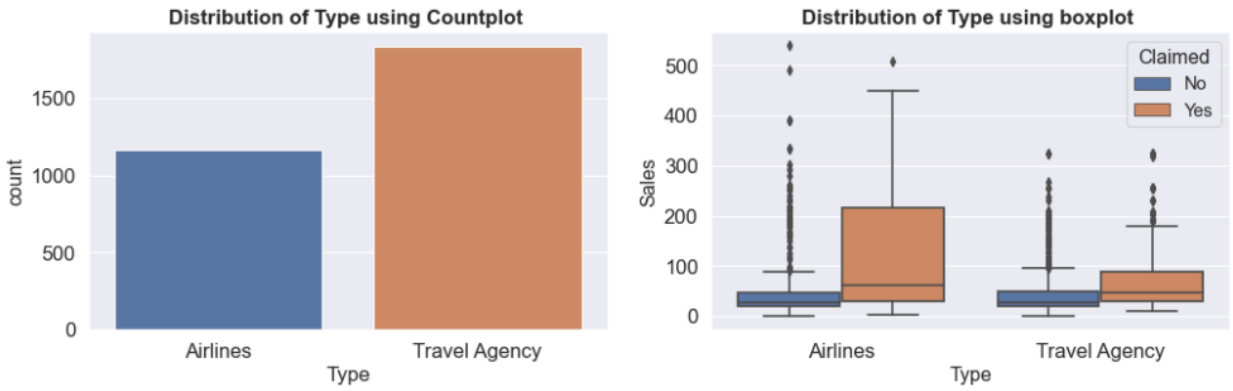

Fig 9: The box plot and the count plot showing the distribution of data of 'Type' variable

From the univariate analysis using Count plot, we see that type of tour insurance firms' people have used the most is the Travel Agency accounting to 1837 while the people who prefer Airlines tour insurance firms is a little more than 1163. From the bivariate analysis using boxplot, there is presence of outliers in all the four box plots. The median of people who have got the claim using the Travel Agency is the highest among all. We can also notice that the average sales of the Airlines for the customers who got their claim is the maximum,

slightly more than 200 while for travel agency it is just 100. The average sales of for both the tour insurance firms where the customers who didn't get their claim is same.

i.2.Agency code

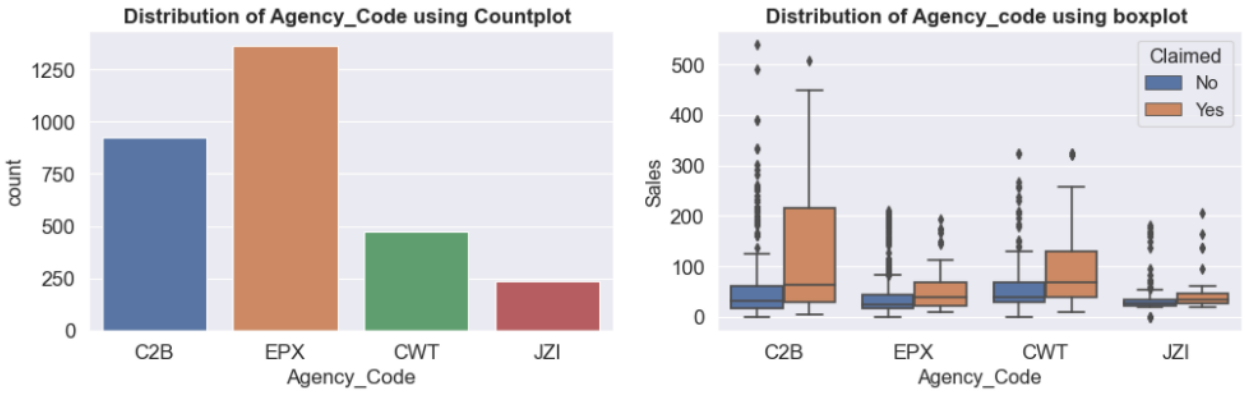

Fig 10: The box plot and the count plot showing the distribution of data of 'Agency_Code' variable

From the univariate analysis using Count plot, we see that tour firm with the code EPX had maximum number of customers, accounting to 1365, after which comes that C2B Agency accounting to 9264 and the minimum number of people have used JZI Agency, just 239 customers. From the bivariate analysis using boxplot, there is presence of outliers in all the eight box plots. The median of people who have used the agency code C2B and got their claim is similar to the median of the people who have used the agency code JZI and got their claim. Also, both these categories has only one outlier.

i.3. Channel

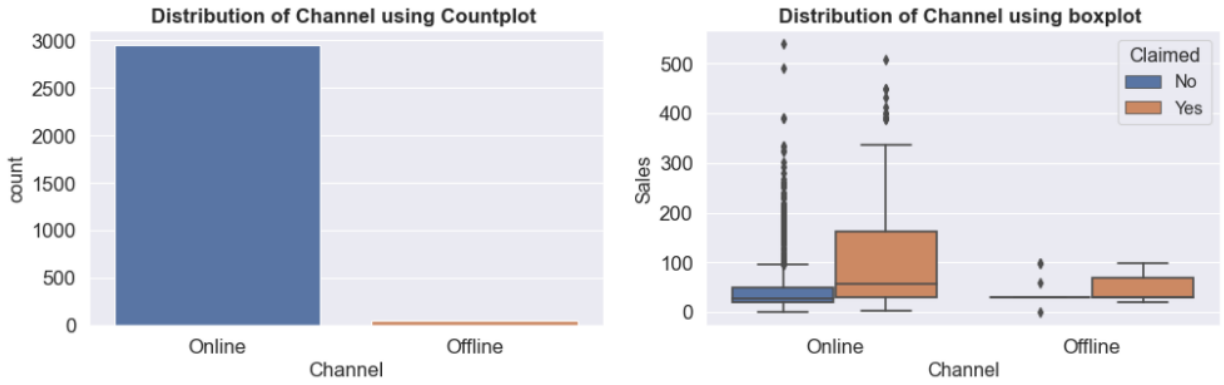

Fig 11: The box plot and the count plot showing the distribution of data of 'Channel' variable

From the univariate analysis using Count plot, we see that maximum number of people have preferred online channel (accounting to 2954) and the people preferring offline being very

negligible (accounting to 46). From the bivariate analysis using boxplot, there is presence of outliers. Among the two channels, it is noticed that only three people who used offline channel have not got their claim. It is also noticed that median of people who got claim through online medium is almost similar to the upper quartile of the people who couldn't get their claim who have used online medium.

i.4. Destination

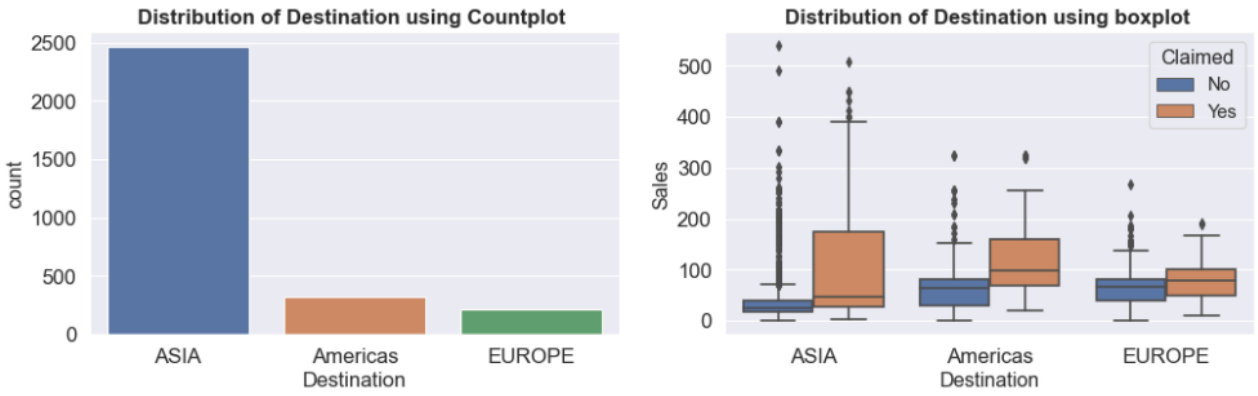

Fig 12: The box plot and the count plot showing the distribution of data of 'Destination' variable

From the univariate analysis using Count plot, we see that Destination that people have visited the most is Asia (accounting for almost 2465 customers) after which comes the Americas (accounting to almost 320 customers) and the least being the Europe (215 customers). From the bivariate analysis using boxplot, there is presence of outliers in all the three destinations. Across the three different destinations, the Americas got a slightly higher mean when compared to the rest two.

i.5.Product Name

Fig 13: The box plot and the count plot showing the distribution of data of 'Product Name' variable

From the univariate analysis using Count plot, we see that tour insurance product that people have used the most is the Customised Plan (1136 customers) after which comes the Cancellation plan (678 customers) and the least being the Gold plan (109 customers). From the bivariate analysis using boxplot, we can see that Customised Plan, Cancellation plan and Bronze plan have outliers while there is no presence of outliers in Silver or Gold plan. Across the five different plans, the number of customers who got their claim are greater than those who didn't get their claim. The Gold plan has maximum sales of insurance policies followed by Silver plan while the least sales of insurance are for Cancellation plan. The median of both the gold and silver plan are almost the same.

j. Multivariate analysis

j.1.Heat Map

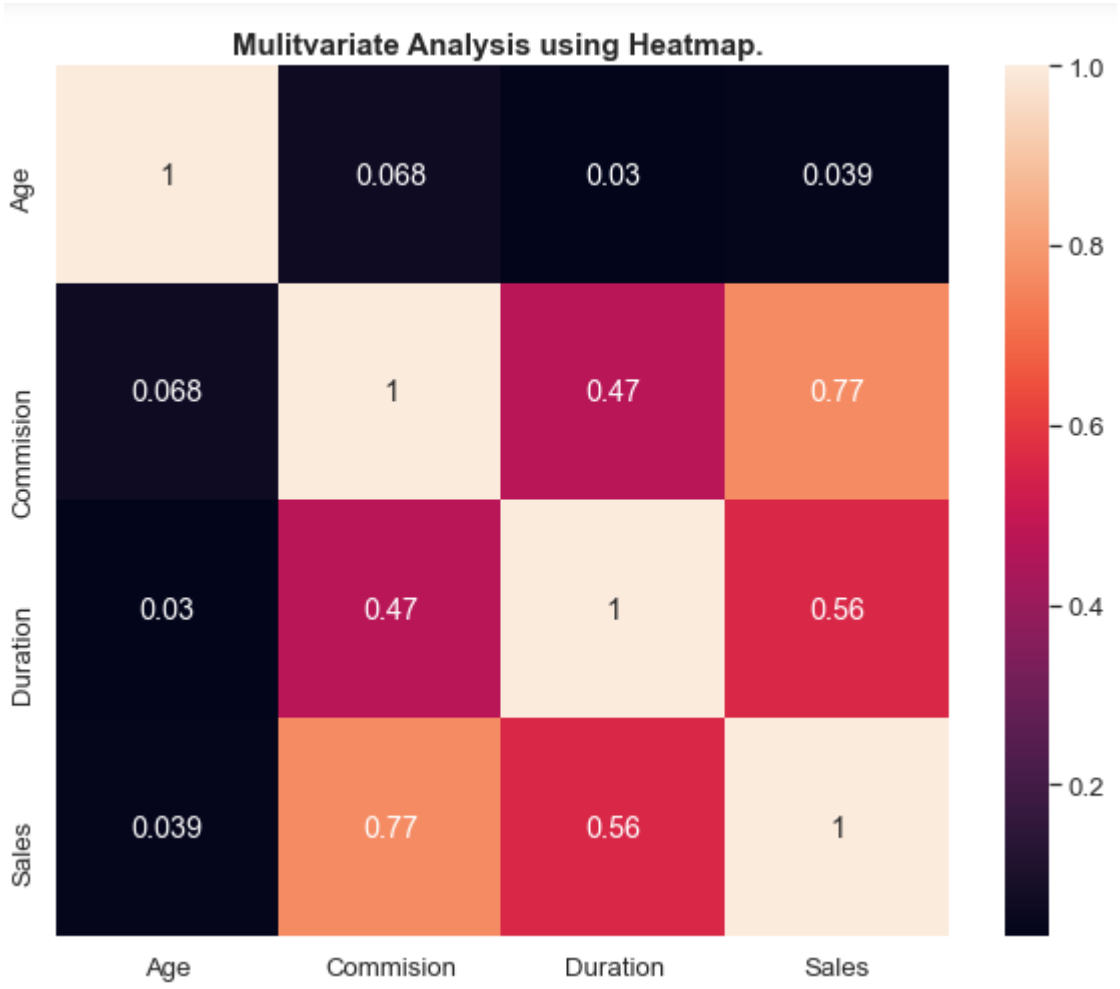

Fig 14: Heat map is portrayed for the multi variate analysis of the data.

From the pair plot it is seen that there exists high correlation between Sales and Commision while the correlation between Duration and Age is the least.

j.2. Pair Plot

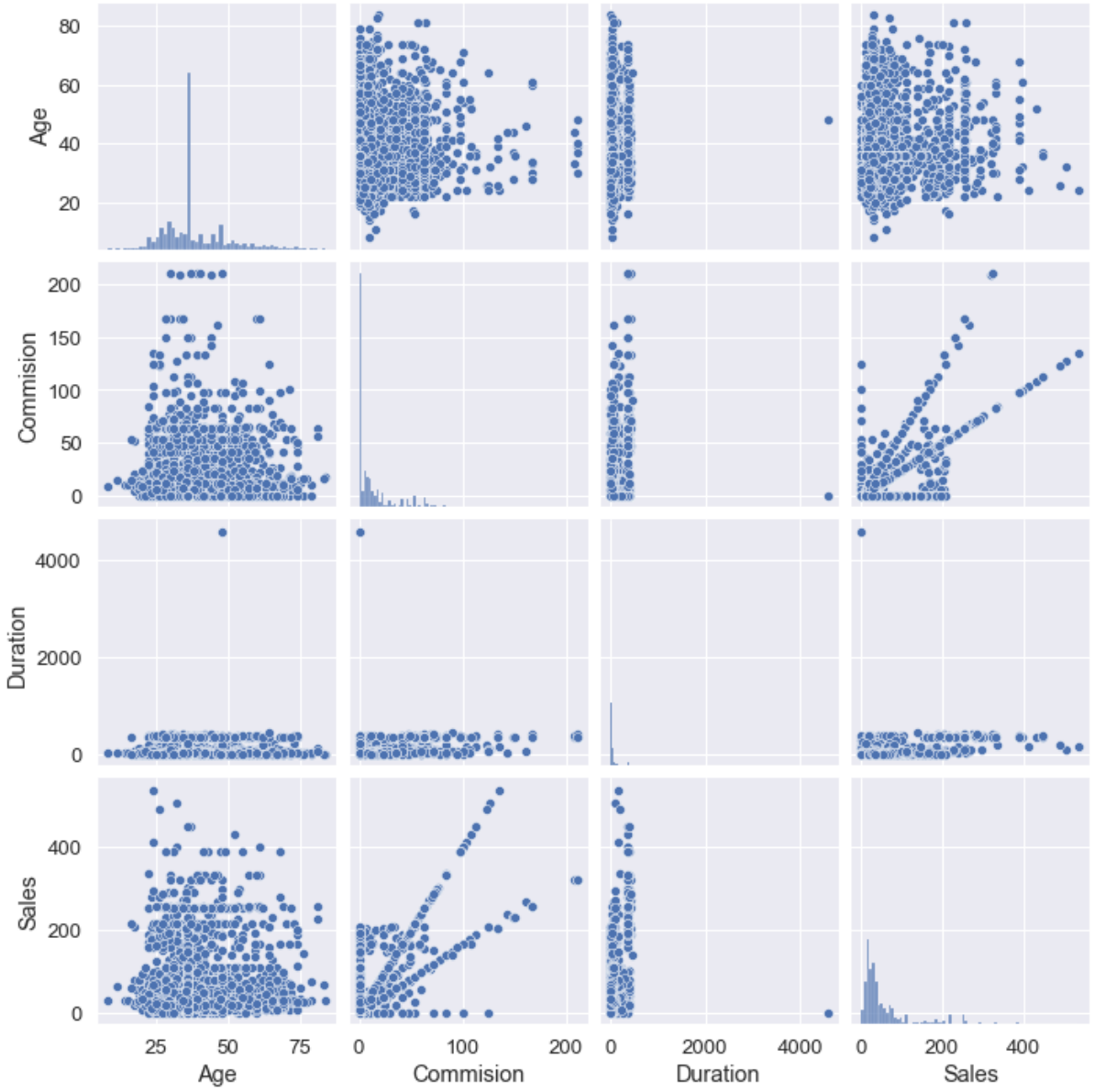

Fig 15: The pair plot showing the multivariate analysis

k. Skewness of the variables

Age	1.149713
Agency_Code	-0.155126
Type	-0.461352
Claimed	0.832185
Commision	3.148858
Channel	-7.892734
Duration	13.784681
Sales	2.381148
Product Name	0.432670
Destination	2.188556

Table 6: The above table shows the skewness of various variables in the given dataset.

From the above given table, we can conclude that the variables Agency_Code, Type and Channel are negatively skewed while the rest of the variables are positively skewed. Out of the positively skewed variables, Duration has the highest skewness followed by Commision, then by Sales. Among the non-object variables, it is Duration that is having high positive skewness while it is Age that has the least positive skewness. The variables with object data type has been converted into integer data type before calculating the skewness.

I. Outlier proportion

1. Age

```
Range of values:  76
Minimum Age:  8
Maximum Age:  84
Mean value:  38.091
Median value:  36.0
Mode value:  0    36
dtype: int64
Standard deviation:  10.463518245377944
Null values:  False
Age - 1st Quartile (Q1) is:  32.0
Age - 3st Quartile (Q3) is:  42.0
Interquartile range (IQR) of Age is  10.0
Lower outliers in Age:  17.0
Upper outliers in Age:  57.0
Number of outliers in Age upper :  198
Number of outliers in Age lower :  6
% of Outlier in Age upper:  7 %
% of Outlier in Age lower:  0 %
```

2.Commision

```
Range of values:  210.21
Minimum Commision:  0.0
Maximum Commision:  210.21
Mean value:  14.529203333333266
Median value:  4.63
Mode value:  4.63
Standard deviation:  25.48145450662553
Null values:  False
Commision - 1st Quartile (Q1) is:  0.0
Commision - 3st Quartile (Q3) is:  17.235
Interquartile range (IQR) of Commision is  17.235
Lower outliers in Commision:  -25.8525
Upper outliers in Commision:  43.0875
Number of outliers in Commision upper :  362
Number of outliers in Commision lower :  0
% of Outlier in Commision upper:  12 %
% of Outlier in Commision lower:  0 %
```

3.Duration

```
Range of values:  4581
Minimum Duration:  -1
Maximum Duration:  4580
Mean value:  70.00133333333333
Median value:  26.5
Mode value:  26.5
Standard deviation:  134.05331313253495
Null values:  False
Duration - 1st Quartile (Q1) is:  11.0
Duration - 3st Quartile (Q3) is:  63.0
Interquartile range (IQR) of Duration is  52.0
Lower outliers in Duration:  -67.0
Upper outliers in Duration:  141.0
Number of outliers in Duration upper :  382
Number of outliers in Duration lower :  0
% of Outlier in Duration upper:  13 %
% of Outlier in Duration lower:  0 %
```

4.Sales

```
Range of values:  539.0
Minimum Sales:  0.0
Maximum Sales:  539.0
Mean value:  60.24991333333344
Median value:  33.0
Mode value:  33.0
Standard deviation:  70.73395353143047
Null values:  False
Sales - 1st Quartile (Q1) is:  20.0
Sales - 3st Quartile (Q3) is:  69.0
Interquartile range (IQR) of Sales is  49.0
Lower outliers in Sales:  -53.5
Upper outliers in Sales:  142.5
Number of outliers in Sales upper :  353
Number of outliers in Sales lower :  0
% of Outlier in Sales upper:  12 %
% of Outlier in Sales lower:  0 %
```

Q2.2. 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 for independent attributes
from sklearn.model_selection import train_test_split

X_train, X_test, train_labels, test_labels = train_test_split(X, y, test_size=.30, random_state=1)
```

The above code shows that the given data has been split into the training and testing data with a ratio of 70:30 respectively.

1.CART

```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'max_depth': [5, 6, 7, 8, 9, 10],
                         'min_samples_leaf': [3, 4, 5, 6, 7],
                         'min_samples_split': [115, 116, 117, 118, 119, 120]})
```

The above is the list of parameters that has been loaded into the grid search CV

```
{'max_depth': 7, 'min_samples_leaf': 3, 'min_samples_split': 115}
```

The above set of parameters were considered the best for this modelling.

- Selection of different parameters:

a) Max_depth: The maximum depth is the depth till which you allow the tree to grow to. The deeper you allow, the more complex your model will become. If you increase max_depth, training error will also increase. So, it is always recommended to keep the maximum depth between 7 and 10. Here I have given an array of values from 5 to 10 out of which the gridsearch CV chose max_depth=7 better for max_depth among the array of inputted values.

b) Min_samples_leaf: Usually for training we take 70% of the given data. In min sample leaf, we give 1-3% of the total number of the remaining 70% records.
c) Min_samples_split: By min_sample_split, we mean to say that minimum samples that has to be present in the leaf before its next split.In min sample split, we give at least 3 times the size of min sample leaves.

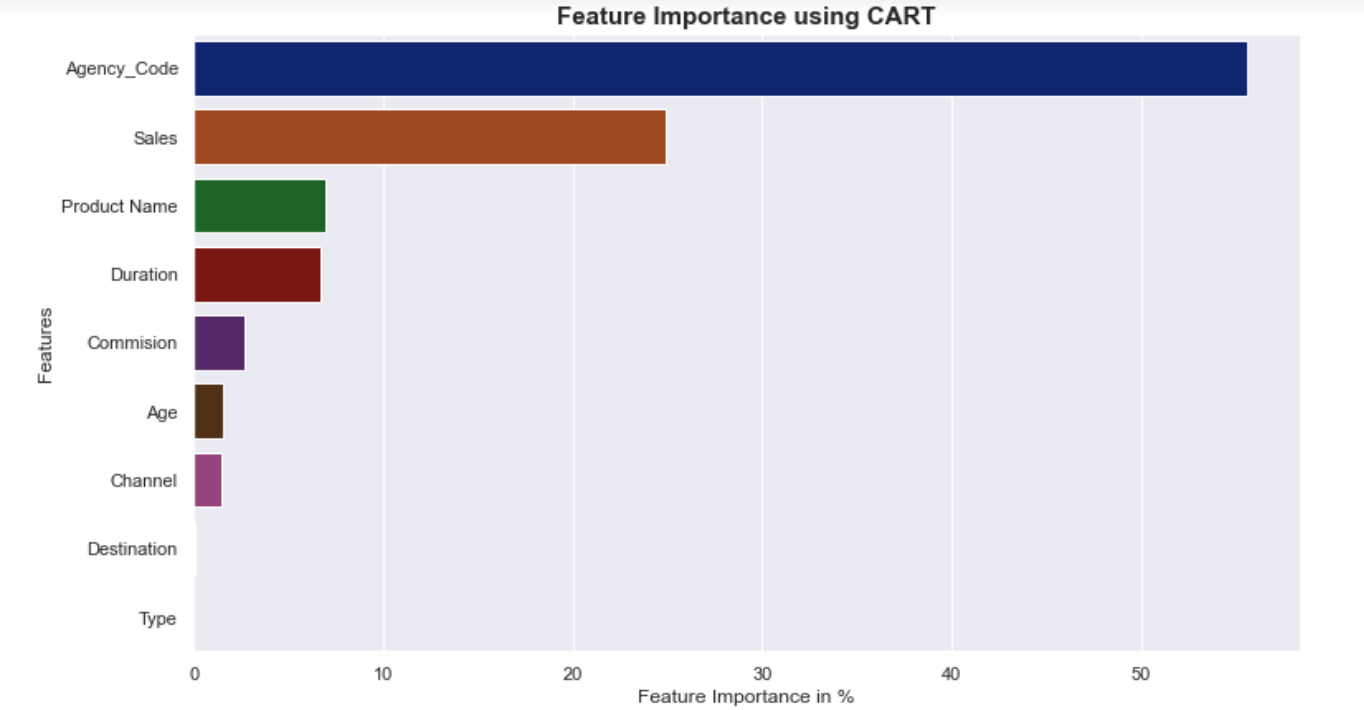

Fig 16: The above bar plot shows the feature importance of various variables in the given dataset which plays a major role in CART modelling.

	Imp
Agency_Code	0.556011
Sales	0.249295
Product Name	0.069635
Duration	0.067060
Commision	0.027085
Age	0.015485
Channel	0.014727
Destination	0.000701
Type	0.000000

Table 7: Table showing the feature importance for the CART modelling technique.

From the above table, we can see that the variable Agency_Code has the maximum importance while building the CART model building followed by Sales, while Type of tour insurance firms has no importance at all in model building.

2.RANDOM FOREST

```
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [6, 7], 'max_features': [4, 6],
                         'min_samples_leaf': [4, 5],
                         'min_samples_split': [110, 115],
                         'n_estimators': [101, 301]})
```

The above is the list of parameters that has been loaded into the grid search CV

```
{'max_depth': 7,
 'max_features': 4,
 'min_samples_leaf': 5,
 'min_samples_split': 110,
 'n_estimators': 101}
```

The above set of parameters were considered the best for this modelling.

- Selection of different parameters:

a. Max_depth: The maximum depth is the depth till which you allow the tree to grow to. The deeper you allow, the more complex your model will become. If you increase max_depth, training error will also increase. So, it is always recommended to keep the maximum depth between 7 and 10. Here I have given an array of values from 5 to 10 out of which the gridsearch CV chose max_depth=7 better for max_depth among the array of inputted values.

b. Min_samples_leaf: Usually for training we take 70% of the given data. In min sample leaf, we give 1-3% of the total number of the remaining 70% records.

c. Min_samples_split: By min_sample_split, we mean to say that minimum samples that has to be present in the leaf before its next split.In min sample split, we give at least 3 times the size of min sample leaves.

d. Max_features: max_featrures is the maximum number of features from the given dataset (the columns in the dataset) that can be considered before splitting a node into its leaf nodes.

e. **n-estimators:** n-estimators is the number of trees that i want to build within random classifier.

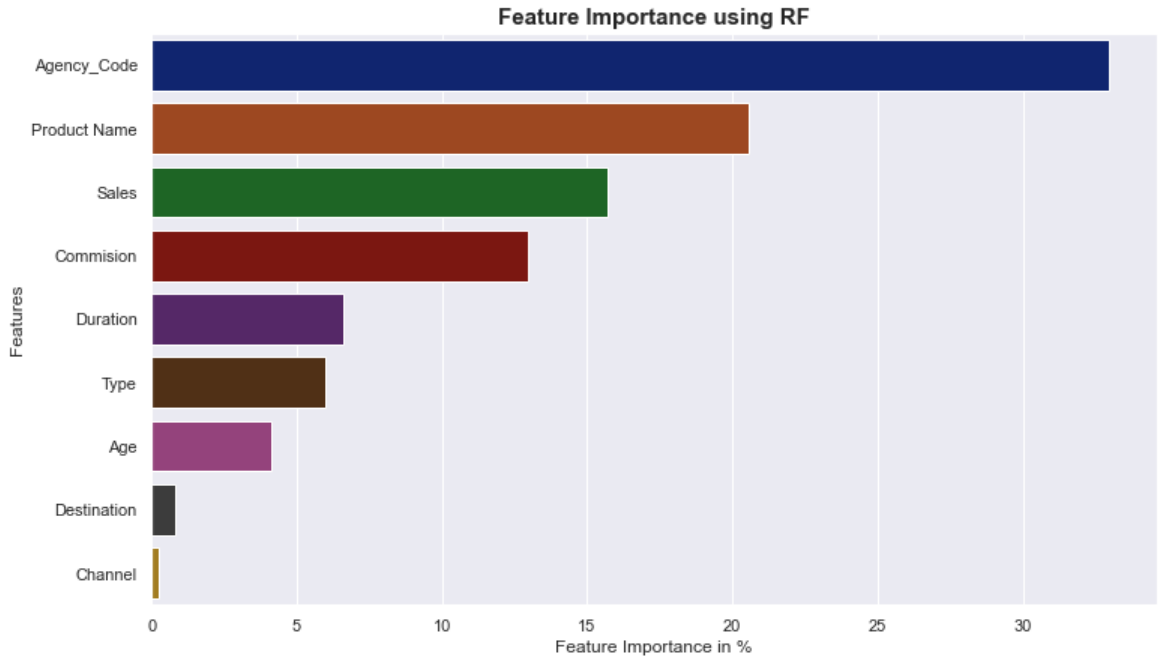

Fig 17: The above bar plot shows the feature importance of various variables in the given dataset which plays a major role in Random Forest modelling.

	Imp
Agency_Code	0.329741
Product Name	0.205587
Sales	0.157130
Commision	0.129780
Duration	0.066147
Type	0.059836
Age	0.041437
Destination	0.007930
Channel	0.002413

Table 8: Table showing the feature importance for the RF modelling technique.

From the above table, we can see that the variable Agency_Code has the maximum importance while building the RF model building followed by Product_Name, while Distribution channel of tour insurance agencies has least importance among all the variables in model building.

3.ARTIFICIAL NEURAL NETWORKS

```
GridSearchCV(cv=3, estimator=MLPClassifier(),
             param_grid={'activation': ['logistic', 'relu'],
                         'hidden_layer_sizes': [(100, 100, 100)],
                         'max_iter': [10000], 'solver': ['sgd', 'adam'],
                         'tol': [0.1, 0.01]})
```

The above is the list of parameters that has been loaded into the grid search CV

```
{'activation': 'relu',
 'hidden_layer_sizes': (100, 100, 100),
 'max_iter': 10000,
 'solver': 'adam',
 'tol': 0.1}
```

The above set of parameters were considered the best for this modelling.

- Selection of different parameters:

a) activation: An activation function in a neural network defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of network. Out of the two given activation function, relu was selected the best because relu allows back propagation while simultaneously making computational efficient. Relu does not activate all the neurons at the same time.

b) Hidden_layer_sizes: hidden layers=square root of number of independent variables or hidden layers=square of number of independent variables. Here hidden layer= (100,100,100) means I am interested in building a three-layer neuron network with 100 neurons in each of it.

c) Max_iter: max_iter is equivalent to maximum number of epochs you want the model to get trained on. It is called as maximum because the learning could get stopped before reaching the maximum number of iterations. Larger the iteration value given, greater the execution time.

d) solver: Solvers are one of the hyperparameters of the ANN. Solver is the algorithm used in the process of back propagation to calculate the weights of the neural network.

e) **tol:** If larger tolerance values are opted, less execution time is required and lesser accuracy will be obtained. Otherwise, smaller tolerance value opted, longer execution time taken and greater accuracy will be attained. Here, in the grid search CV. I have given a tuple of both 0.01 and 0.1 values out of which 0.1 was predicted the best.

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

1.CART

a. Classification Report

- Training Data

	precision	recall	f1-score	support
0	0.83	0.91	0.87	1471
1	0.73	0.55	0.63	629
accuracy			0.81	2100
macro avg	0.78	0.73	0.75	2100
weighted avg	0.80	0.81	0.80	2100

From the above classification report, we can see that using the CART model, the precision for training data is 0.73, recall is 0.55, f1 score is 0.63 and accuracy is 0.81.

- Testing Data

	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	0.76	0.77	0.75	900

From the above classification report, we can see that using the CART model, the precision for testing data is 0.74, recall is 0.46, f1 score is 0.57 and accuracy is 0.77.

b. ROC curve and ROC_AUC score

➢ Training Data

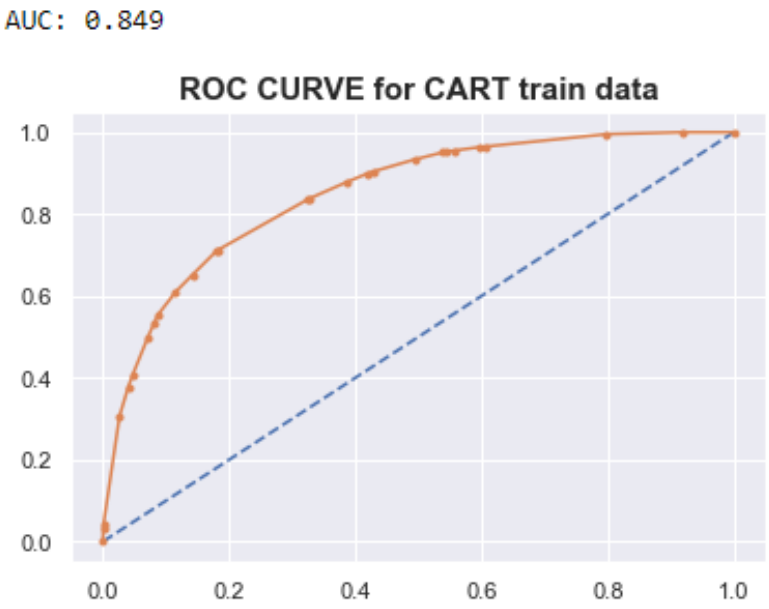

Fig 18: The ROC curve for the training data using the CART model

Here, the AUC score is 0.849.

➢ Testing Data

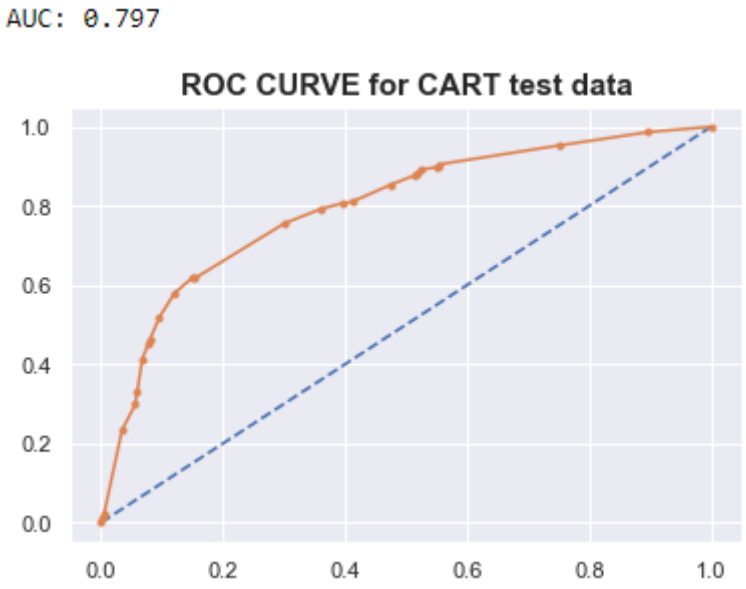

Fig 19: The ROC curve for the testing data using the CART model

Here, the AUC score is 0.797.

From the above ROC_curve for both testing and training data, we can see that the curve becomes flatter for the testing data in comparison to the testing data. Even the AUC score drops from 0.849 to 0.797 which shows that this model is not good enough.

c. Confusion Matrix

➢ Training Data

```
array([[1342,  129],
       [ 280,  349]], dtype=int64)
```

Given above is the confusion matrix for Training data using the CART model.

- Here, True Positive =1342, which is actually True (value=positive or 1) and has been predicted True too
- False Negative=129, which is actually True (value=positive or 1), but predicted False (value=negative or 0)
- False Positive =280, which is actually False (value=negative or 0), but predicted True (1)
- True Negative =349, which is actually False and has been predicted False too

➢ Testing Data

```
array([[556,  49],
       [158, 137]], dtype=int64)
```

Given above is the confusion matrix for Testing data using the CART model.

- Here, True Positive =556, which is actually True (value=positive or 1) and has been predicted True too
- False Negative=49 which is actually True (value=positive or 1), but predicted False (value=negative or 0)
- False Positive =158, which is actually False (value=negative or 0), but predicted True (1)
- True Negative =137, which is actually False and has been predicted False too

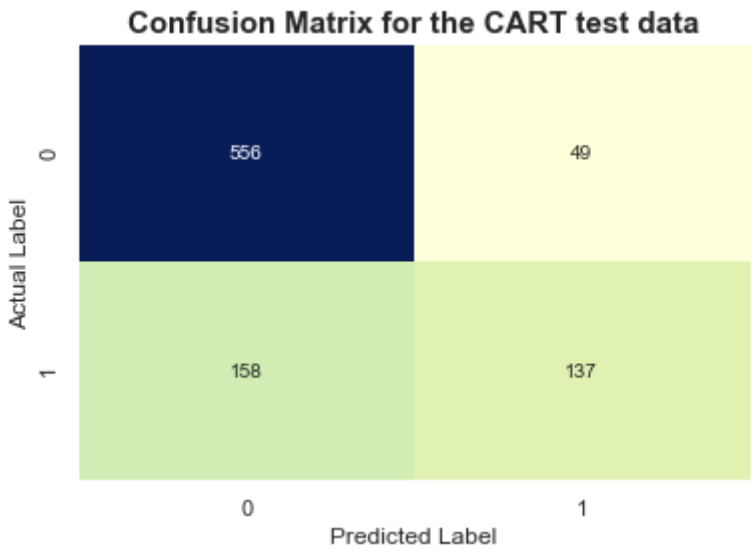

Fig 20: The figure shows the confusion matrix for the CART test data.

d. Accuracy

```
Accuracy Score of the above created CART model is 77.0 %
```

INFERENCES:

The accuracy of the test data model is less than the train data model. The model is good.

2.RANDOM FOREST

a. Classification Report

- Training Data

	precision	recall	f1-score	support
0	0.83	0.91	0.87	1471
1	0.72	0.57	0.64	629
accuracy			0.81	2100
macro avg	0.78	0.74	0.75	2100
weighted avg	0.80	0.81	0.80	2100

From the above classification report, we can see that using the Random Forest model, the precision for training data is 0.72, recall is 0.57, f1 score is 0.64 and accuracy is 0.81.

➢ Testing Data

	precision	recall	f1-score	support
0	0.79	0.91	0.85	605
1	0.74	0.49	0.59	295
accuracy			0.78	900
macro avg	0.76	0.70	0.72	900
weighted avg	0.77	0.78	0.76	900

From the above classification report, we can see that using the Random Forest model, the precision for testing data is 0.74, recall is 0.49, f1 score is 0.59 and accuracy is 0.78.

b. ROC curve and ROC_AUC score

➢ Training Data

```
AUC: 0.852
```

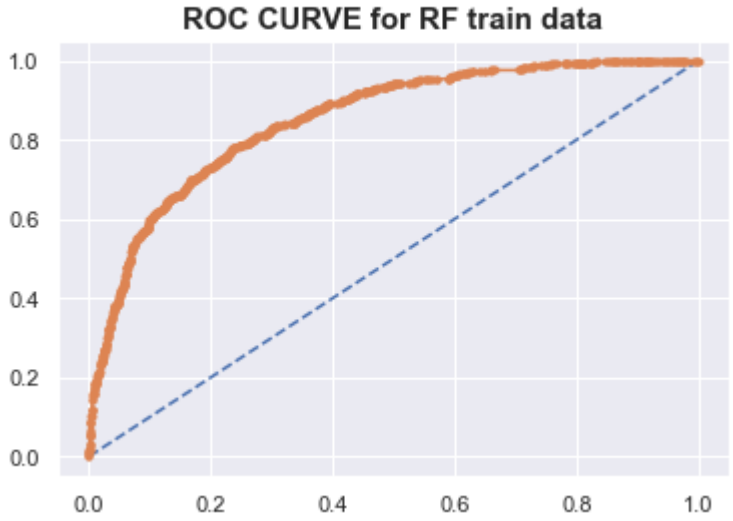

Fig 21: The ROC curve for the training data using the Random Forest model

➢ Testing Data

AUC: 0.822

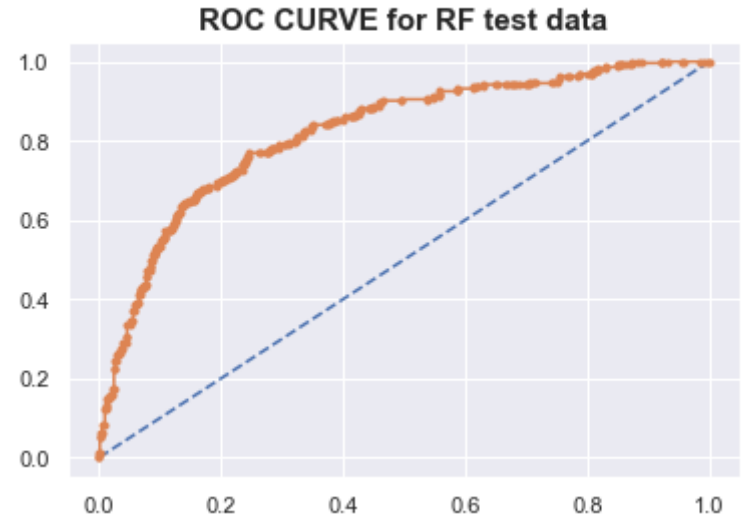

Fig 22: The ROC curve for the testing data using the Random Forest model

From the above ROC_curve for both testing and training data, we can see that the curve is more or less the same in both training and testing data with minute differences. The AUC score drops from 0.852 to 0.822; this drop is negligible and can be ignored. Thus, we can conclude this model is fair enough.

c. Confusion Matrix

➢ Training Data

```
array([[1332,  139],
       [ 269,  360]], dtype=int64)
```

Given above is the confusion matrix for Training data using the Random Forest model.

- Here, True Positive =1332, which is actually True (value=positive or 1) and has been predicted True too
- False Negative=139, which is actually True (value=positive or 1), but predicted False (value=negative or 0)
- False Positive =269, which is actually False (value=negative or 0), but predicted True (1)
- True Negative =360, which is actually False and has been predicted False too

➢ Testing Data

```
array([[554,  51],
       [155, 140]], dtype=int64)
```

Given above is the confusion matrix for Training data using the Random Forest model.

- Here, True Positive =554, which is actually True (value=positive or 1) and has been predicted True too
- False Negative=51, which is actually True (value=positive or 1), but predicted False (value=negative or 0)
- False Positive =155, which is actually False (value=negative or 0), but predicted True (1)
- True Negative =140, which is actually False and has been predicted False too

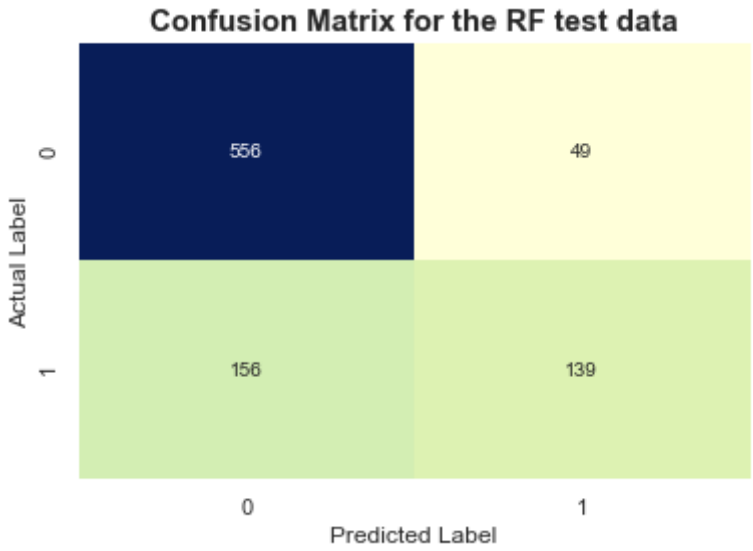

Fig 23: The figure shows the confusion matrix for the RNN test data.

d. Accuracy

```
Accuracy Score of the above created RF model is 77.0 %
```

3.ARTIFICIAL NEURAL NETWORK

a. Classification Report

- Training Data

	precision	recall	f1-score	support
0	0.84	0.86	0.85	1471
1	0.66	0.62	0.64	629
accuracy			0.79	2100
macro avg	0.75	0.74	0.74	2100
weighted avg	0.79	0.79	0.79	2100

From the above classification report, we can see that using the Artificial neural network model, the precision for training data is 0.66, recall is 0.62, f1 score is 0.64 and accuracy is 0.79.

- Testing Data

	precision	recall	f1-score	support
0	0.79	0.88	0.83	605
1	0.68	0.51	0.58	295
accuracy			0.76	900
macro avg	0.73	0.70	0.71	900
weighted avg	0.75	0.76	0.75	900

From the above classification report, we can see that using the Artificial neural network model, the precision for testing data is 0.68, recall is 0.51, f1 score is 0.58 and accuracy is 0.76.

b. ROC curve and ROC_AUC score

- Training Data

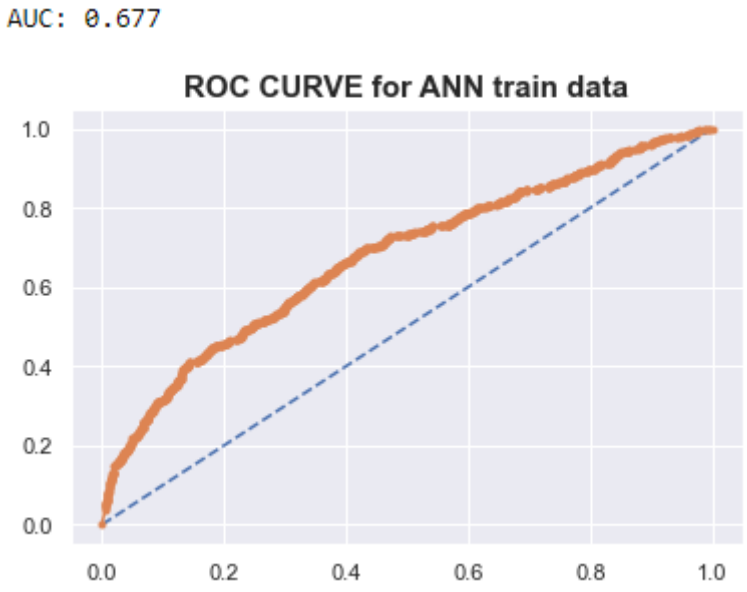

Fig 24: The ROC curve for the training data using the Artificial Neural Network model

➢ Testing Data

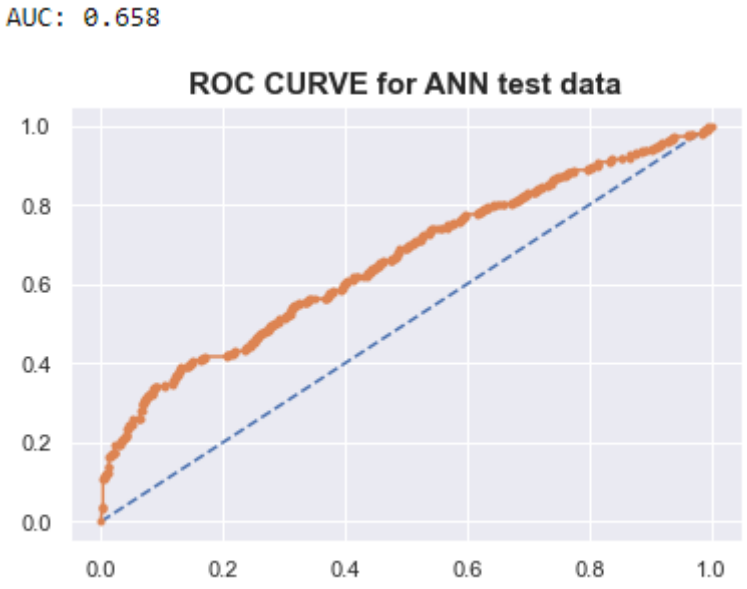

Fig 25: The ROC curve for the testing data using the Artificial Neural Network model

From the above ROC_curve for both testing and training data, we can see that the curve becomes more weaker after modelling using ANN. Even the AUC score drops from 0.677 to 0.658 which shows that this model is very poor.

c. Confusion Matrix

➢ Training Data

```
array([[1271,  200],
       [ 241,  388]], dtype=int64)
```

Given above is the confusion matrix for Training data using the ANN model.

- Here, True Positive =1271, which is actually True (value=positive or 1) and has been predicted True too
- False Negative=200 which is actually True (value=positive or 1), but predicted False (value=negative or 0)
- False Positive =241, which is actually False (value=negative or 0), but predicted True (1)
- True Negative =388, which is actually False and has been predicted False too

➢ Testing Data

```
array([[534,  71],
       [144, 151]], dtype=int64)
```

Given above is the confusion matrix for Training data using the ANN model.

- Here, True Positive =534, which is actually True (value=positive or 1) and has been predicted True too
- False Negative=71 which is actually True (value=positive or 1), but predicted False (value=negative or 0)
- False Positive =144, which is actually False (value=negative or 0), but predicted True (1)
- True Negative =151, which is actually False and has been predicted False too

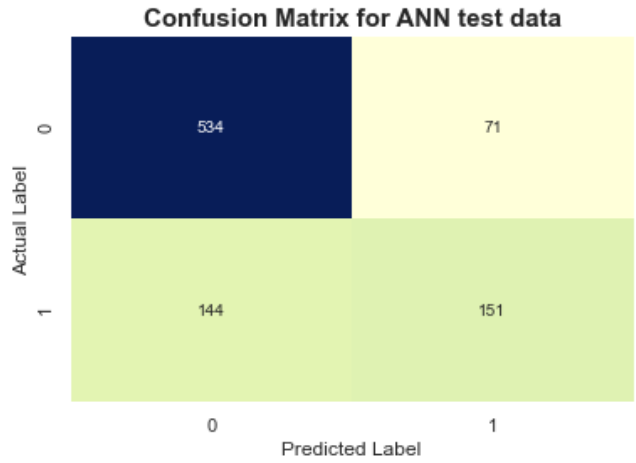

Fig 26: The figure shows the confusion matrix for the ANN test data.

d. Accuracy

```
Accuracy Score  of the above created ANN model is 76.0 %
```

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

	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.81	0.77	0.81	0.77	0.79	0.76
AUC	0.849	0.797	0.854	0.821	0.677	0.658
Recall	0.55	0.46	0.60	0.48	0.62	0.51
Precision	0.73	0.74	0.71	0.73	0.66	0.68
F1 Score	0.63	0.57	0.65	0.58	0.64	0.58

From the above table we have compared the performance metrics of all three models. It is observed that both CART and Random Forest is having same accuracy for the test data, but when we consider other metrics such as recall, precision and f1 score, Random forest has highest values in these metrics when compared to CART. Since it is said that the best model should have highest accuracy, precision, f1 score and recall, I would say out of all the three models created, Random Forest came out as the best model.

We have also compared the Roc curves of the following models and as we can see the Roc curve of Random forest is more well defined (while comparing respective model train data and test data) than the other two as well. It shows the model is working better than the other too. Also, is it noticed that the AUC score of train and test data of Random Forest is highest among all the three models. Hence, we can finalise the Random Model to be the best one among all three models.

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

- More real time data should be included for better modelling and analysis.
- According to the data given, 90%of the insurance is done via online platform.
- Even though more sales happen via Agency than Airlines, claims happen more for Airlines. The reason has to be found out.
- Also, almost all the offline cases are shown to have raised the claims. Reasons need to be found out.
- Also based on the RF model we are getting 81%accuracy, according to the claim pattern, customers should book the airline tickets or plans.

THE END