greatlearning
Power Ahead

DATA MINING PROJECT BUSINESS REPORT

VAISHNAV U
PGP-DSBA ONLINE
03/01/2023

Table of Contents

Content

Problem-1

Problem-2

List of tables

List of figures

Bank marketing data

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. The task is to identify the segments based on credit card usage.

Introduction

The data consists of 210 rows and 7 columns. Based on the columns we will develop a customer segmentation which in-turn helps to give promotional offers to its customers.Using clustering technique proper customer segmentation can be conducted

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)

Sample of dataset

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

Table 1.1

Problem-1

1.1 **Read the data and do exploratory data analysis (3 pts). Describe the data briefly. Interpret the inferences for each (3 pts). Initial steps like head() .info(), Data Types, etc . Null value check. Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Summary stats, Skewness, Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.**

Exploratory data analysis

1	spending	210 non-null	float 64
2	advance_payments	210 non-null	float 64
3	probability_of_full_payment	210 non-null	float 64
4	current_balance	210 non-null	float 64
5	credit_limit	210 non-null	float 64
6	min_payment_amt	210 non-null	float 64
7	max_spent_in_single_shopping	210 non-null	float 64

Descriptive statistics of data

	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.847524	2.909699	10.5900	12.27000	14.35500	17.305000	21.1800
advance_payments	210.0	14.559286	1.305959	12.4100	13.45000	14.32000	15.715000	17.2500
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.85690	0.87345	0.887775	0.9183
current_balance	210.0	5.628533	0.443063	4.8990	5.26225	5.52350	5.979750	6.6750
credit_limit	210.0	3.258605	0.377714	2.6300	2.94400	3.23700	3.561750	4.0330
min_payment_amt	210.0	3.700201	1.503557	0.7651	2.56150	3.59900	4.768750	8.4560
max_spent_in_single_shopping	210.0	5.408071	0.491480	4.5190	5.04500	5.22300	5.877000	6.5500

Table 1.2

From the above tables, it is clear that the dataset doesn't contain any null values. Also, all the columns are numerical. The descriptive statistics table shows that all the data points are fairly distributed. Average spending as per the data collected is 14.84(in 1000s) the minimum spending is 10.5(in 1000s) and the maximum spending is 21.18(in 1000s). The amount paid by customers in advance has an average of 14.55(in 100s) with a standard deviation of 1.30, the minimum advance payment made is 12.41(in 100s) and the maximum is 17.25(in 100s). The average probability of payment done in full by the customer to the bank is 87%, the minimum probability is 80% and the maximum is 91%. One of the most important features in this customer segmentation is the balance amount left in the account to make purchases. The average current balance of customers is 5.62(in 1000s), and the minimum and maximum current balances are 4.89 and 6.69 respectively.Average credit limit is 3.25(in 10000s).The minimum limit of amount in credit card for a customer is 2.63(in 10000s) and the maximum amount is 4.03(in 10000s).

Univariate analysis

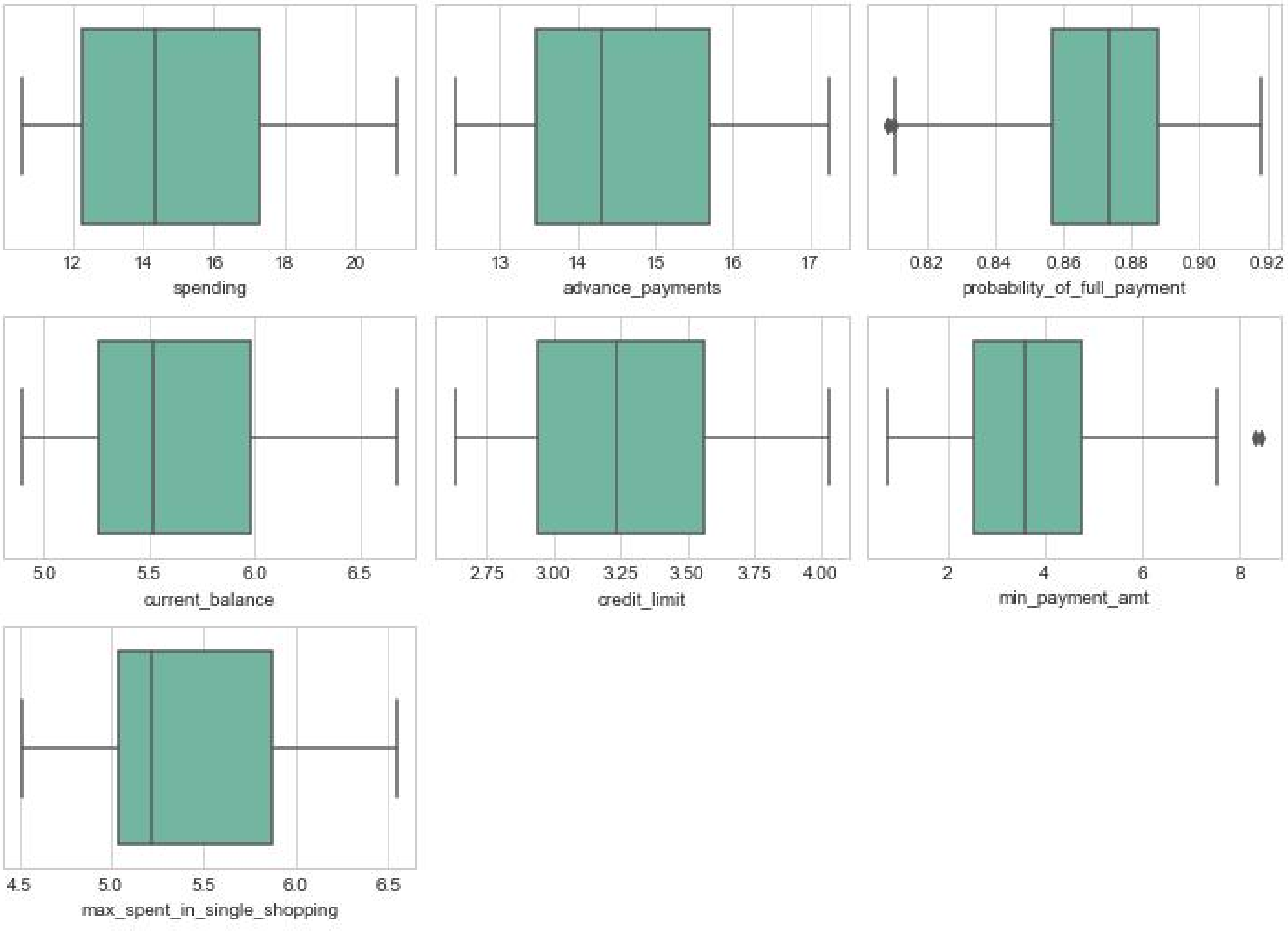

Fig 1.1

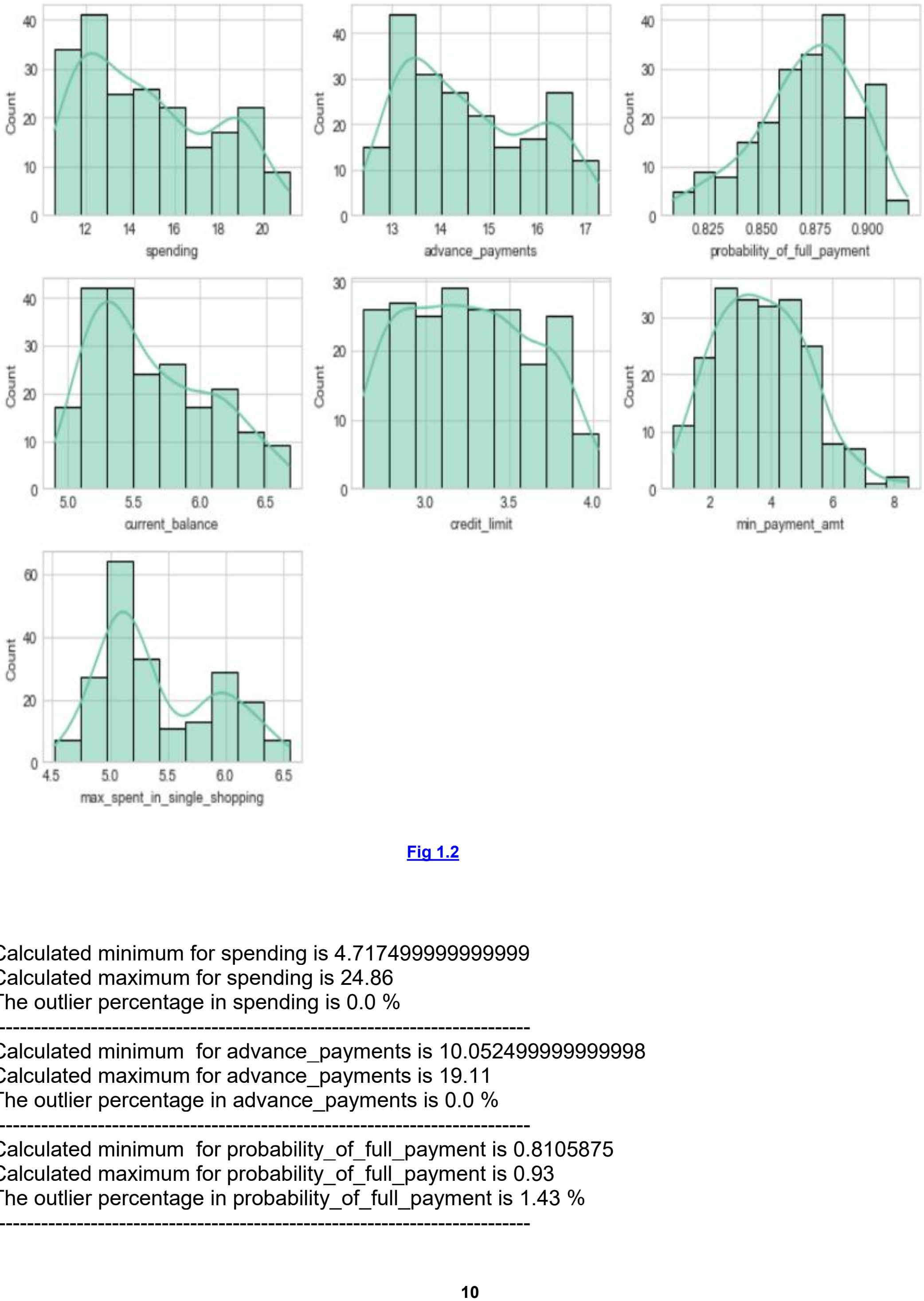

Fig 1.2

Calculated minimum for spending is 4.717499999999999
Calculated maximum for spending is 24.86
The outlier percentage in spending is 0.0 %

Calculated minimum for advance_payments is 10.052499999999998
Calculated maximum for advance_payments is 19.11
The outlier percentage in advance_payments is 0.0 %

Calculated minimum for probability_of_full_payment is 0.8105875
Calculated maximum for probability_of_full_payment is 0.93
The outlier percentage in probability_of_full_payment is 1.43 %

Calculated minimum for current_balance is 4.186
Calculated maximum for current_balance is 7.06
The outlier percentage in current_balance is 0.0 %

Calculated minimum for credit_limit is 2.017375
Calculated maximum for credit_limit is 4.49
The outlier percentage in credit_limit is 0.0 %

Calculated minimum for min_payment_amt is -0.7493749999999992
Calculated maximum for min_payment_amt is 8.08
The outlier percentage in min_payment_amt is 0.95 %

Calculated minimum for max_spent_in_single_shopping is 3.797
Calculated maximum for max_spent_in_single_shopping is 7.12
The outlier percentage in max_spent_in_single_shopping is 0.0 %

Box plots shows that all the data points are approximately normally distributed.Also probability_of_full_payment and min_payment_amt column have outliers which does not affect the data.There are only 1.43% of outliers in probability_of_full_payment column and 0.95% outliers in min_payment_amt column.Histogram also shows the distribution of data.Spending,advance payment and max_spent_in_single_shopping columns clearly shows that the data points can be segmented to create another set of data with it’s unique properties.

Bivariate analysis

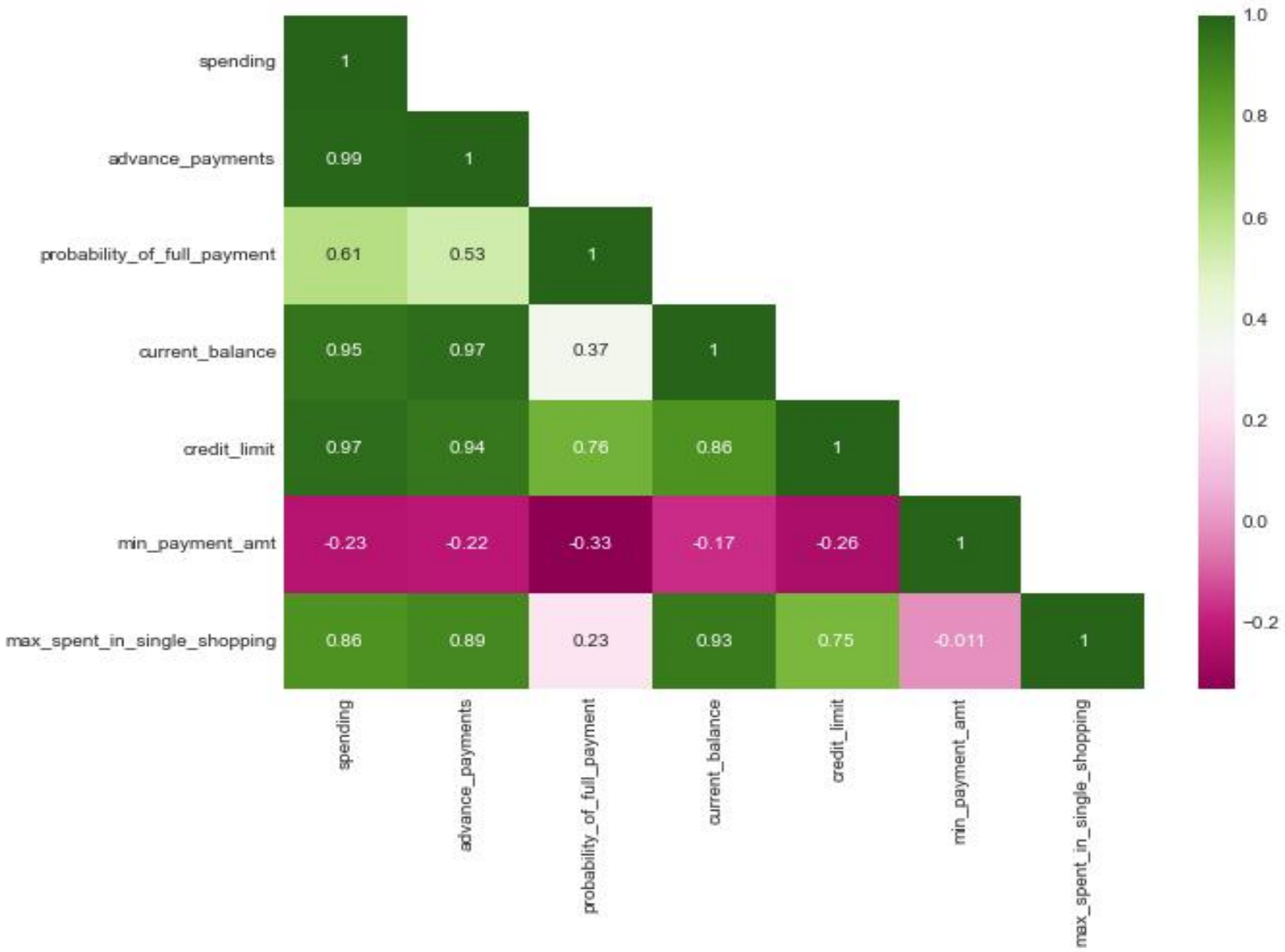

Fig 1.3

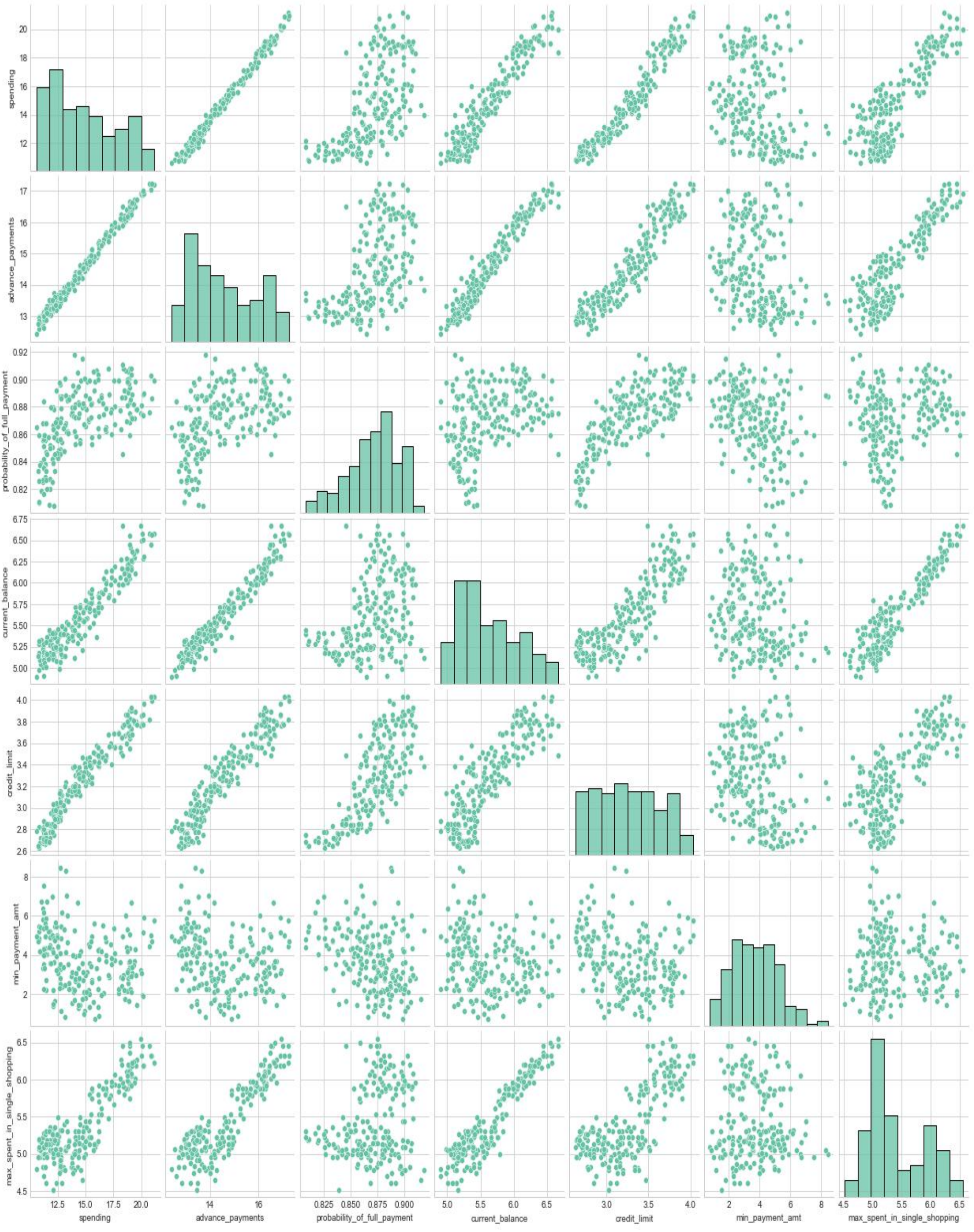
spending
advance_payments
probability_of_full_payment
current_balance
credit_limit
min_payment_amt
max_spent_in_single_shopping

Fig 1.4

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt
spending	1.000000	0.994341	0.608288	0.949985	0.970771	-0.229572
advance_payments	0.994341	1.000000	0.529244	0.972422	0.944829	-0.217340
probability_of_full_payment	0.608288	0.529244	1.000000	0.367915	0.761635	-0.331471
current_balance	0.949985	0.972422	0.367915	1.000000	0.860415	-0.171562
credit_limit	0.970771	0.944829	0.761635	0.860415	1.000000	-0.258037
min_payment_amt	-0.229572	-0.217340	-0.331471	-0.171562	-0.258037	1.000000
max_spent_in_single_shopping	0.863693	0.890784	0.226825	0.932806	0.749131	-0.011079

Table 1.3

Heat map, pair plot, and correlation table show that most of the columns are highly correlated.0.7 to 0.9 are considered a high correlation, 0.5 to 0.7 is moderately correlated, and below 0.3 indicate variables that have a low correlation. So we can conclude that spending and advance payment are highly correlated. Spending and advance payment show a high correlation with other variables such as credit limit, current balance, etc. Minimum payment in combination with other variables is the only place we can spot a negative correlation .ie a decrease of a value in a variable, and an increase of the subsequent variable value.

Skewness :

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

Table 1.4

If the skewness is between -0.5 and 0.5, the data are fairly symmetrical.

If the skewness is between -1 and – 0.5 or between 0.5 and 1, the data are moderately skewed.

If the skewness is less than -1 or greater than 1, the data are highly skewed.

As skewness is considered we can conclude that most of the data are fairly symmetrical or moderately skewed category.

1.2 **Do you think scaling is necessary for clustering in this case? Justify The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for example std dev, variance, etc. Should justify whether there is a necessity for scaling and which method is he/she using to do the scaling. Can also comment on how that method works.**

	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 1.5

The above table shows the data after it is being scaled .

Yes,**scaling** is necessary for clustering .Clustering models are distance-based algorithms. In order to measure similarities between observations and form clusters they use a distance metric. So, features with high ranges will have a bigger influence on the clustering. Therefore, standardization is required before building a clustering model.Here we have used z-score standardization.This technique consists of subtracting the mean of the column from each value in a column, and then dividing the result by the standard deviation of the column. The formula to achieve this is the following: $z = (x-\mu)/\sigma$

The result of standardization is that the features will be rescaled so that they'll have the properties of a standard normal distribution, as follows: $\mu=0 , \sigma=1$

Standard deviation of the columns are:

The standard deviation for spending item is 2.9

The standard deviation for advance_payments item is 1.3

The standard deviation for probability_of_full_payment item is 0.02

The standard deviation for current_balance item is 0.44

The standard deviation for credit_limit item is 0.38

The standard deviation for min_payment_amt item is 1.5

Standard deviation of the columns after scaling are:

The standard deviation scaled data for spending item is 1.0

The standard deviation scaled data for advance_payments item is 1.0

The standard deviation scaled data for probability_of_full_payment item is 1.0

The standard deviation scaled data for current_balance item is 1.0

The standard deviation scaled data for credit_limit item is 1.0

The standard deviation scaled data for min_payment_amt item is 1.0

The standard deviation scaled data for max_spent_in_single_shopping item is 1.0

Variance of the columns are:

The variance of spending item is 8.43

The variance of advance_payments item is 1.7

The variance of probability_of_full_payment item is 0.0

The variance of current_balance item is 0.2

The variance of credit_limit item is 0.14

The variance of min_payment_amt item is 2.25

The variance of max_spent_in_single_shopping item is 0.24

Variance of the columns after scaling are:

The variance of scaled data for spending item is 1.0

The variance of scaled data for advance_payments item is 1.0

The variance of scaled data for probability_of_full_payment item is 1.0

The variance of scaled data for current_balance item is 1.0

The variance of scaled data for credit_limit item is 1.0

The variance of scaled data for min_payment_amt item is 1.0

The variance of scaled data for max_spent_in_single_shopping item is 1.0

Box plot after scaling :

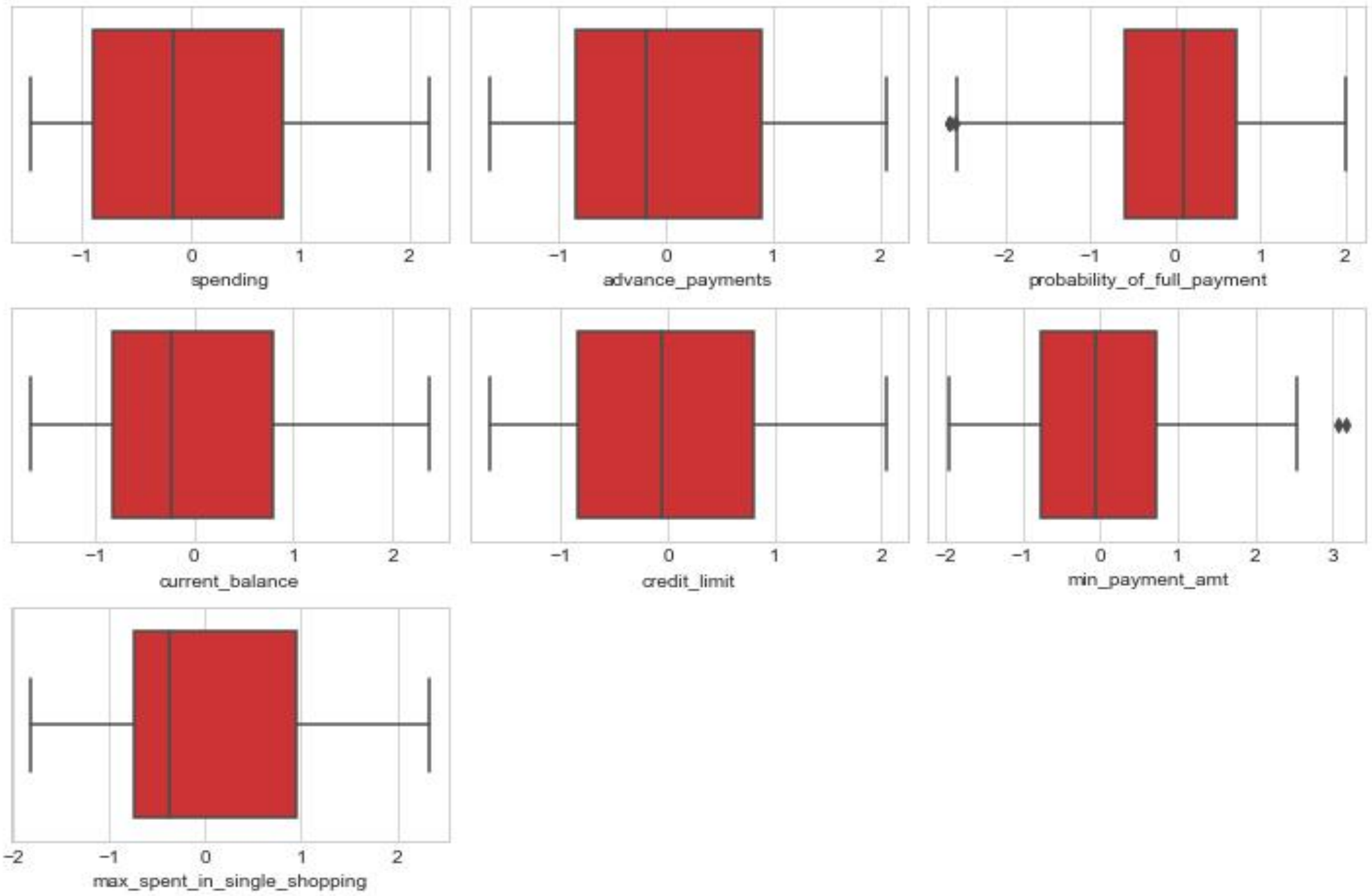

Fig 1.5

It is evident from the box plots that scaling does not effect the outliers in the data and the distribution of data.

1.3 **Apply hierarchical clustering to scaled data (3 pts). Identify the number of optimum clusters using Dendrogram and briefly describe them (4). Students are expected to apply hierarchical clustering. It can be obtained via Fclusters or Agglomerative Clustering. Report should talk about the used criterion, affinity and linkage. Report must contain a Dendrogram and a logical reason behind choosing the optimum number of clusters and Inferences on the dendrogram. Customer segmentation can be visualized using limited features or whole data but it should be clear, correct and logical. Use appropriate plots to visualize the clusters.**

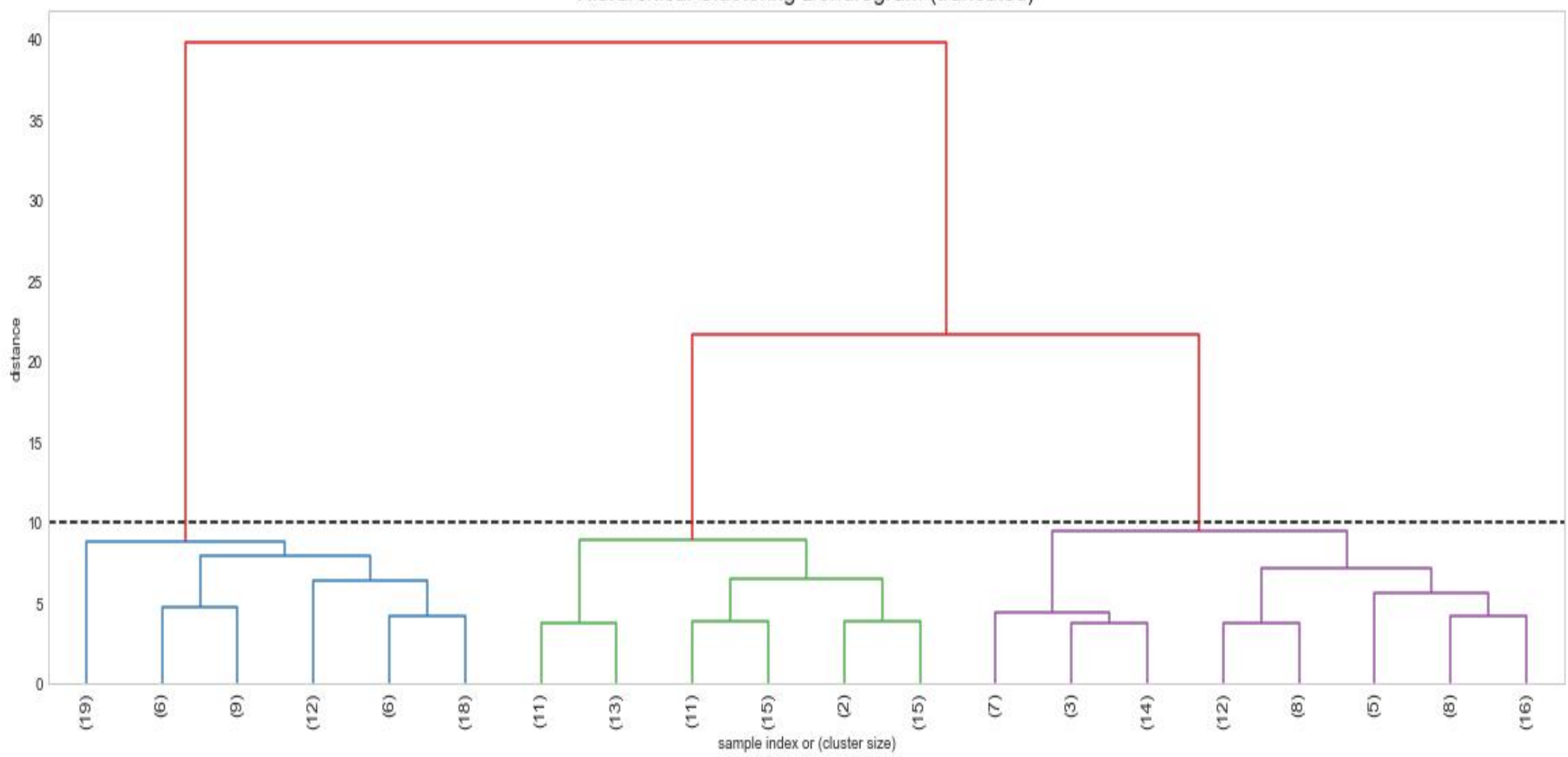

Fig 1.6

Here we have used Agglomerative Clustering. Agglomerative Clustering is a type of hierarchical clustering algorithm. It is an unsupervised machine learning technique that divides the data points into several clusters such that data points in the same cluster are more similar and data points in different clusters are dissimilar. A dendrogram is specific to the agglomerative hierarchical method of clustering. The agglomerative hierarchical method of clustering starts by considering each point as a separate cluster and starts joining points to clusters in a hierarchical fashion based on their distances. To get the optimal number of clusters for hierarchical clustering, we make use of a dendrogram (Fig 1.6) that shows the sequences of merges or splits of clusters. If two clusters are merged, the dendrogram will join them in a graph and the height of the join will be the distance between those clusters. Mathematically height of the join helps to predict the optimum number of clusters. The dendrogram shown above has the last 20 truncated clusters shown and the number of data points in each cluster is shown inside the bracket. Ward linkage method is used here which uses within sum of squares distance to combine data points to a cluster. Since the clustering process is completely controlled by the distance between two points and the distance between two clusters, the distance between two points can be determined by Euclidean Distance.

Thus 3 clusters can be considered as optimum based on business and mathematical points of view.

Pair plot to visualize the clustering:

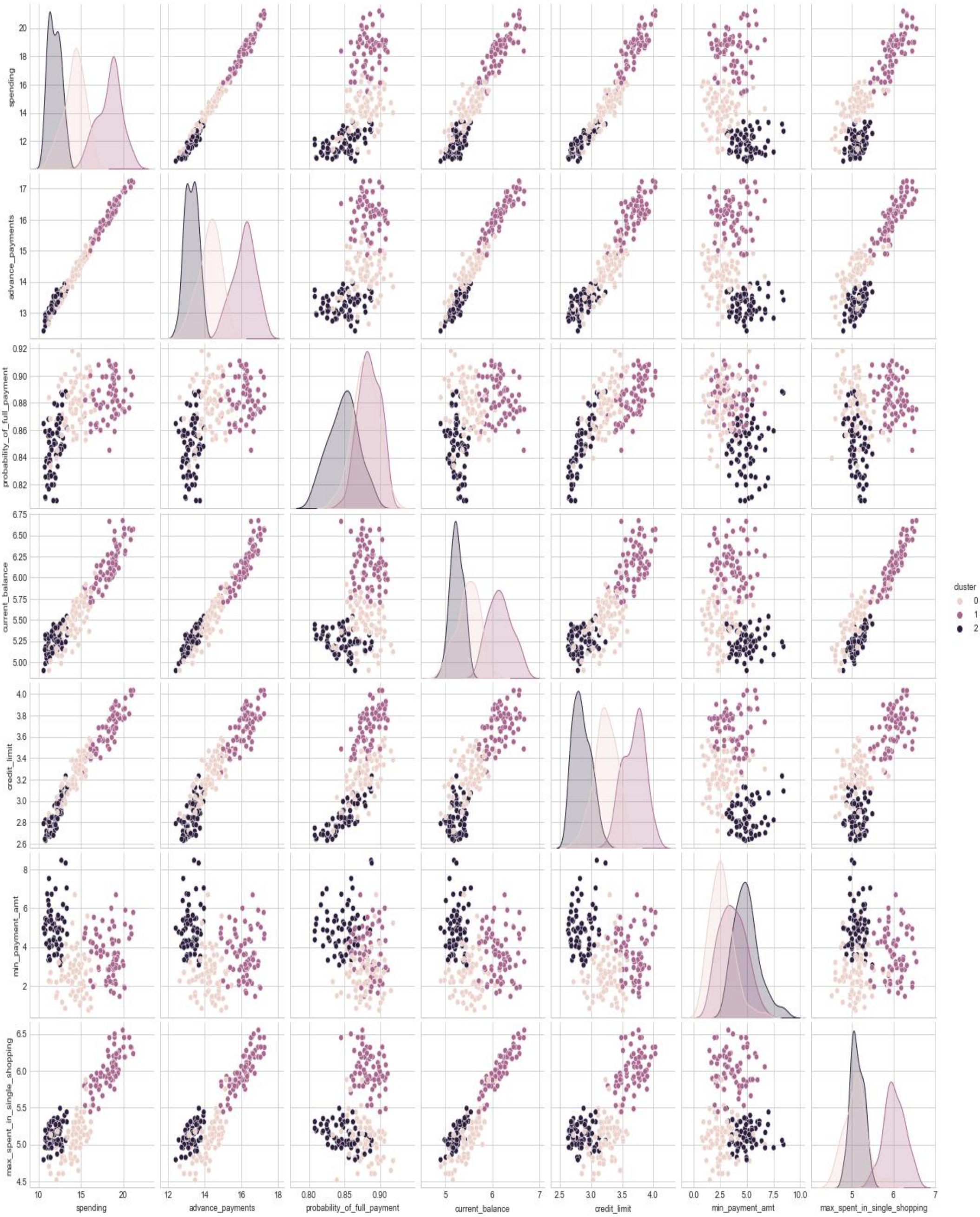
spending
advance_payments
probability_of_full_payment
current_balance
credit_limit
min_payment_amt
max_spent_in_single_shopping
cluster
0
1
2

Fig 1.7

spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	cluster
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	0
18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1
10.83	12.96	0.8099	5.278	2.641	5.182	5.185	2
17.99	15.86	0.8992	5.890	3.694	2.068	5.837	1

Table 1.6

The above table shows the sample of data after hierarchical clustering .From the first five rows we can spot the three clusters as 0,1 & 2.

The pair plot above specifically visualize all three clusters in different column combination.It is evident that the clustering technique worked well for the data set.

1.4 Apply K-Means clustering on scaled data and determine optimum clusters (2 pts). Apply elbow curve (3 pts). Interpret the inferences from the model (2.5 pts). K-means clustering code application with different number of clusters. Calculation of WSS(inertia for each value of k) Elbow Method must be applied and visualized with different values of K. Reasoning behind the selection of the optimal value of K must be explained properly. Report must contain logical and correct explanations for choosing the optimum clusters using the elbow method. Append cluster labels obtained from K-means clustering into the original data frame. Customer Segmentation can be visualized using appropriate graphs.

K-means clustering with different values :

Number of cluster = 4

```
array([1, 0, 1, 3, 1, 3, 3, 0, 1, 3, 1, 0, 3, 1, 0, 3, 0, 3, 0, 3, 3, 3,
       1, 3, 0, 2, 0, 3, 3, 3, 0, 3, 3, 0, 3, 3, 3, 3, 3, 1, 1, 0, 2, 1,
       3, 3, 0, 1, 1, 1, 3, 1, 1, 1, 1, 2, 3, 3, 3, 1, 0, 3, 3, 2, 0, 1,
       1, 0, 1, 0, 0, 3, 1, 1, 3, 1, 0, 3, 2, 0, 0, 0, 0, 1, 3, 2, 2, 2,
       2, 3, 0, 1, 0, 3, 0, 1, 1, 2, 3, 2, 0, 1, 2, 1, 0, 1, 1, 3, 0, 1,
       2, 0, 1, 3, 3, 2, 0, 0, 3, 1, 0, 3, 3, 3, 0, 0, 1, 3, 0, 0, 3, 0,
       0, 1, 3, 1, 1, 3, 2, 0, 2, 0, 3, 3, 0, 3, 1, 3, 0, 3, 0, 3, 0, 2,
       0, 0, 0, 3, 0, 2, 1, 3, 1, 2, 1, 3, 2, 0, 0, 3, 0, 3, 0, 1, 1, 1,
       0, 0, 2, 3, 0, 0, 0, 0, 2, 2, 0, 2, 0, 3, 0, 0, 3, 1, 0, 2, 1, 3,
       1, 3, 0, 2, 0, 3, 2, 0, 2, 0, 2, 2])
```

Fig 1.8

Number of cluster = 3

```
array([1, 2, 1, 0, 1, 0, 0, 2, 1, 0, 1, 2, 0, 1, 2, 0, 2, 0, 0, 0, 0, 0,
       1, 0, 2, 1, 2, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 1, 1, 2, 1, 1,
       0, 0, 2, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 2, 0, 0, 2, 2, 1,
       1, 2, 1, 0, 2, 0, 1, 1, 0, 1, 2, 0, 1, 2, 2, 2, 2, 1, 0, 2, 1, 2,
       1, 0, 2, 1, 2, 0, 0, 1, 1, 1, 0, 1, 2, 1, 2, 1, 2, 1, 1, 0, 0, 1,
       2, 2, 1, 0, 0, 1, 2, 2, 0, 1, 2, 0, 0, 0, 2, 2, 1, 0, 2, 2, 0, 2,
       2, 1, 0, 1, 1, 0, 1, 2, 2, 2, 0, 0, 2, 0, 1, 0, 2, 0, 2, 0, 2, 2,
       0, 2, 2, 0, 2, 1, 1, 0, 1, 1, 1, 0, 2, 2, 2, 0, 2, 0, 2, 1, 1, 1,
       2, 0, 2, 0, 2, 2, 2, 2, 1, 1, 0, 2, 2, 0, 0, 2, 0, 1, 2, 1, 1, 0,
       1, 0, 2, 1, 2, 0, 1, 2, 1, 2, 2, 2])
```

Fig 1.9

The above figures show the K means clusters using a different number of clusters(3 &4).There are 4 clusters in first figure namely 0,1,2 & 3 as in 3 clusters it is 0,1 & 2.

Optimum number of cluster can be found using the WCSS(inertia for each k value).

The wss value for different clusters is as shown below:

The WSS value for 2 clusters is 659.1717544870407

The WSS value for 3 clusters is 430.6589731513006

The WSS value for 4 clusters is 371.30172127754213

The WSS value for 5 clusters is 327.96082400790306

The WSS value for 6 clusters is 290.59003059682186

The WSS value for 7 clusters is 264.8315308747815

The WSS value for 8 clusters is 240.68372595015984

The WSS value for 9 clusters is 220.85285825594738

The WSS value for 10 clusters is 206.38291036015787

For a given number of clusters, the total within-cluster sum of squares (WCSS) is computed. That value of k is chosen to be optimum, where addition of one more cluster does not lower the value of total WCSS appreciably.

The same can be plotted using Elbow method.The Elbow method looks at the total WCSS as a function of the number of clusters.

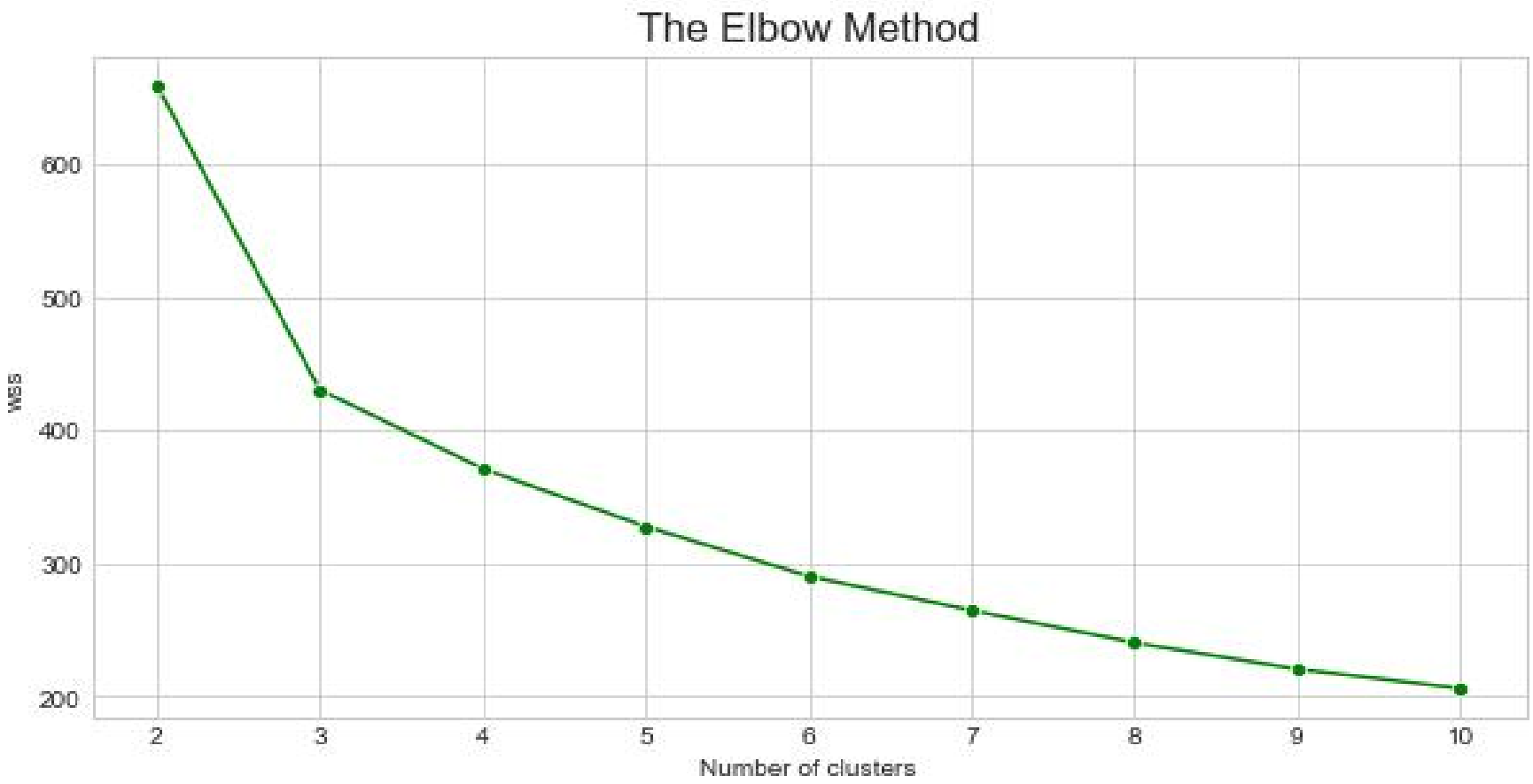

Fig 1.10

From the elbow plot we can conclude that 3 is the optimum number of cluster.Also when the wss value is found we can see a sudden drop in the value from cluster number 3.

spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	kmeans_cluster_3
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	1
10.83	12.96	0.8099	5.278	2.641	5.182	5.185	0
17.99	15.86	0.8992	5.890	3.694	2.068	5.837	1

Table 1.7

Table above shows the appended cluster labels into the data set provided.We have chosen 3 clusters , all of them are visible in the first 5 rows of the dataset.

Pair plot to visualize the k-means clustering:

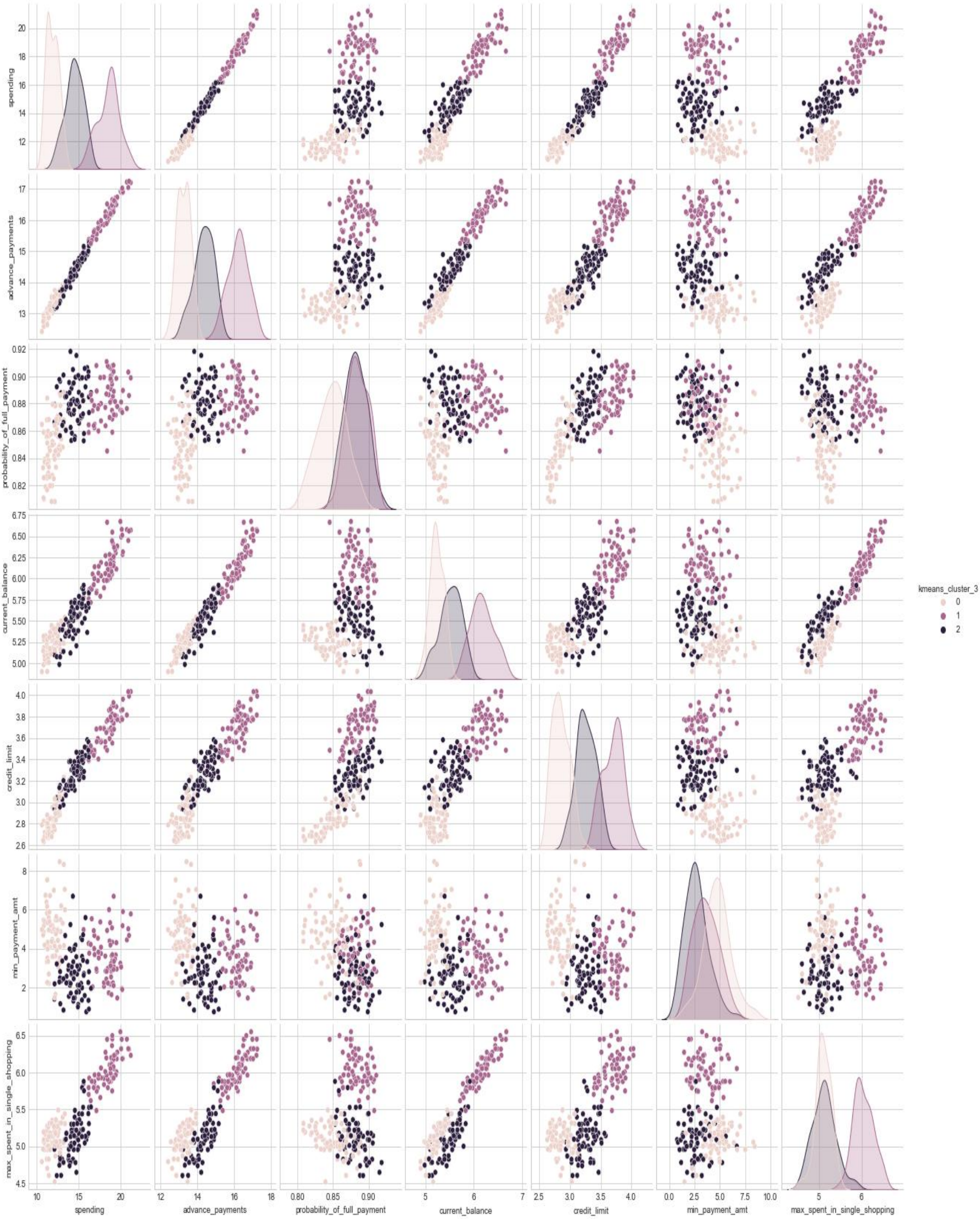
spending
advance_payments
probability_of_full_payment
current_balance
credit_limit
min_payment_amt
max_spent_in_single_shopping
kmeans_cluster_3
0
1
2

Fig 1.11

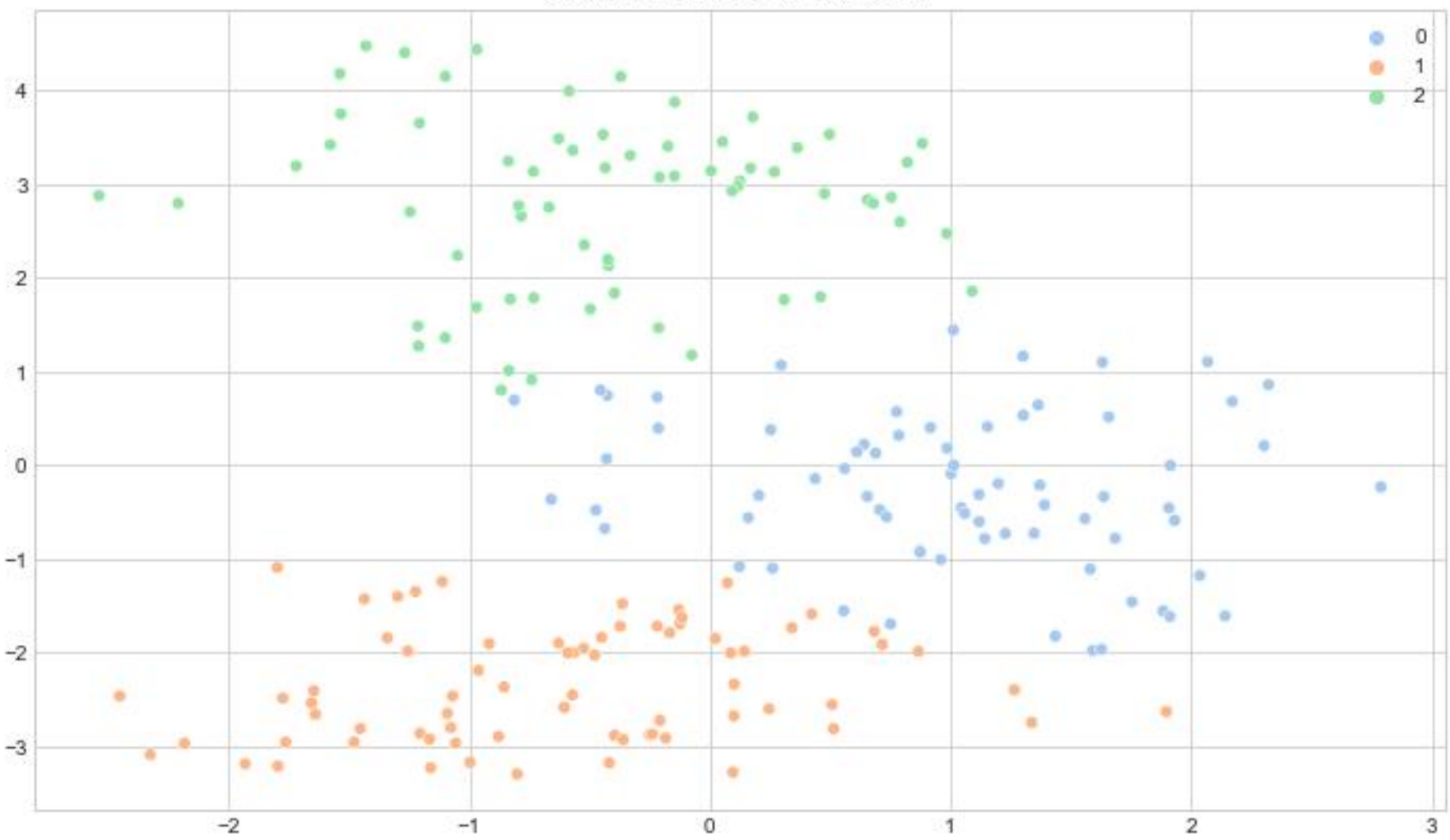

Fig 1.12

The pair plot above specifically visualize all three clusters in different column combination.It is evident that the clustering technique worked well for the data set.

Cluster plot is a helpful way to look at the spread and overlap of clusters. Ideally, for a perfect algorithm, the clusters will be well separated with no (or minimum) overlap.**Both of these plots suggest a well separated clusters.**

1.5 **Describe cluster profiles for the clusters defined (2.5 pts). Recommend different promotional strategies for different clusters in context to the business problem in-hand (2.5 pts). After adding the final clusters to the original data frame, do the cluster profiling. Divide the data in the finalized groups and check their means. Explain each of the group briefly. There should be at least 3-4 Recommendations. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks will only be allotted if the recommendations are correct and business specific. variable means. Students to explain the profiles and suggest a mechanism to approach each cluster. Any logical explanation is acceptable.**

Table showing mean of clusters(k-means)

kmeans_cluster_3	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	11.856944	13.247778	0.848253	5.231750	2.849542	4.742389	5.101722
1	18.495373	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701
2	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803

Table 1.8

The above table shows the three clusters obtained from k means clustering and the average of these clusters.

Low spending group

The first cluster 0 can be called as Low spending group .These group of customers make low advance payment ,less probability of paying the full amount ,less balance amount left in the account to make purchases ,lesser credit limit and less amount spent in one purchase.But these group of customers pay more than other groups while making monthly purchase.

Recommendation:

1.As these group of customers make monthly purchase more give promotional offers to basic house hold items for these customers.
2.Customers in these group are least in maximum amount spent in one purchase.So customer specific advertisement such as lesser amount grocery and utilities should be offered as a promotional strategy.
3.Provide coupons for the utilities/items with are not usually bought by the customers due to the price factor. This can ease them in buying the product.

High spending group

The second cluster 1 can be called as High spending group .These group of customers make high advance payment ,higher probability of paying the full amount ,more balance amount left in the account to make purchases ,higher credit limit and high amount spent in one purchase.

Recommendation:

1.For these group of customers any promotional offer/coupons will work.
2.Customer specific advertisement as for high price items should be provided.
3.These customers have the highest probability of payment done in full .So provide them with offers if they make one time payment.
4.Sales promotion through personal should be available for these customers.
5.Credit limit can be increased which in turn allows customers to purchase more.

Medium spending group

The third cluster 2 can be called as Medium spending group .As the name implies these customers spends lesser than high spending customers and more than low spending customers.These customers make least month payments and high spending in one purchase similar to high spending group.

Recommendation:

1. Credit limit can be increased for these customers .
2. As these customers have higher probability of full time payment ,promotional offer/coupons can be provided if they make payment in full.
3. Credit cards with low interest rates and few fees typically attract more customers.
4. Product give-aways and allowing potential customers to sample a product .

Table showing mean of clusters(Heirachical)

cluster	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	14.199041	14.233562	0.879190	5.478233	3.226452	2.612181	5.086178
1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371
2	11.872388	13.257015	0.848072	5.238940	2.848537	4.949433	5.122209

Table 1.9

The above table shows the three clusters obtained from hierarchical clustering and the average of these clusters.Similar to k means cluster we can divide the customers to High,Medium and low spending groups.Recommendation are same as given above.

Insurance data

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 and compare the models' performances in train and test sets.

Introduction

The data consists of 3000 rows and 10columns . Univariate ,bivariate analysis of the data will be conducted here based on the various factors. Then the model is created which predict the claim status and provide recommendations to management.Model will be created using CART & RF and their performance will be compared.

Data description

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

Sample of dataset

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	C2B	Airlines	No	0.70	Online	7	2.51	Customised Plan	ASIA
1	36	EPX	Travel Agency	No	0.00	Online	34	20.00	Customised Plan	ASIA
2	39	CWT	Travel Agency	No	5.94	Online	3	9.90	Customised Plan	Americas
3	36	EPX	Travel Agency	No	0.00	Online	4	26.00	Cancellation Plan	ASIA
4	33	JZI	Airlines	No	6.30	Online	53	18.00	Bronze Plan	ASIA

Table 2.1

Problem-2

Exploratory Data Analysis

	Column	Non-Null Count	Data type
1	Age	3000 non-null	int64
2	Agency_Code	3000 non-null	object
3	Type	3000 non-null	object
4	Claimed	3000 non-null	object
5	Commision	3000 non-null	float64
6	Channel	3000 non-null	object
7	Duration	3000 non-null	int64
8	Sales	3000 non-null	float64
9	Product Name	3000 non-null	object
10	Destination	3000 non-null	object

Descriptive statistics of data:

	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 2.2

Descriptive statistics of data show that there are 3000 rows in the data set. Null values are not present in the dataset. The columns Age, Commission, Duration, and Sales are numerical. All other columns are categorical/object types. The average age of insured people is 38, the minimum is 8 and the maximum is 84. Mostly occurring code of tour firms is EPX and there are 4 tour firms in total. There are two types of insurance firms in which Travel agency is the most frequent in the dataset. The target variable of the model is claimed and there are 2 unique values in the claimed column. There are 3 unique destinations and 5 product names in the dataset. Asia is the most frequent destination and a customized plan is the frequent product name. The average duration of the tour is 70 days.

2.1 **Read the data and do exploratory data analysis (4 pts). Describe the data briefly. Interpret the inferences for each (2 pts). Initial steps like head() .info(), Data Types, etc . Null value check. Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Summary stats, Skewness, Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.**

Univariate analysis

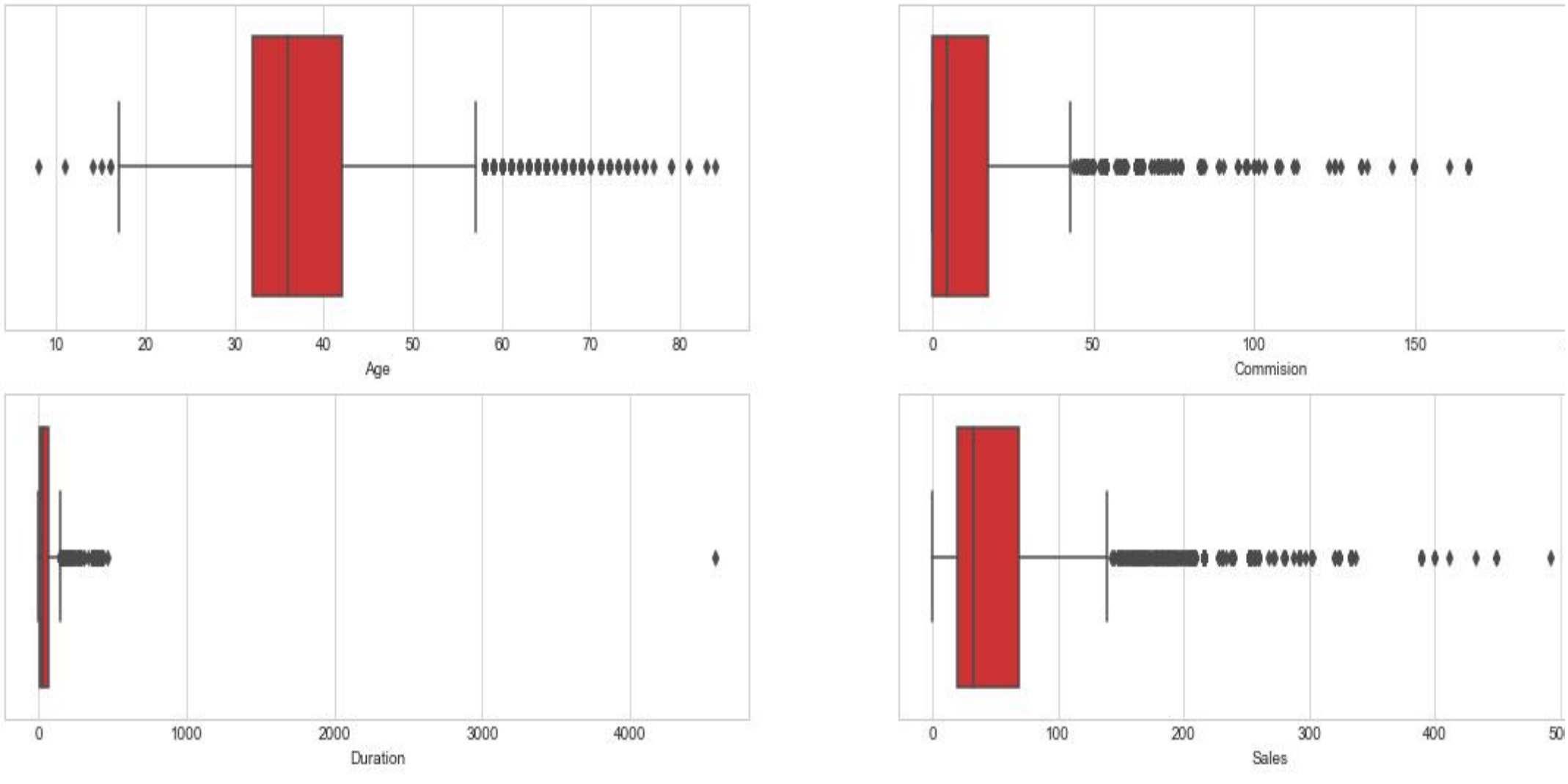

Fig 2.1

These are the box plots for numerical columns in the dataset. It is evident from the box plot that there are outliers in the data. Except age column all other numerical columns are right skewed.

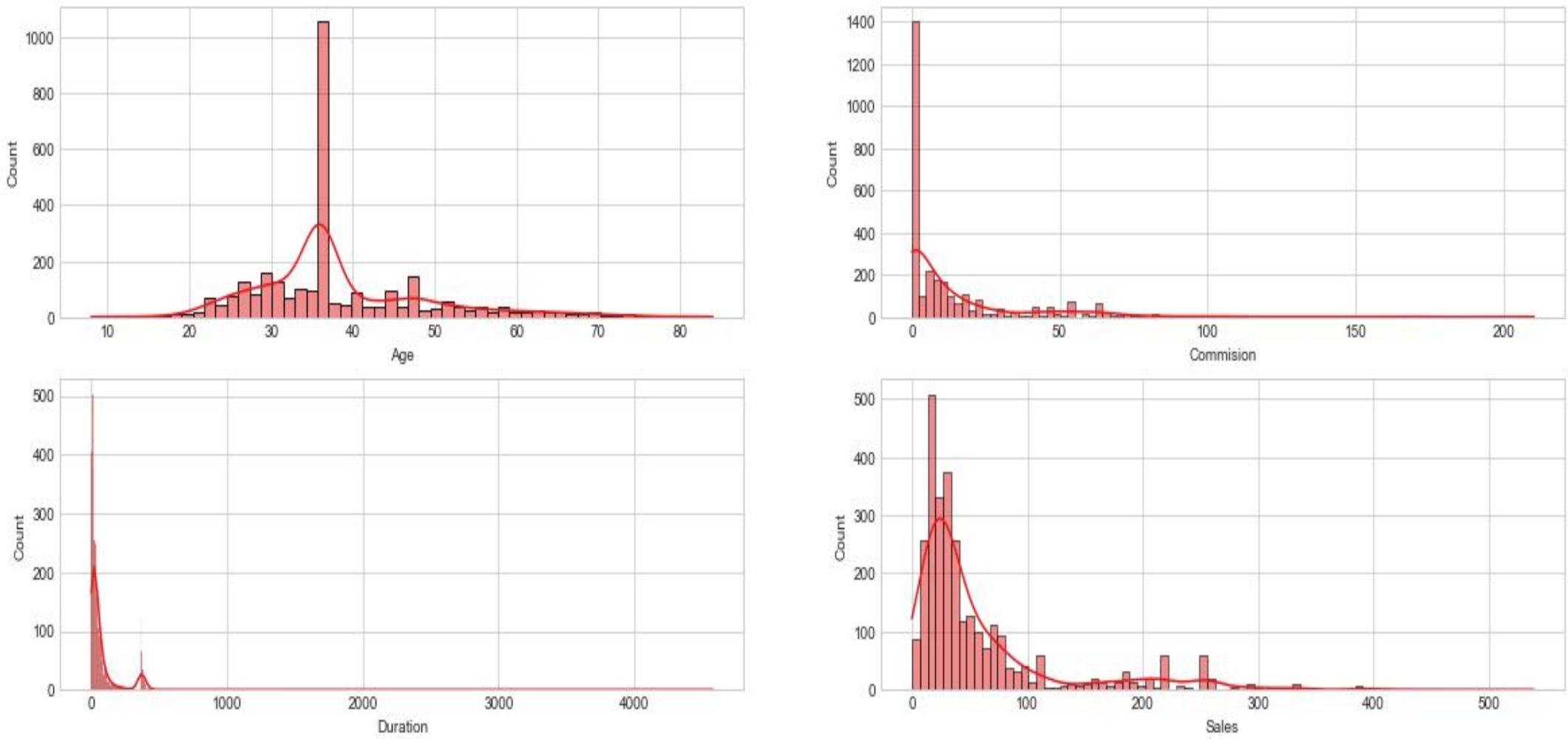

Fig 2.2

The inference from box plot can be proven true by visualizing histogram for the numerical columns.Here age column seems to be normally distributed and other columns are right skewed.

Bivariate & Multi-variate analysis

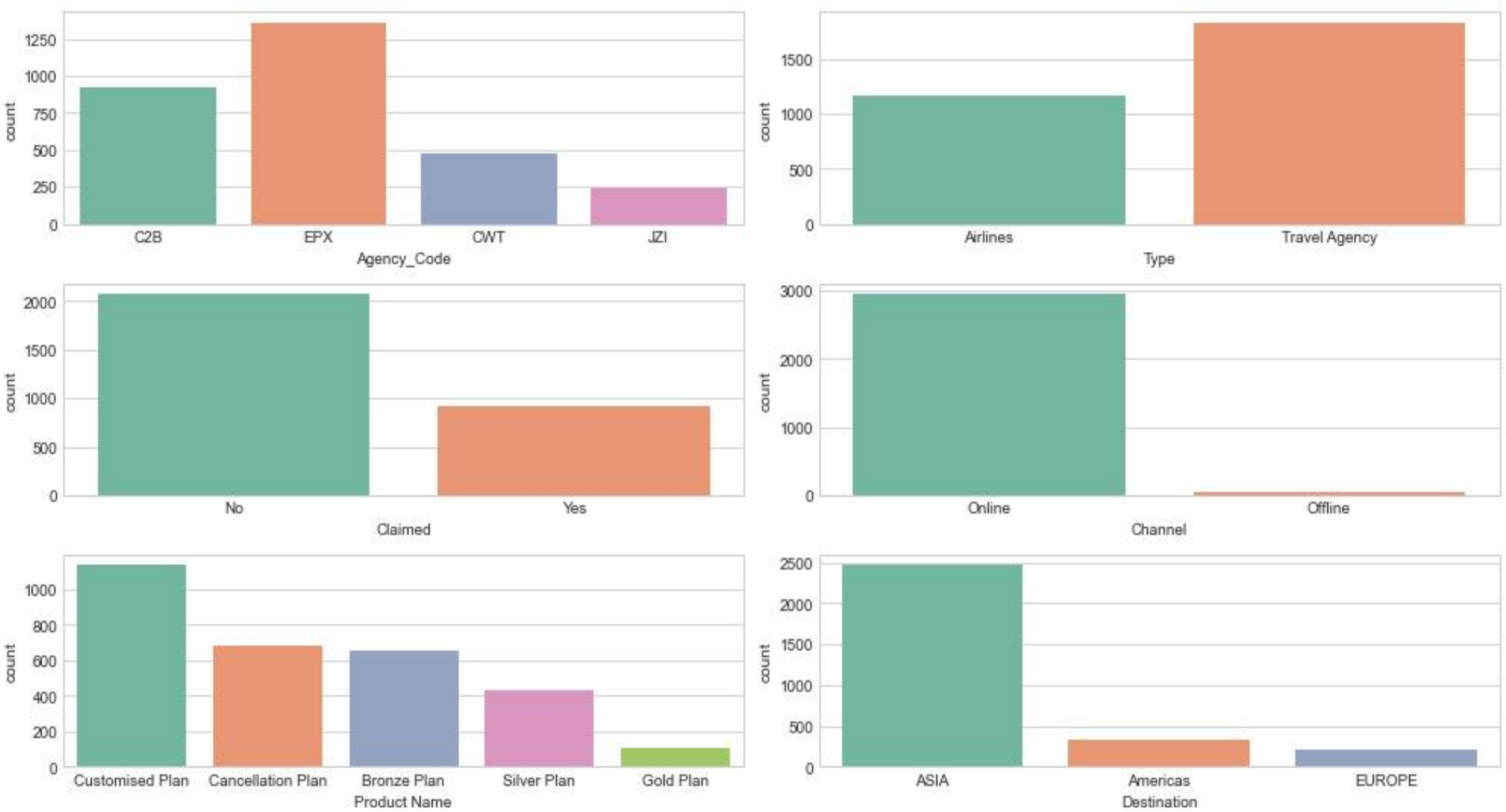

Fig 2.3

Count plot for the categorical columns are shown above.Code of tour firm is represented in agency code.EPX is the most unqiue agency code followed by C2B ,CWT & JZI.Most people rely on Travel agency than Airlines in type of tour insurance firms.It is evident from the count plot that most of the insurance are not claimed and the frequent method used by insurers to sell their products to customers is through online . There are 5 tour insurance products ,the most unique product is Customised plan.

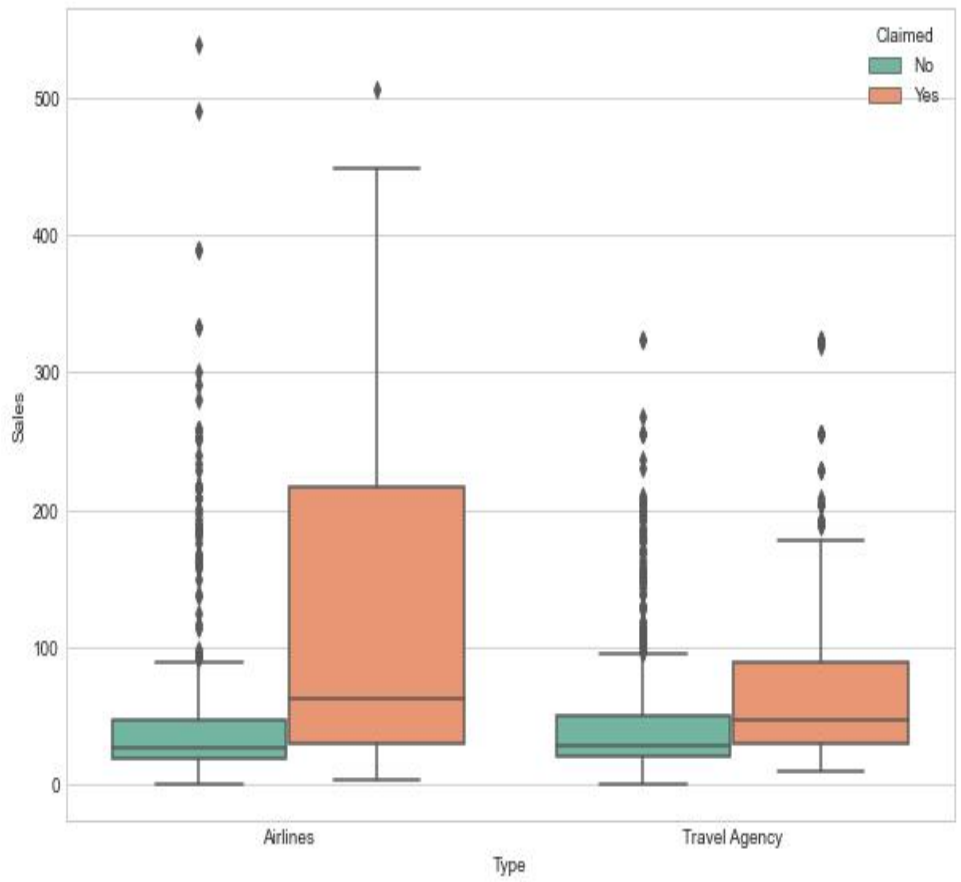

Fig 2.4

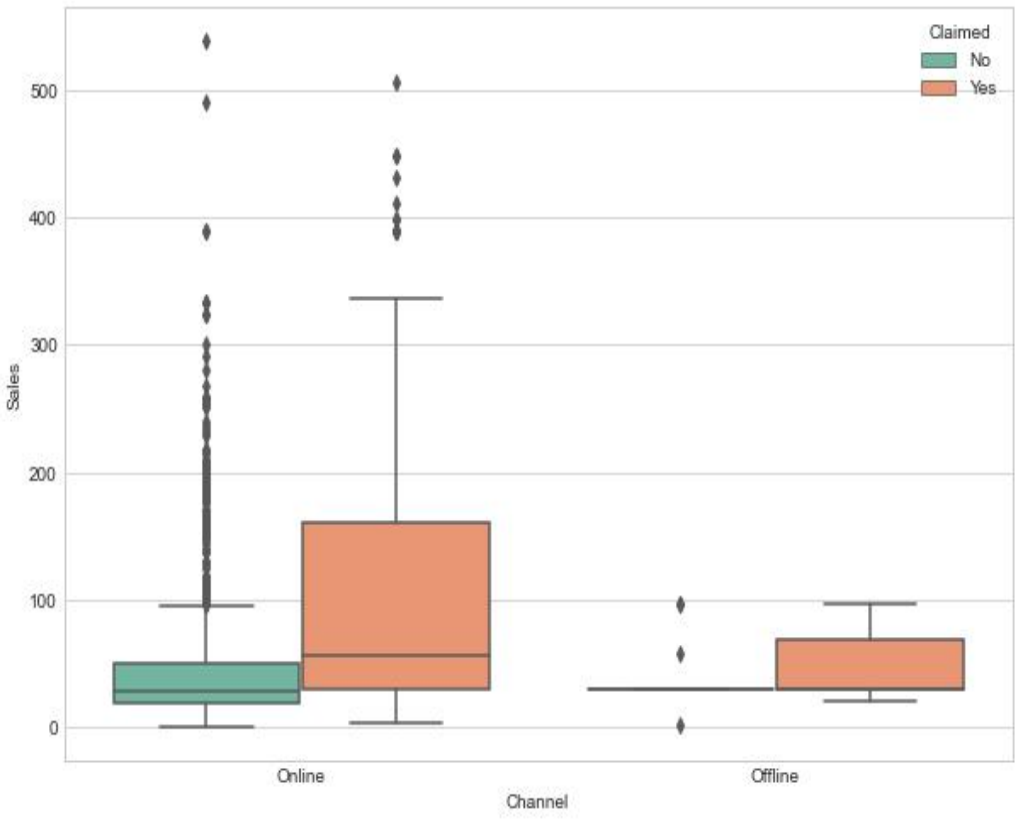

Fig 2.5

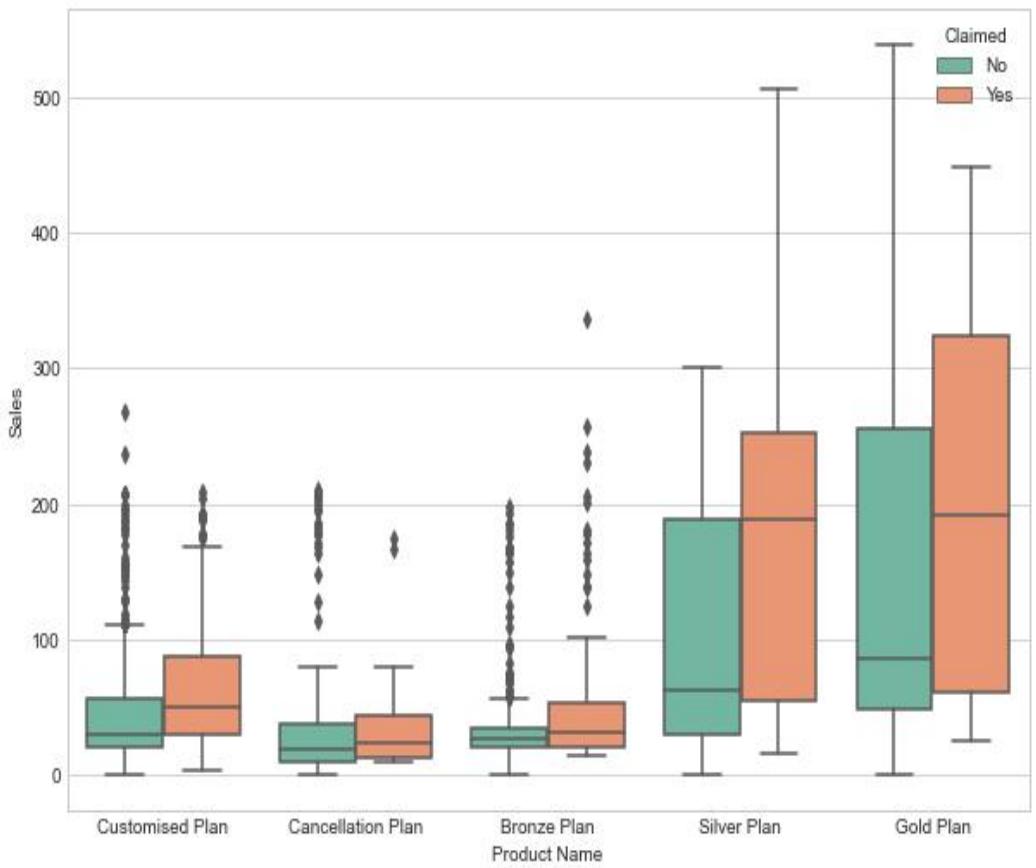

Fig 2.6

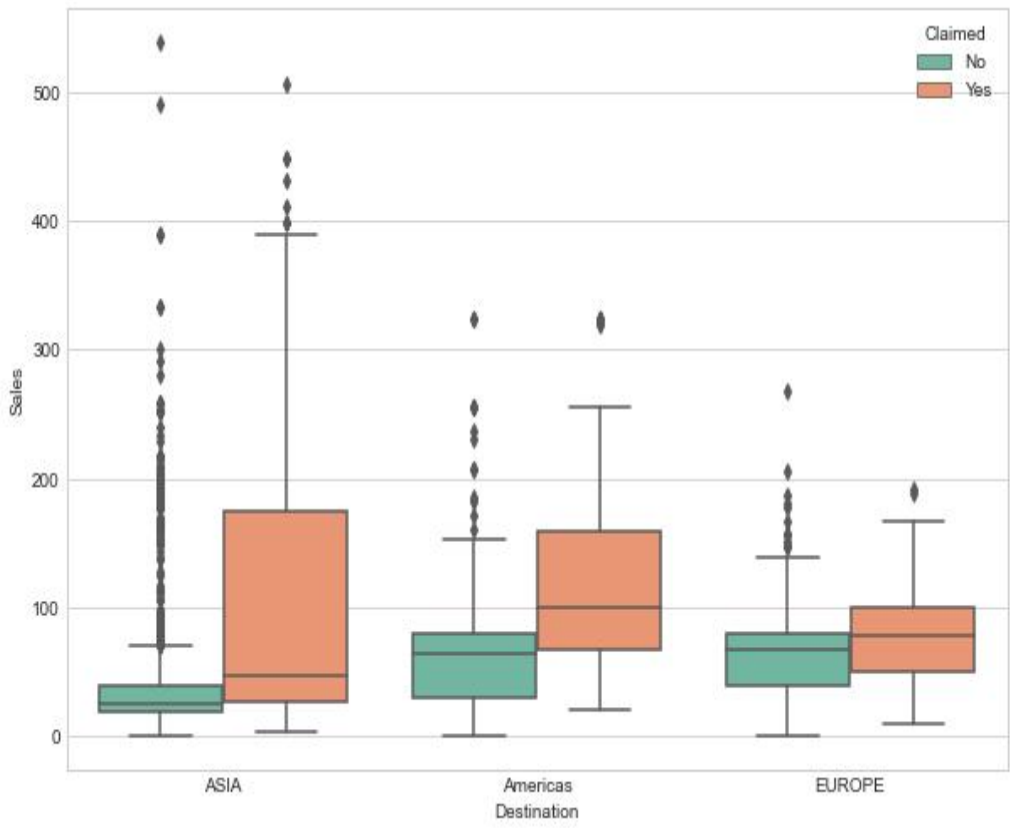

Fig 2.7

The above plots show the Type,Channel,Product name and Destination with respect to sales and the insurance claimed.

Pairplot of all the numerical columns in dataset:

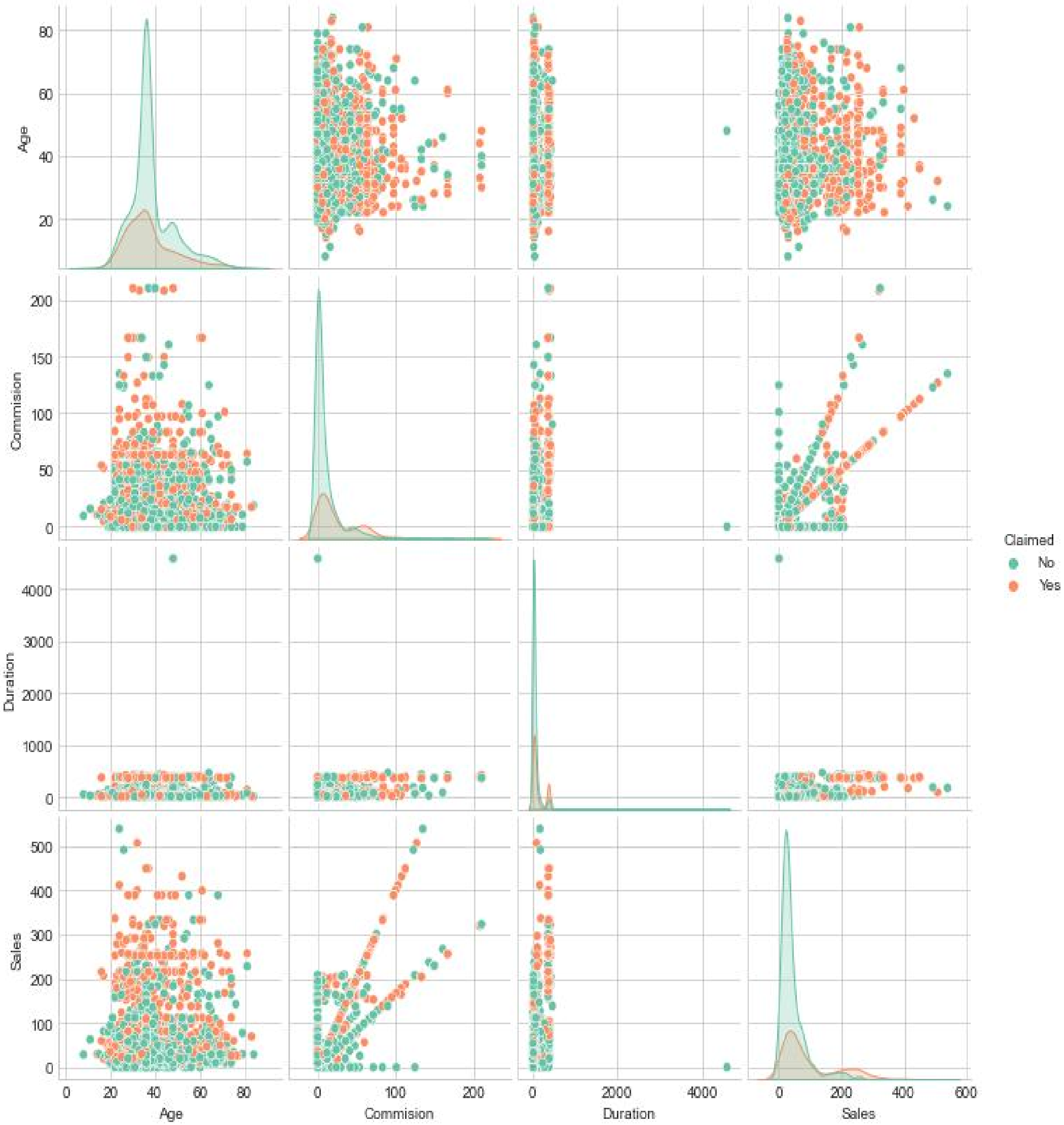

Fig 2.8

Here all the numerical columns such as Age,Commission ,Duration and Sales are plotted with respect to the claims made

Heat map to show correlation of all the numerical columns in dataset:

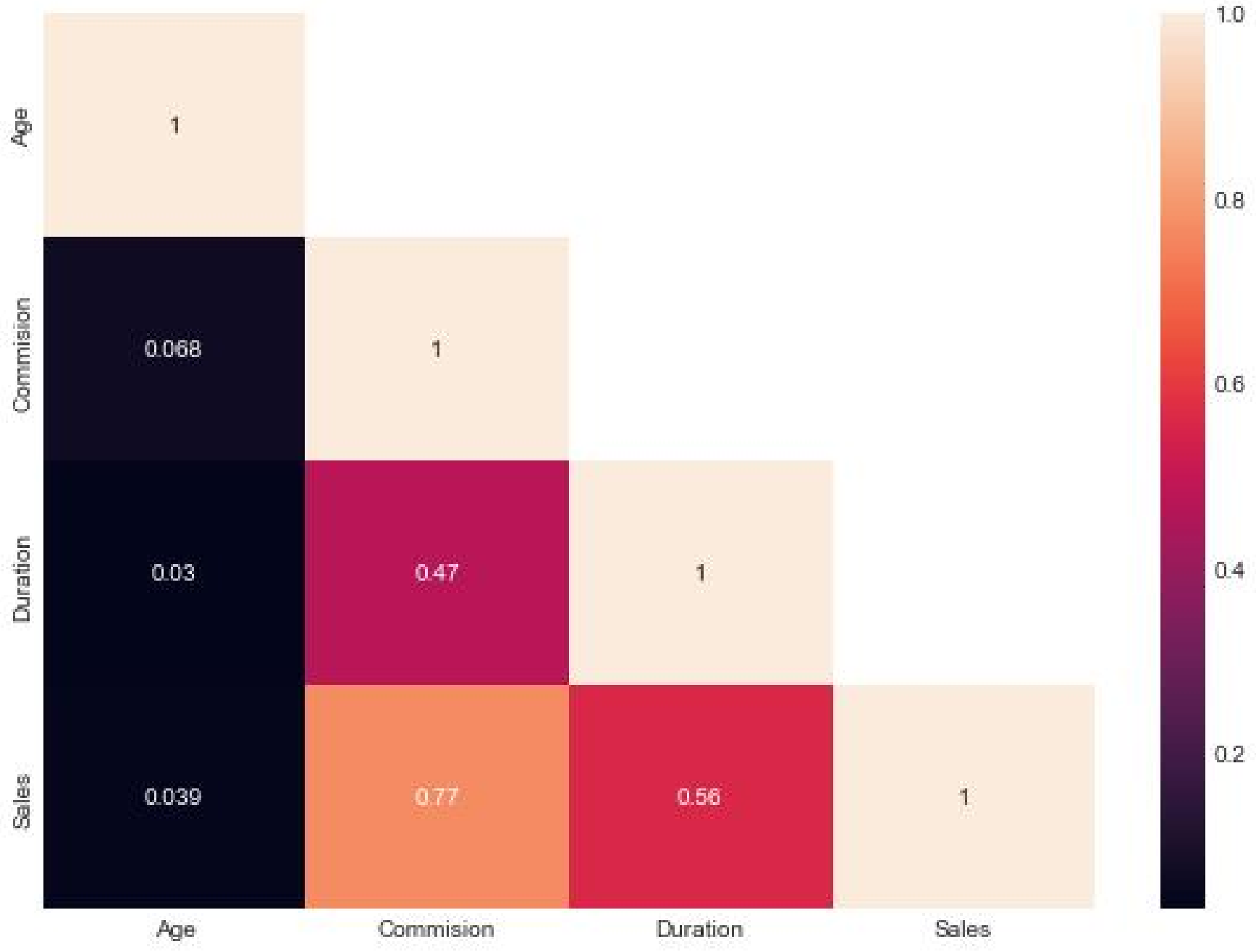

Fig 2.9

The heat map of the dataset shows that there is only low correlation between columns. 0.7 to 0.9 is considered a high correlation, 0.5 to 0.7 is moderately correlated and below 0.3 to 0.5 indicate variables that have a low correlation.Sales with respect to Commission is the only column that have moderately high correlation.

Correlation table :

	Age	Commision	Duration	Sales
Age	1.000000	0.067717	0.030425	0.039455
Commision	0.067717	1.000000	0.471389	0.766505
Duration	0.030425	0.471389	1.000000	0.558930
Sales	0.039455	0.766505	0.558930	1.000000

Table 2.3

Outlier percentage in numerical columns :

Calculated minimum for Age is 17.0
Calculated maximum for Age is 57.0
The outlier percentage in Age is 6.8 %

Calculated minimum for Commision is -25.8525
Calculated maximum for Commision is 43.09
The outlier percentage in Commision is 12.07 %

Calculated minimum for Duration is -67.0
Calculated maximum for Duration is 141.0
The outlier percentage in Duration is 12.73 %

Calculated minimum for Sales is -53.5
Calculated maximum for Sales is 142.5
The outlier percentage in Sales is 11.77 %

Skewness :

Age	1.149713
Commision	3.148858
Duration	13.784681
Sales	2.381148

Table 2.4

If the skewness is between -0.5 and 0.5, the data are fairly symmetrical.

If the skewness is between -1 and – 0.5 or between 0.5 and 1, the data are moderately skewed.

If the skewness is less than -1 or greater than 1, the data are highly skewed.

As skewness is considered we can conclude that most of the data are highly skewed .

2.2 **Data Split: Split the data into test and train(0.5 pts), build classification model CART (2.5 pts), Random Forest (2.5 pts). Object data should be converted into categorical/numerical data to fit in the models. (pd.categorical().codes(), pd.get_dummies(drop_first=True)) Data split, ratio defined for the split, train-test split should be discussed. Any reasonable split is acceptable. Use of random state is mandatory. Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Apply grid search for each model and make models on best_params. Feature importance for each model.**

Converting object to categorical or code:

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

Table 2.5

Train_test_split

Train data first five rows:

	Age	Agency_Code	Type	Commision	Channel	Duration	Sales	Product Name	Destination
1045	36	2	1	0.00	1	30	20.00	2	0
2717	36	2	1	0.00	1	139	42.00	2	1
2835	28	0	0	46.96	1	367	187.85	4	0
2913	28	0	0	12.13	1	29	48.50	4	0
959	48	1	1	18.62	1	53	49.00	3	0

Table 2.6

Test data first five rows:

	Age	Agency_Code	Type	Commision	Channel	Duration	Sales	Product Name	Destination
1957	22	1	1	28.50	1	28	75.0	0	2
2087	55	0	0	6.63	1	24	26.5	0	0
1394	29	0	0	4.00	1	33	16.0	0	0
1520	27	0	0	15.88	1	40	63.5	4	0
1098	36	2	1	0.00	1	35	27.0	1	0

Table 2.7

Target variable train data first five rows:

```
1045    0
2717    0
2835    1
2913    1
959     0
Name: Claimed, dtype: int8
```

Table 2.8

Target variable test data first five rows:

```
1957    0
2087    1
1394    1
1520    1
1098    0
Name: Claimed, dtype: int8
```

Table 2.9

Classification model using CART

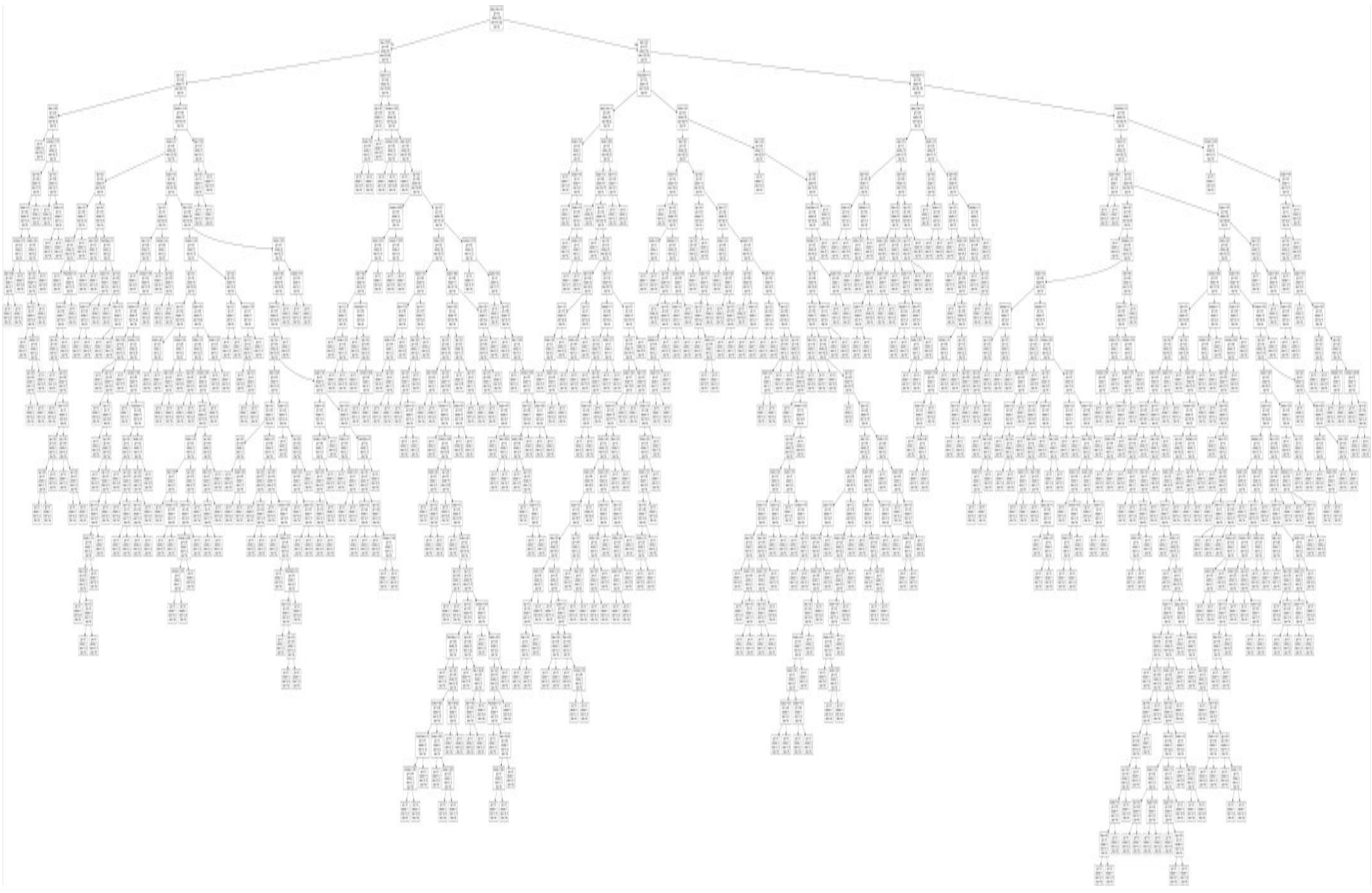

Fig 2.10

Above figure shows a fully grown decision tree. Data is split into 70% of train data and 30% of test data with random state value as 1.Here we have used gini index as criterion.The parent node splits with the variable agency code less than 0.5 which has 2100 samples and a gini index of 0.42.There are 1471 no in the root node.Here only the criterion gini index is considered ,max_depth,min_sample_leaf etc. are not considered.

To get the best model we need to prune the current decision tree with some additional parameters. After applying grid search we get the following output.

max_depth=4, min_samples_leaf=10, min_samples_split=45,cv = 3,random_state=1

These are the optimum values to build an accurate model.

While applying grid search we have given cv value of 3. Cross-validation(cv) is a re-sampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation.

Max depth of 4 is selected due to the reason that after depth 4 the tree does not produce uniform trees / branches so pruning at depth 4 makes the tree uniform and it is good for the model.

Min_sample_leaf shows that at the terminal node there are atleast 10 samples.Min_sample_split makes sure that before split that node consists of atleast 45 samples .If there are less than 45 samples the tree will not split into child nodes.All these parameters helps in build a good decision tree model.

Random_state is used to set the seed for the random generator so that we can ensure that the results that we get can be reproduced. Because of the nature of splitting the data in train and test is randomised you would get different data assigned to the train and test data unless you can control for the random factor .Random state is given as 1 here.

Important features:

	Imp
Agency_Code	0.597721
Sales	0.253110
Product Name	0.076348
Duration	0.035679
Commision	0.029747
Type	0.007395
Age	0.000000
Channel	0.000000
Destination	0.000000

Table 2.10

It is clear from the table that agency code is the most important feature,the next best feature is sales.If we closely observe the root node of the tree it can be seen that the tree first splits based on agency code.Feature Importance refers to techniques that calculate a score for all the input features for a given model .The scores simply represent the “importance” of each feature. A higher score means that the specific feature will have a larger effect on the model that is being used to predict a certain variable.

Classification model using RF

To get the best model we have to apply the following parameter and values.

max_depth=8, min_samples_leaf=5, min_samples_split=30, n_estimators=301,

oob_score=True ,random_state=1, cv=5)

These values are obtained from 672 combinations of values.

While applying grid search we have given cv value of 5. Cross-validation(cv) is a re-sampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation.

Max depth of 3 thru 10 was provided during grid search and the best depth suggested is 8.

Min_sample_leaf shows that at the terminal node there are atleast 5 samples.Min_sample_split makes sure that before split that node consists of atleast 30 samples .If there are less than 30 samples the tree will not split into child nodes.All these parameters are selected from wide range of values and helps in build a good model.

In Random Forest Classifier and Regression, random_state controls the randomness of the bootstrapping of the samples used when building trees and the sampling of the features to consider when looking for the best split at each node.Random state is given as 1 here.

When we create a bootstrapped dataset, ~1/3 of the original data does not end up in the boot strapped dataset.This is called out-of-bag dataset.To create oob score we need to pass the oob_score parameter as true in model.

n_estimators represents number of decision trees in the random forest mode.By default it is 10.We have used 501 and 301 in grid search and the best number of trees is estimated as 301.

Important features:

	Imp
Agency_Code	0.234423
Product Name	0.197882
Sales	0.175133
Commision	0.143833
Duration	0.099229
Age	0.071230
Type	0.063419
Destination	0.011780
Channel	0.003072

Table 2.11

Feature Importance refers to techniques that calculate a score for all the input features for a given model .The scores simply represent the “importance” of each feature. A higher score means that the specific feature will have a larger effect on the model that is being used to predict a certain variable.

2.3 **Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy (1 pts), Confusion Matrix (2 pts), Plot ROC curve and get ROC_AUC score for each model (2 pts), Make classification reports for each model. Write inferences on each model (2 pts). Calculate Train and Test Accuracies for each model. Comment on the validness of models (overfitting or underfitting) Build confusion matrix for each model. Comment on the positive class in hand. Must clearly show obs/pred in row/col Plot roc_curve for each model. Calculate roc_auc_score for each model. Comment on the above calculated scores and plots. Build classification reports for each model. Comment on f1 score, precision and recall, which one is important here.**

We will separately calculate performance metrics for CART and RF along with the predicted test and train values.

CART

Train data predicted-Decision tree

```
array([0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0,
       1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1,
       0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1,
       0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0,
       0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 1], dtype=int8)
```

Fig 2.11

Test data predicted-Decision tree

```
array([0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0,
       1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1,
       0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0],
dtype=int8)
```

Fig 2.12

Train data Accuracy score,Confusion matrix ,roc_auc_score,f1 score, precision and recall

Accuracy score : 0.792857

Confusion matrix :

	PREDICTION	
ACTUAL	**0** (PREDICTED NEGATIVE,NOT CLAIMED)	**1** (PREDICTED POSITIVE,CLAIMED)
0 (ACTUAL NEGATIVE,NOT CLAIMED)	1263 (TN)	208 (FP)
1 (ACTUAL POSTIVE,CLAIMED)	227 (FN)	408 (TP)

Table 2.12

Classification report :

	precision	Recall	F1-score	support
0	0.85	0.86	0.85	1471
1	0.66	0.64	0.65	629
accuracy			0.79	2100
macro avg	0.75	0.75	0.75	2100
weighted avg	0.79	0.79	0.79	2100

Table 2.13

ROC_AUC_SCORE : 0.831

ROC_CURVE PLOT :

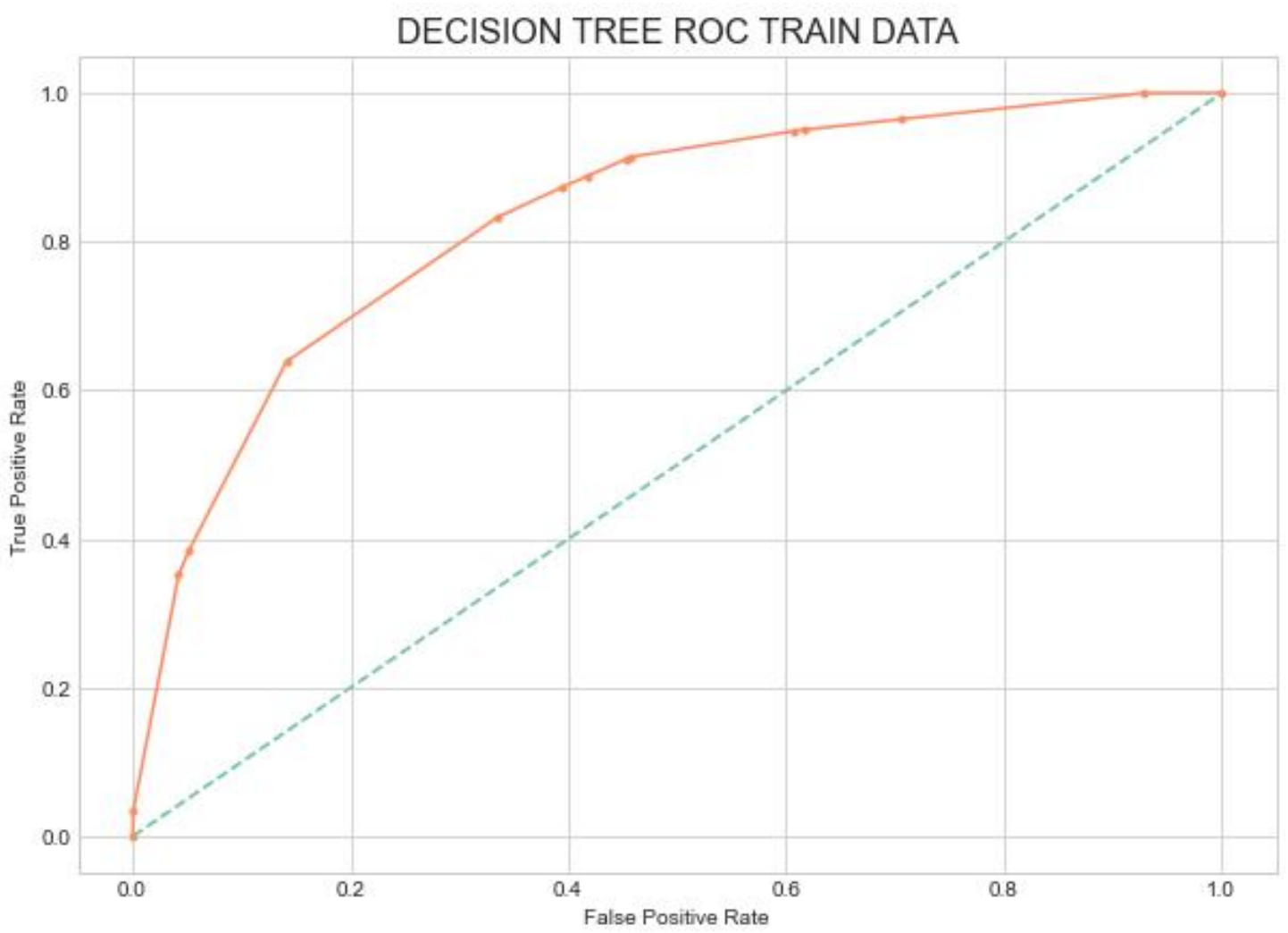

Fig 2.13

Test data Accuracy score,Confusion matrix ,roc_auc_score,f1 score, precision and recall

Accuracy score : 0.78111

Confusion matrix :

	PREDICTION	
ACTUAL	**0** (PREDICTED NEGATIVE,NOT CLAIMED)	**1** (PREDICTED POSITIVE,CLAIMED)
0 (ACTUAL NEGATIVE,NOT CLAIMED)	536 (TN)	69 (FP)
1 (ACTUAL POSTIVE,CLAIMED)	128 (FN)	167 (TP)

Table 2.14

Classification report :

	precision	Recall	F1-score	support
0	0.81	0.89	0.84	605
1	0.71	0.57	0.63	295
accuracy			0.78	900
macro avg	0.76	0.73	0.74	900
weighted avg	0.77	0.78	0.77	900

Table 2.15

ROC_AUC_SCORE : 0.795

ROC_CURVE PLOT :

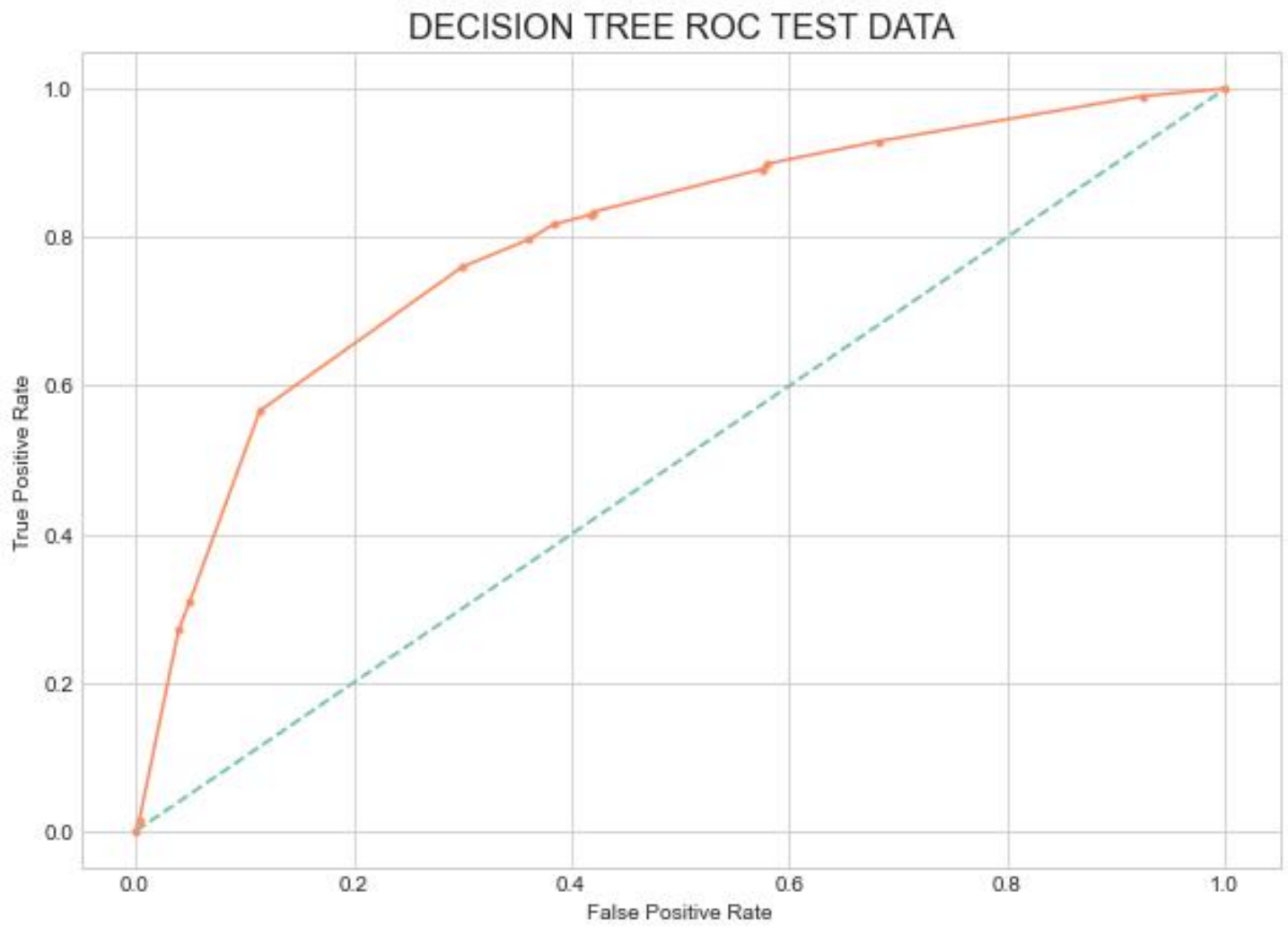

Fig 2.14

Accuracy is the number of correct predictions made divided by the total number of predictions made.**Train data has 79% and test data has 78% accuracy score**.Lesser the false predictions more the accuracy. An accuracy measure of anything between 70%-90% is not only ideal, it's realistic.

Confusion matrix is a 2x2 tabular structure reflecting the performance of the model in four blocks.True positive(TP) and True negative(TN) are the correct predictions.False positive(FP) and False negative(FN) are the incorrect predictions.Lesser the false predictions more the accuracy.
There are four ways to check if the predictions are right or wrong:
TN / True Negative: the case was negative and predicted negative.
TP / True Positive: the case was positive and predicted positive.
FN / False Negative: the case was positive but predicted negative.
FP / False Positive: the case was negative but predicted positive.
From confusion matrix it is clear that there are more number of values in true negative(TN) for both train and test data.This indicate the models ability to predict actual zero as zero in other words it can be established that non claimed insurance can be predicted correctly with this model rather than claimed insurance.

A Classification report is used to measure the quality of predictions.rue Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report
Precision : The precision tells us the accuracy of positive predictions.Among the points identified as positive by the model,how many are really positive this is the objective of precision .High precision means not many True values were predicted as False.Here we need to check precision for the claimed(1 in confusion matrix).**Train data shows there is 66% precision and test data shows 71%.**

The **recall**, also named sensitivity, tells us the fraction of correctly identified positive predictions. What fraction of the True predictions were actually True recall finds this property. High recall means Predicted most True values correctly.**Here 64% recall value is seen for train data and 57% for the test data.**

The f**1-score**, or F measure, measures precision and recall at the same time by finding the harmonic mean of the two values.**Per classification report we have 65% and 63% of f1-score in train and test data respectively**.

AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes.The area under the ROC curve (AUC) results were considered excellent for AUC values between 0.9-1, good for AUC values between 0.8-0.9, fair for AUC values between 0.7-0.8, poor for AUC values between 0.6-0.7 and failed for AUC values between 0.5-0.6.**Here we have AUC score of 0.78 for train data and 0.79 for test data which implies the values are fair**.The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

F1 score is a weighted average of precision and recall. As we know in precision and in recall there is false positive and false negative f1-score consider both of them.So we can conclude that f1 score is more important than recall and precision.

RANDOM FOREST

Train data predicted-RF

```
array([0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0,
       1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0,
       1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0,
       0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1,
       0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
       0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0,
       0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1,
       0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1,
       0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0], dtype=int8)
```

Fig 2.15

Test data predicted-RF

```
array([0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
       1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0],
      dtype=int8)
```

Fig 2.16

Train data Accuracy score,Confusion matrix ,roc_auc_score,f1 score, precision and recall

Accuracy score : 0.8257

Oob(Out of bag score) score : 0.7928

Confusion matrix :

	PREDICTION	
ACTUAL	**0** (PREDICTED NEGATIVE,NOT CLAIMED)	**1** (PREDICTED POSITIVE,CLAIMED)
0 (ACTUAL NEGATIVE,NOT CLAIMED)	1343 (TN)	128 (FP)
1 (ACTUAL POSTIVE,CLAIMED)	238 (FN)	391 (TP)

Table 2.16

Classification report :

	precision	Recall	F1-score	support
0	0.85	0.91	0.88	1471
1	0.75	0.62	0.68	629
accuracy			0.83	2100
macro avg	0.80	0.77	0.78	2100
weighted avg	0.82	0.83	082	2100

Table 2.17

ROC_AUC_SCORE : 0.877

ROC_CURVE PLOT :

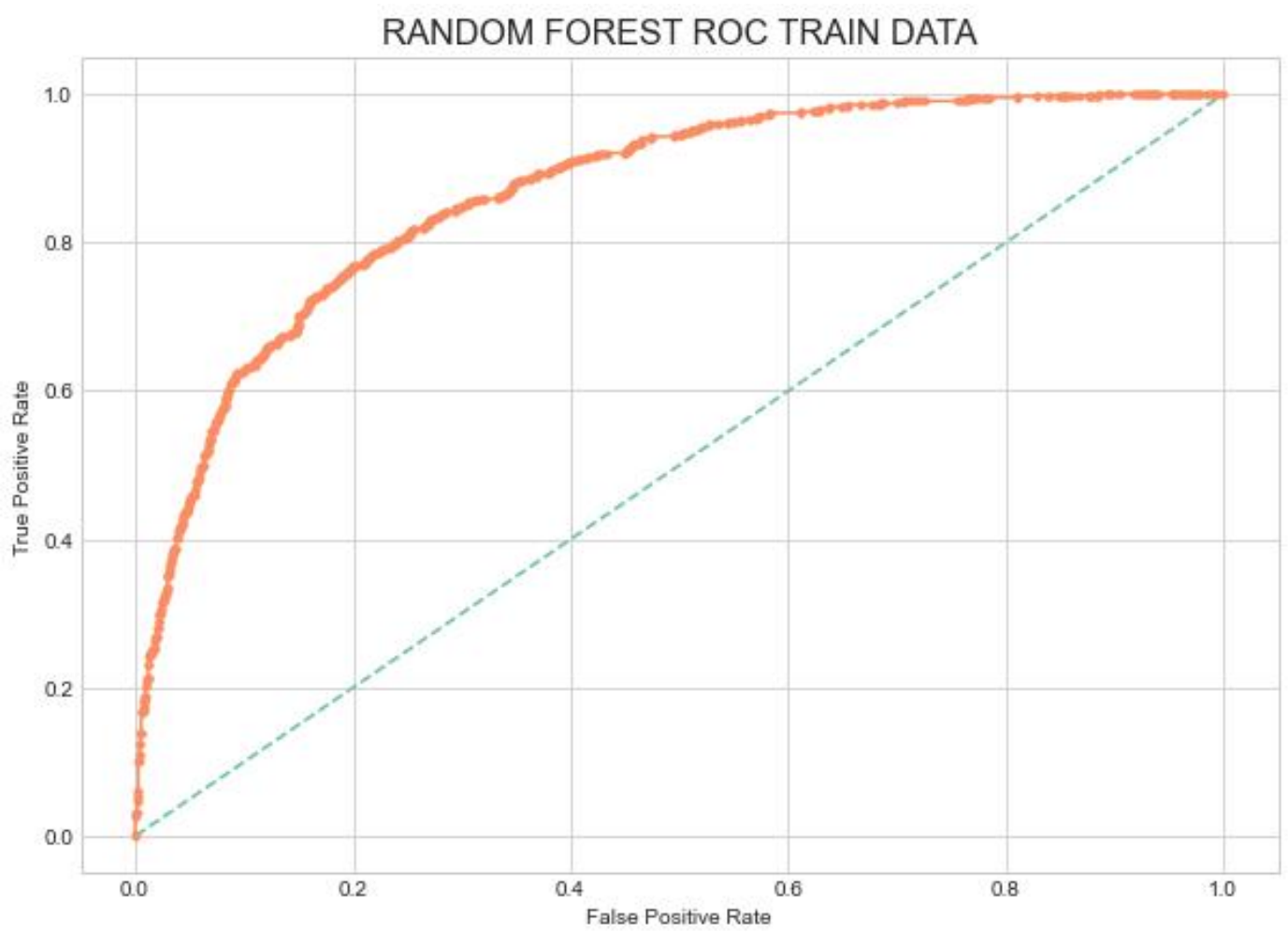

Fig 2.17

Test data Accuracy score,Confusion matrix ,roc_auc_score,f1 score, precision and recall

Accuracy score : 0.7777

Confusion matrix :

	PREDICTION	
ACTUAL	**0** (PREDICTED NEGATIVE,NOT CLAIMED)	**1** (PREDICTED POSITIVE,CLAIMED)
0 (ACTUAL NEGATIVE,NOT CLAIMED)	553 (TN)	52 (FP)
1 (ACTUAL POSTIVE,CLAIMED)	148 (FN)	147 (TP)

Table 2.18

Classification report :

	precision	Recall	F1-score	support
0	0.79	0.91	0.85	605
1	0.74	0.50	0.60	295
accuracy			0.78	900
macro avg	0.76	0.71	0.72	900
weighted avg	0.77	0.78	0.76	900

Table 2.19

ROC_AUC_SCORE : 0.823

ROC_CURVE PLOT :

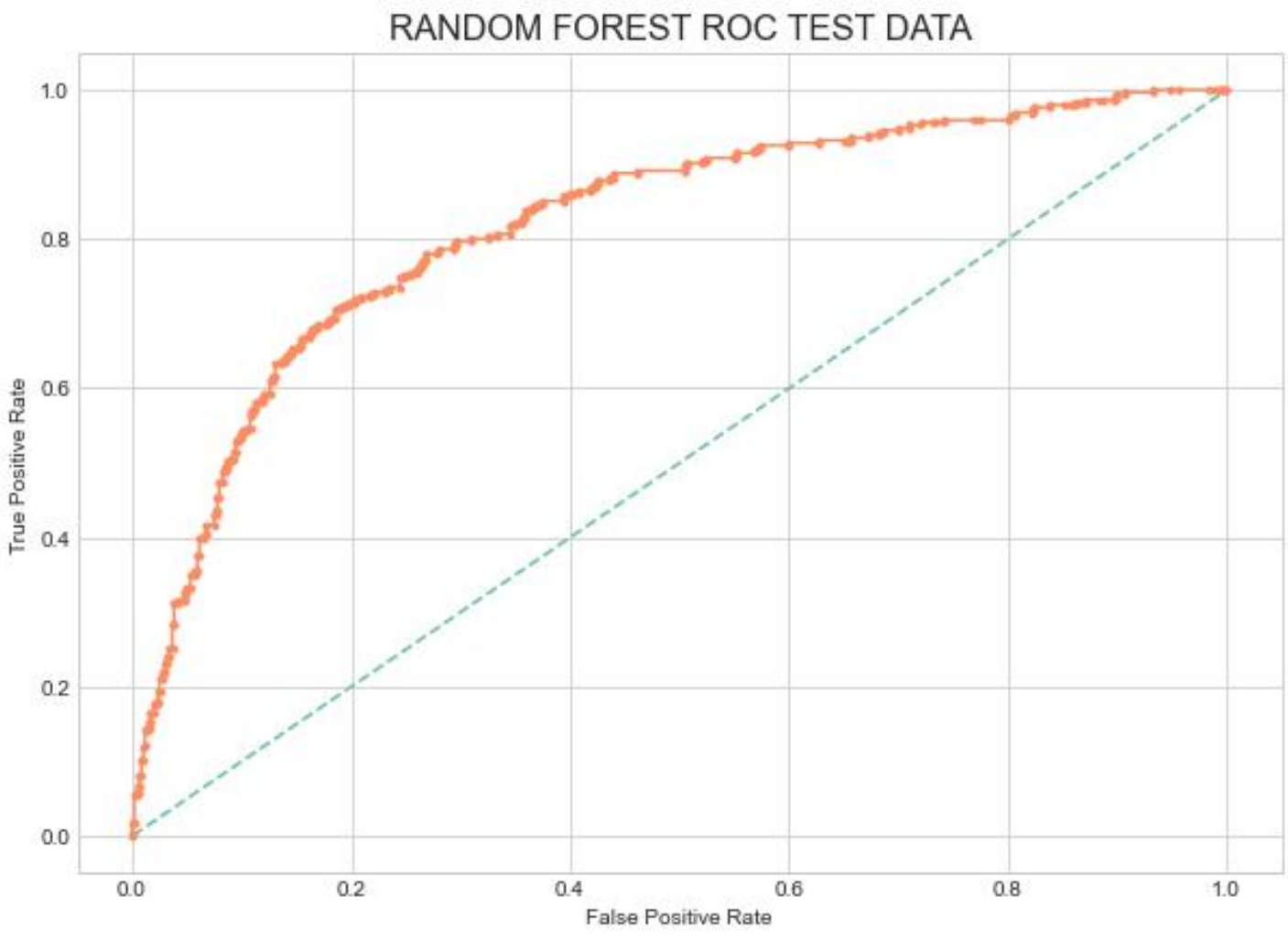

Fig 2.18

Accuracy is the number of correct predictions made divided by the total number of predictions made.**Train data has 82% and test data has 77% accuracy score**.Lesser the false predictions more the accuracy. An accuracy measure of anything between 70%-90% is not only ideal, it's realistic.

Confusion matrix is a 2x2 tabular structure reflecting the performance of the model in four blocks.True positive(TP) and True negative(TN) are the correct predictions.False positive(FP) and False negative(FN) are the incorrect predictions.Lesser the false predictions more the accuracy.
From confusion matrix it is clear that there are more number of values in true negative(TN) for both train and test data.This indicate the models ability to predict actual zero as zero in other words it can be established that non claimed insurance can be predicted correctly with this model rather than claimed insurance.

A Classification report is used to measure the quality of predictions.True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report
Precision : The precision tells us the accuracy of positive predictions.Among the points identified as positive by the model,how many are really positive this is the objective of precision .High precision means not many True values were predicted as False.Here we need to check precision for the claimed(1 in confusion matrix).**Train data shows there is 75% precision and test data shows 74%.**

The **recall**, also named sensitivity, tells us the fraction of correctly identified positive predictions.
What fraction of the True predictions were actually True recall finds this property. High recall means Predicted most True values correctly.**Here 62% recall value is seen for train data and 50% for the test data.**

The f**1-score**, or F measure, measures precision and recall at the same time by finding the harmonic mean of the two values.**Per classification report we have 68% and 60% of f1-score in train and test data respectively**.

AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes.The area under the ROC curve (AUC) results were considered excellent for AUC values between 0.9-1, good for AUC values between 0.8-0.9, fair for AUC values between 0.7-0.8, poor for AUC values between

0.6-0.7 and failed for AUC values between 0.5-0.6.**Here we have AUC score of 0.87 for train data and 0.82 for test data which implies the values are fair**.The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

The OOB_score is computed as the number of correctly predicted rows from the out-of-bag sample.**Oob score is 0.792**

F1 score is a weighted average of precision and recall. As we know in precision and in recall there is false positive and false negative f1-score consider both of them.So we can conclude that f1 score is more important than recall and precision.

Over-fitting and Under-fitting

Over fitting occurs when our decision tree tries to cover all the data points or more than the required data points present in the given dataset. First stage of decision tree model was to create a over fitted model . Then all the desired features are considered and pruning is done.Thus the model becomes less biased. When we add trees to the Random Forest then the tendency to over fitting should decrease.

Under fit is due to the fact that the minimum requirement of splitting a node is so high that there are no significant splits observed.Here both CART & RF models are not over fitted or under fitted.

2.4 **Final Model - Compare all models on the basis of the performance metrics in a structured tabular manner (2.5 pts). Describe on which model is best/optimized (1.5 pts). A table containing all the values of accuracies, precision, recall, auc_roc_score, f1 score. Comparison between the different models(final) on the basis of above table values. After comparison which model suits the best for the problem in hand on the basis of different measures. Comment on the final model.**

CART & RF PERFORMANCE METRICS TABLE

	CART - TRAIN DATA	CART - TEST DATA	RF - TRAIN DATA	RF - TEST DATA
Precision	0.66	0.71	0.75	0.74
Recall	0.64	0.57	0.62	0.50
F1-score	0.65	0.63	0.68	0.60
Accuracy	0.79	0.78	0.83	0.79
AUC	0.83	0.79	0.87	0.82

Table 2.20

Here we have created a table based on CART & RF train - test data.The table includes precision,recall,f1-score,accuracy and roc_auc_score.We have described each of these values in previous question.

Best /Optimized model:

When comparing all the values across different models it is evident that RF model is the best model. Here the train and test data produces almost the same output.Both decision tree and random forest produces a good model but some of the important features (like accuracy,auc etc) are best in random forest model.So we choose random forest as the best model.

Conclusion :

- Here, we can see that even when we try other values for the hyper parameters, the model performance is not improving much.
- The model is more useful in predicting 0 (ie. not claimed) as the values seen in classification report for 0 is much higher than 1 (ie. claimed)

2.5 Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective. There should be at least 3-4 Recommendations and insights in total. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks should only be allotted if the recommendations are correct and business specific.

The insurance firm have a claim of 30.8% per the given data set.We will see the important factors that contributed to the claim rate.

Already important features effecting the claim is shown in random forest table 2.11 .The most important feature that effects claim rate is agency code.

Table below shows the claim with respect to agency code,product name, type, destination and channel

Agency_Code	Claimed	
C2B	Yes	560
	No	364
CWT	No	331
	Yes	141
EPX	No	1172
	Yes	193
JZI	No	209
	Yes	30

Name: Claimed, dtype: int64

Table 2.21

Product Name	Claimed	
Bronze Plan	No	399
	Yes	251
Cancellation Plan	No	635
	Yes	43
Customised Plan	No	882
	Yes	254
Gold Plan	Yes	70
	No	39
Silver Plan	Yes	306
	No	121

Name: Claimed, dtype: int64

Table 2.22

Type	Claimed	
Airlines	Yes	590
	No	573
Travel Agency	No	1503
	Yes	334

Name: Claimed, dtype: int64

Table 2.23

Destination	Claimed	
ASIA	No	1691
	Yes	774
Americas	No	232
	Yes	88
EUROPE	No	153
	Yes	62

Name: Claimed, dtype: int64

Table 2.24

Channel	Claimed	
Offline	No	29
	Yes	17
Online	No	2047
	Yes	907

Name: Claimed, dtype: int64

Table 2.25

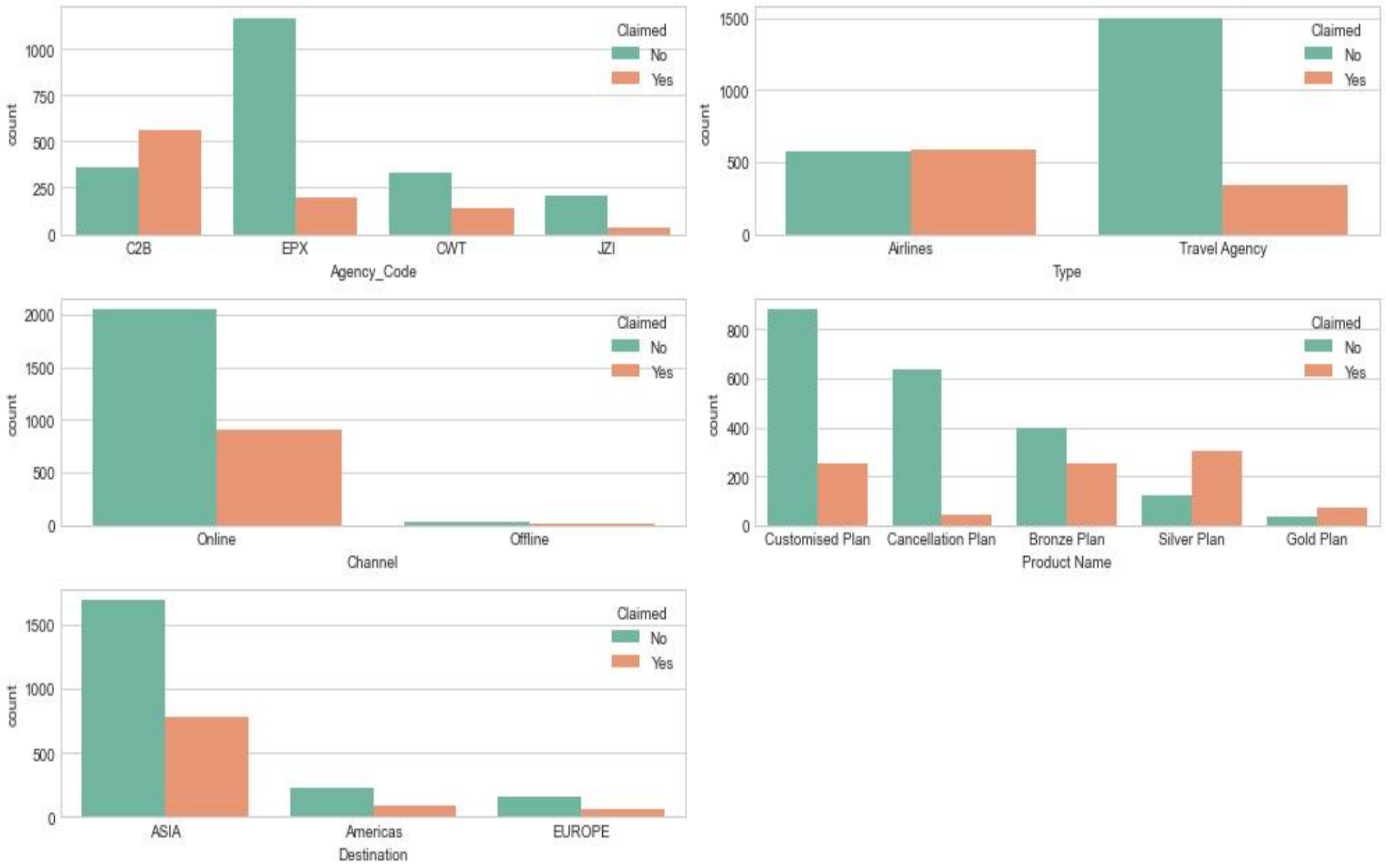

Fig 2.19

The tables shown can be graphically represented as show in FIg. 2.19

Most customers rely on EPX tour firm followed by C2B ,CWT and JZI. In terms of claims EPX and JZI tour insurance agency is facing lower claim frequency ,C2B is facing higher claim frequency.

Recommendation:

- JZI & CWT must improve the number of customers.

- Even though C2B has a high number of customers compared to the agencies above they should rectify the high claim frequency issue. They can resolve their issues following the model of EPX agency which has a high number of customers and low claim frequency.

There are two types of tour insurance firms they are airlines and travel agency.Most customers prefer travel agency rather than airlines.Also claims are more in airlines.

Recommendation:

- Airlines should make sure the claim frequency can be lessened, and they should find the reason for higher claim frequency

Majority of customers use online channel for insurance purchase and most of the claims are through the same channel .Even though online channel has large number of customers offline channel has the highest claim frequency.Offline insurance claim has a value of 36.95% and online claims at 30.70%.

Recommendation:

- The claim percentage clearly shows that there is a clear distinction between the number of customers and the claim frequency percentage.Offline insurance providers must check there process and find the reason behind increased claim frequency.

There are 5 product the tour insurance offers namely customised plan,cancellation plan,bronze plan ,silver plan and gold plan.Customised plan is the most favourite among customers followed by cancellation plan.bronze plan,silver plan and gold plan.It is clear from the graph that claim frequency for gold and silver plans are higher than usual.Cancellation plain has the least claim frequency .

Recommendation:

- Improve the silver plan and gold plan reduce the claim frequency

As per the feature importance table destination and channel are the least effecting factors for claim frequency.

To correctly predict the claim frequency more data is required.The model created based on current data focus more on non claim frequency than claimed frequency.