DATA MINING PROJECT REPORT

Vinyas Shreedhar

PGP-DSBA Online

May'21

Date: 18/09/2021

Table of Contents

List of Figures

List of Tables

Problem 1: Clustering

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

Introduction

The purpose of this whole exercise is to explore the dataset. Do the exploratory data analysis. Explore the dataset using central tendency and other parameters. The data consists of activities of different customers based on their credit card usage. We will perform exploratory data analysis to understand what the given data has to say and then use clustering techniques to develop a customer segmentation so that the bank can give promotional offers to its customers based on the clusters we have identified.

Data Dictionary for Market Segmentation

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

1.1 Read the data 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.

Sample of the Dataset:

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

Table 1. Dataset sample

Dataset has 7 variables with different aspects of credit card usage. Based on these aspects or information the credit card spending is defined.

Exploratory Data Analysis

Information on 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. Info on the Dataset

There are total 210 rows and 7 columns in the dataset. Out of 7, all columns are of float data type. We can see that there are NO missing values in the dataset.

Descriptive Statistics

Summary statistics or 5-point summary helps us to understand the Interquartile Range like minimum, maximum, 25th, 50th and 75th percentiles, mean or average, standard deviation and count of data observations etc. The most popular descriptive statistics are Measures of Central Tendency which are Mean, Median and Mode which are fundamental to any data analysis.

	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 3. Descriptive Statistics

From the above table below are the observations.

1. Spending which is the target variable looks like it's normally distributed as we can see that mean and median are same.
2. advance_payments also seems to be normally distributed. This variable might be of use as it shows that customers are paying the amount in advance which is timely payment for the bank.
3. The average probability_of_full_payment is 87.10%. Hence we need to analyse further to see the rest of the customers who fall under 13% who have not done the payment in full. This variable is normally distributed.
4. Minimum current_balance held by customer is 4899.00.
5. credit_limit of customers range between 26300.00 to 40330.00. The average credit_limit of customers is 32586.05.
6. The minimum of min_payment_amt paid is 76.51. The maximum of min_payment_amt paid is 845.60. This suggests the data is widely spread for this variable and might have outliers. Also looks like normally distributed.
7. The average of max_spent_in_single_shopping is 5408.07. The maximum of max_spent_in_single_shopping is 6550.00.

Univariate Analysis

Boxplots

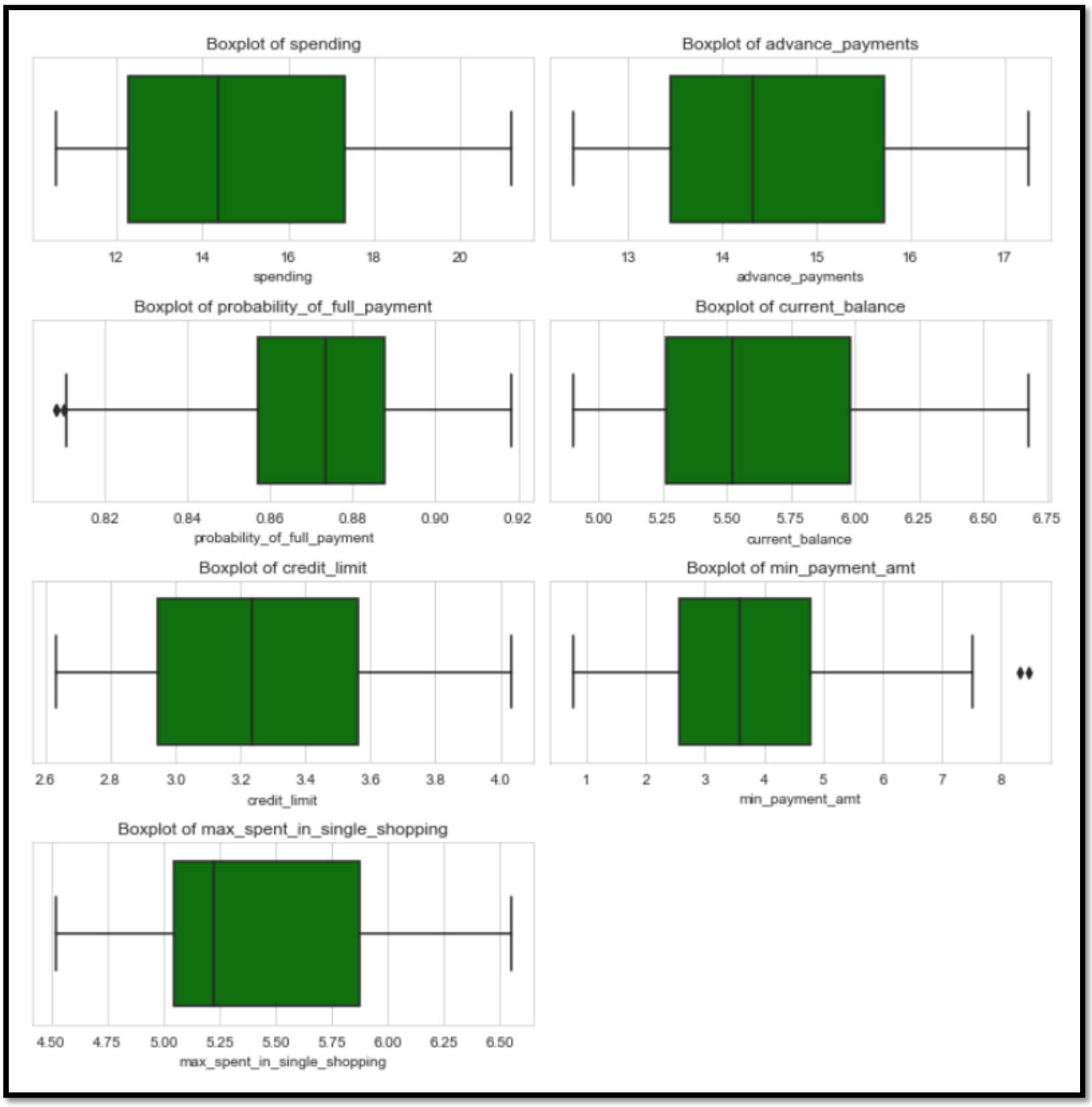

Figure 1. Boxplot of each variable

From the above boxplots, we can see that there are no outliers for all the variables except min_payment_amt. We can infer that there is a segment of customers with high credit limit who made expensive purchases and hence had to pay higher minimum payment amount which is causing the outlier.

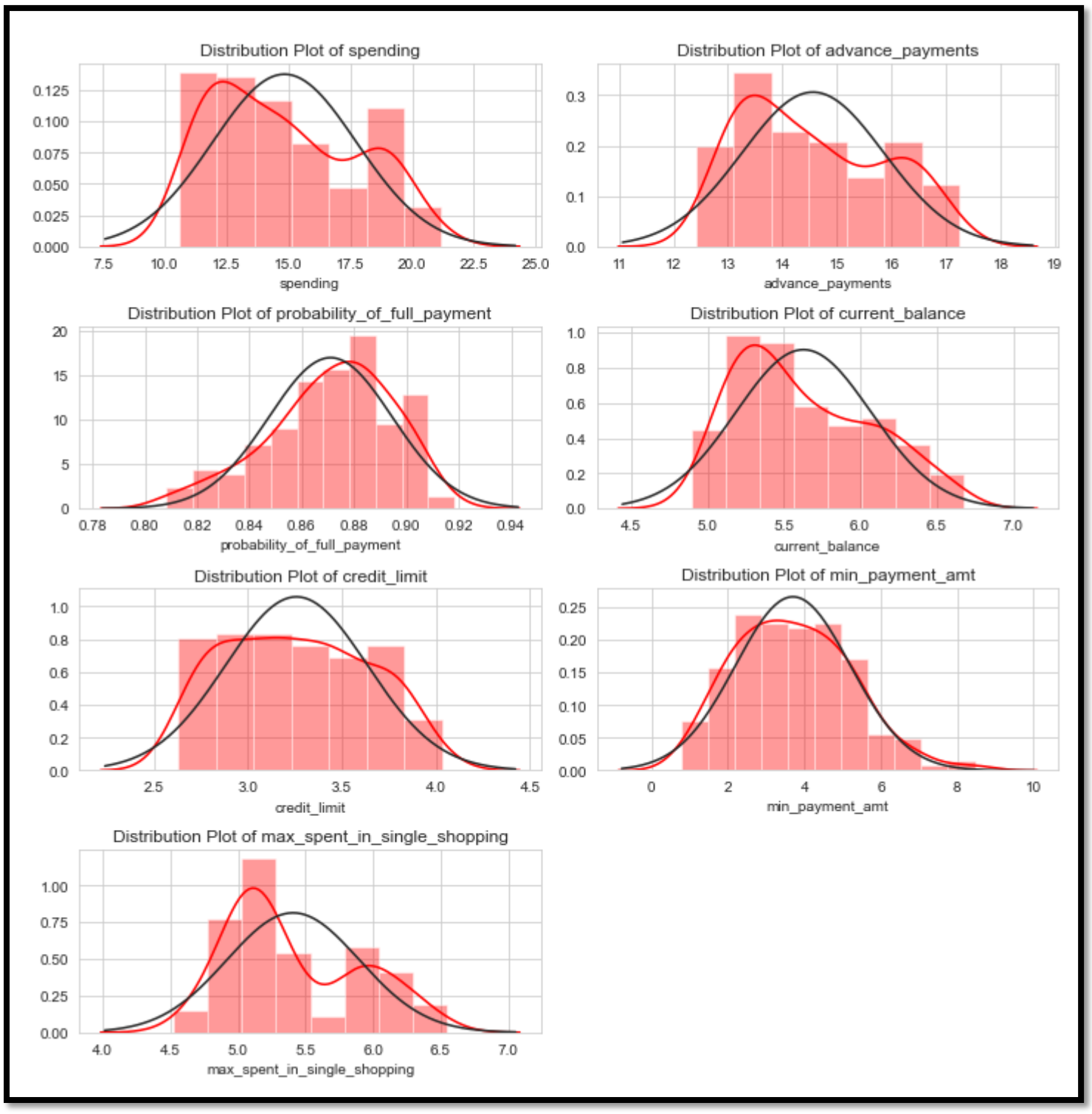

Figure 2. Distribution Plot of each variable

Skewness and Kurtosis

According to Wikipedia, "skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The skewness value can be positive, zero, negative, or undefined." For normally distributed data, the skewness should be about zero. For unimodal continuous distributions, a skewness value greater than 0 means that there is more weight in the right tail of the distribution.

According to Wikipedia, "kurtosis is a measure of the "tailedness" of the probability distribution of a real-valued random variable. Like skewness, kurtosis describes the shape of a probability distribution." In Fisher's definiton, the kurtosis of the normal distribution is zero. The distribution with a higher kurtosis has a heavier tail.

```
Skewness of spending: 0.4
Kurtosis of spending: -1.08
Skewness of advance_payments: 0.39
Kurtosis of advance_payments: -1.11
Skewness of probability_of_full_payment: -0.54
Kurtosis of probability_of_full_payment: -0.14
Skewness of current_balance: 0.53
Kurtosis of current_balance: -0.79
Skewness of credit_limit: 0.13
Kurtosis of credit_limit: -1.1
Skewness of min_payment_amt: 0.4
Kurtosis of min_payment_amt: -0.07
Skewness of max_spent_in_single_shopping: 0.56
Kurtosis of max_spent_in_single_shopping: -0.84
```

Table 4. Skewness and Kurtosis

Bivariate Analysis

Pairplot

Pairplot shows the relationship between the variables in the form of scatter plot and distribution of the variable in the form of histogram.

From the below pairplot we can see that there is positive linear relationship between advance_payments and spending, current_balance and spending, credit_limit and spending, current_balance and advance_payments, credit_limit and advance_payments, max_spent_in_single_shopping and current_balance.

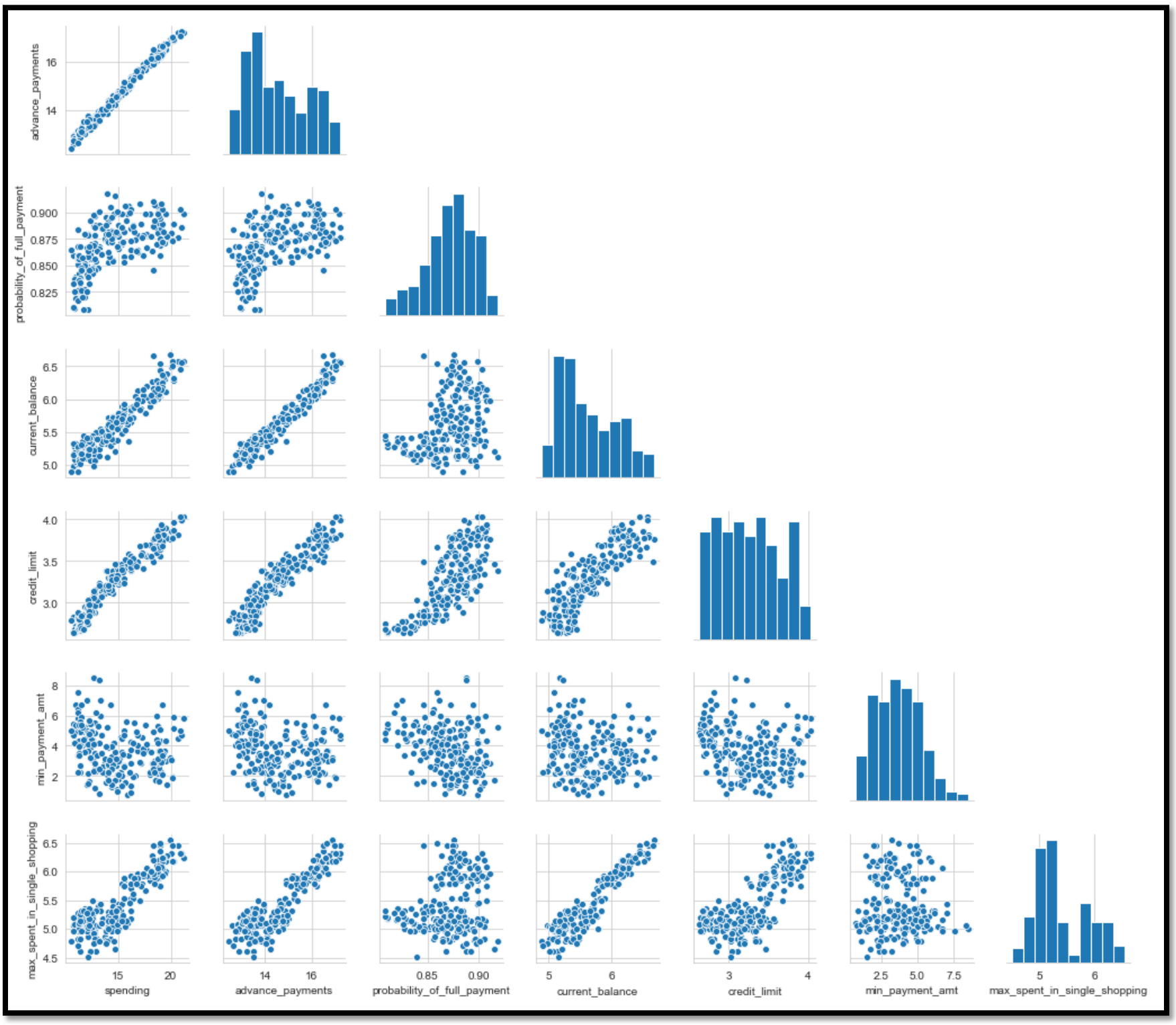

Figure 3. Pairplot

Lmplot

Lmplot shows a linear relationship between two variables using scatter plot for data points and a best line of fit.

Let us see some combinations of variables which have positive linear relationships inferred from the above pairplot.

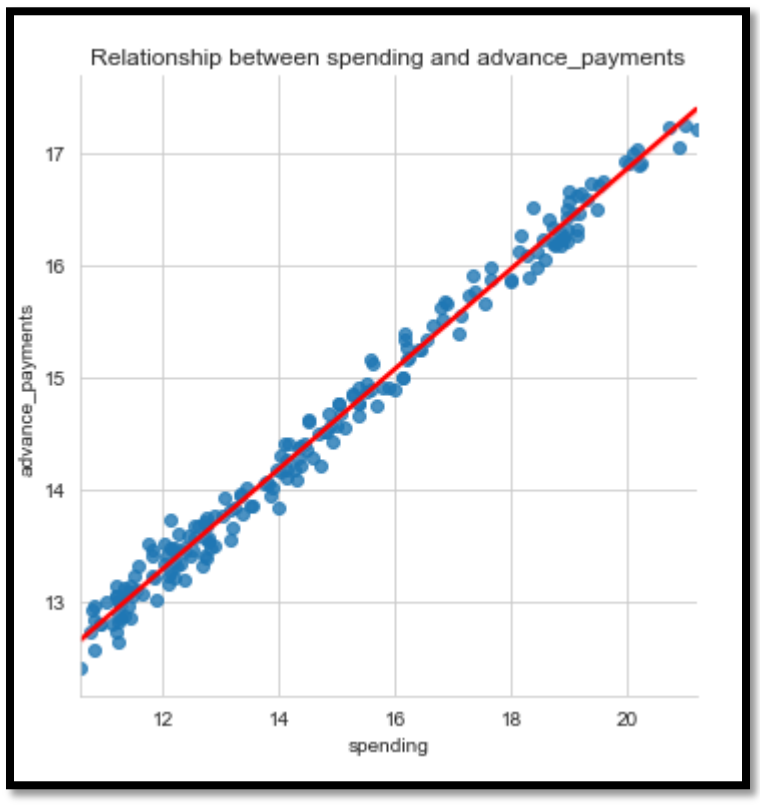

Figure 4. spending and advance_payments

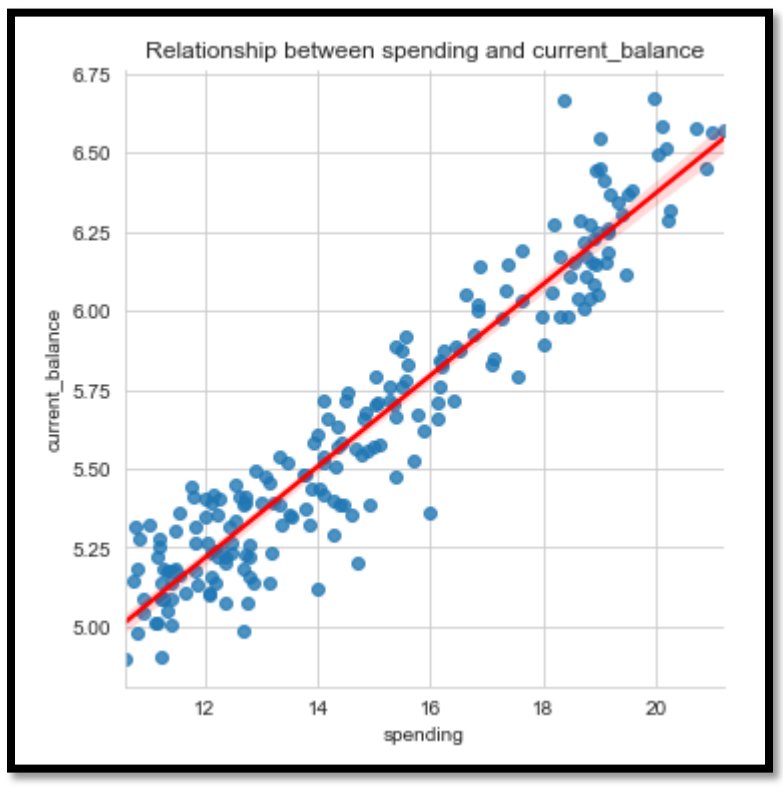

Figure 5. spending and current_balance

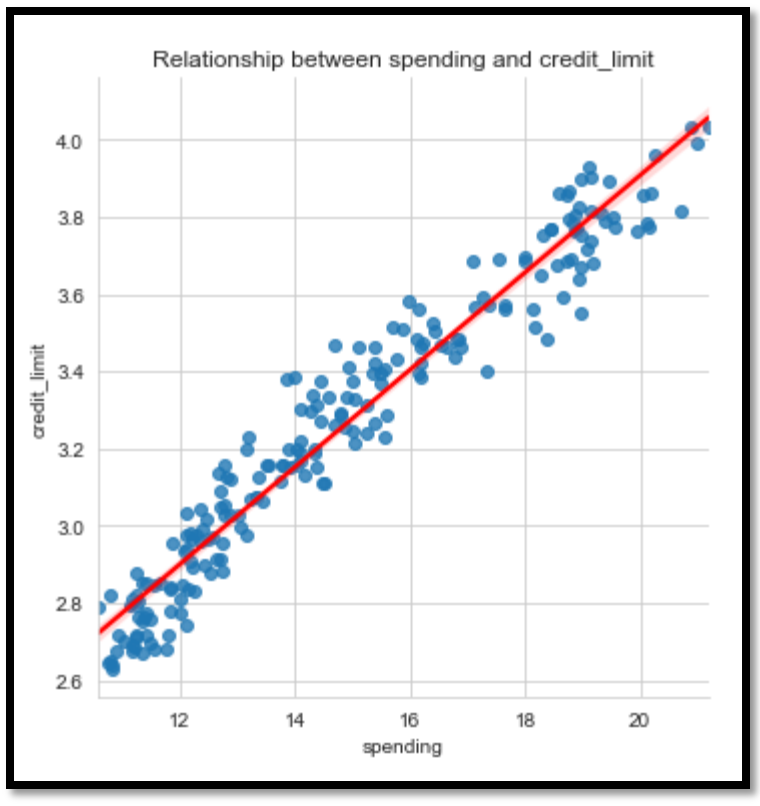

Figure 6. spending and credit_limit

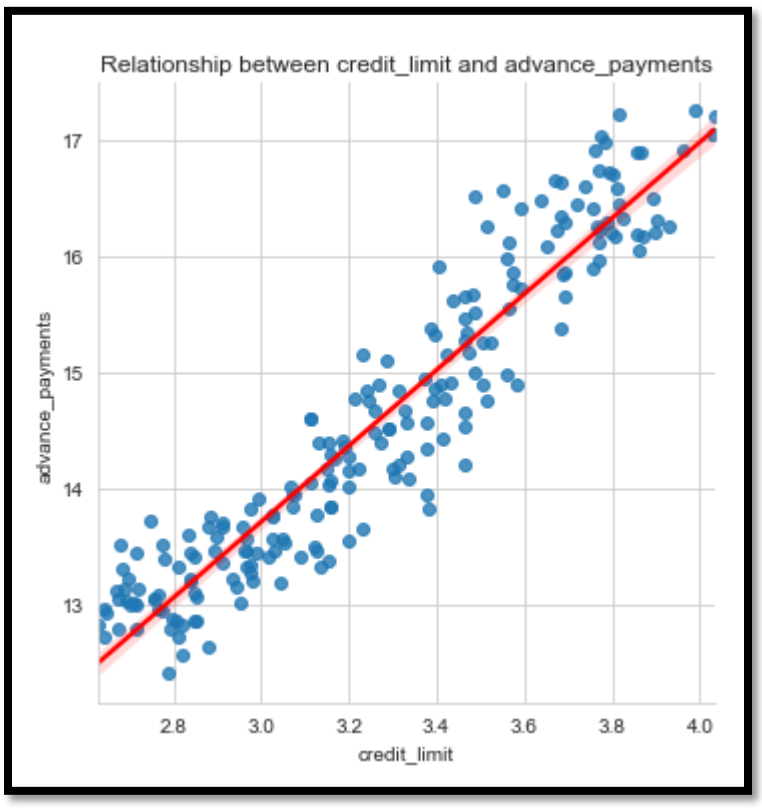

Figure 7. credit_limit and advance_payments

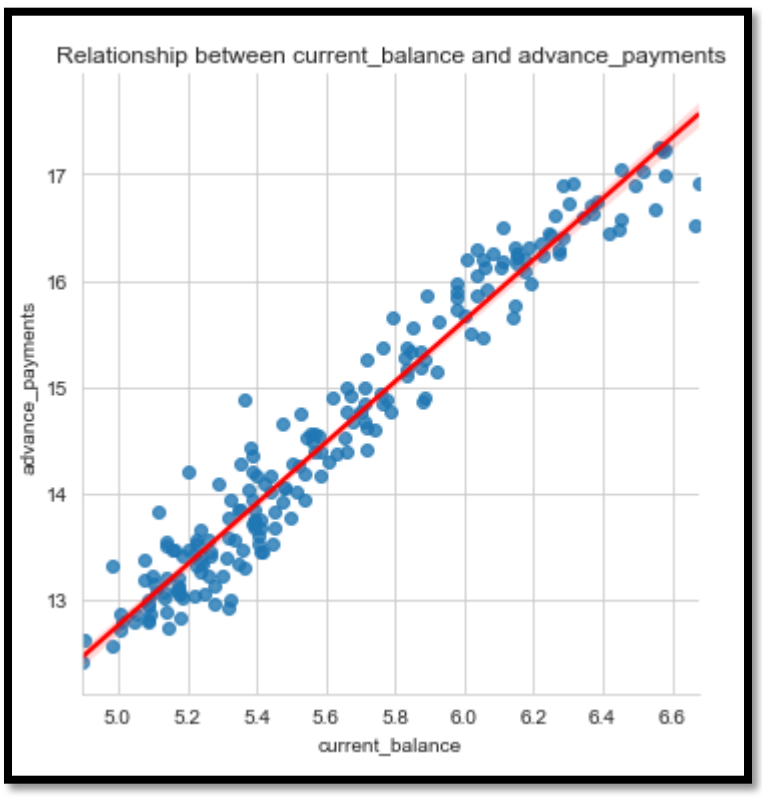

Figure 8. current_balance and advance_payments

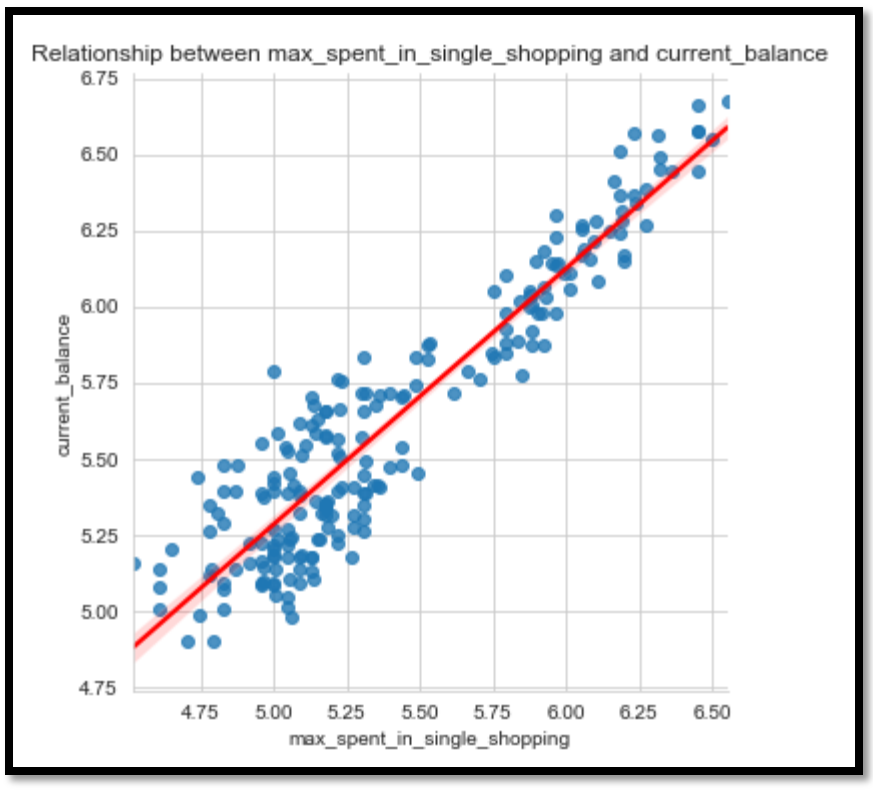

Figure 9. max_spent_in_single_shopping & current_balance

Correlation Heatmap

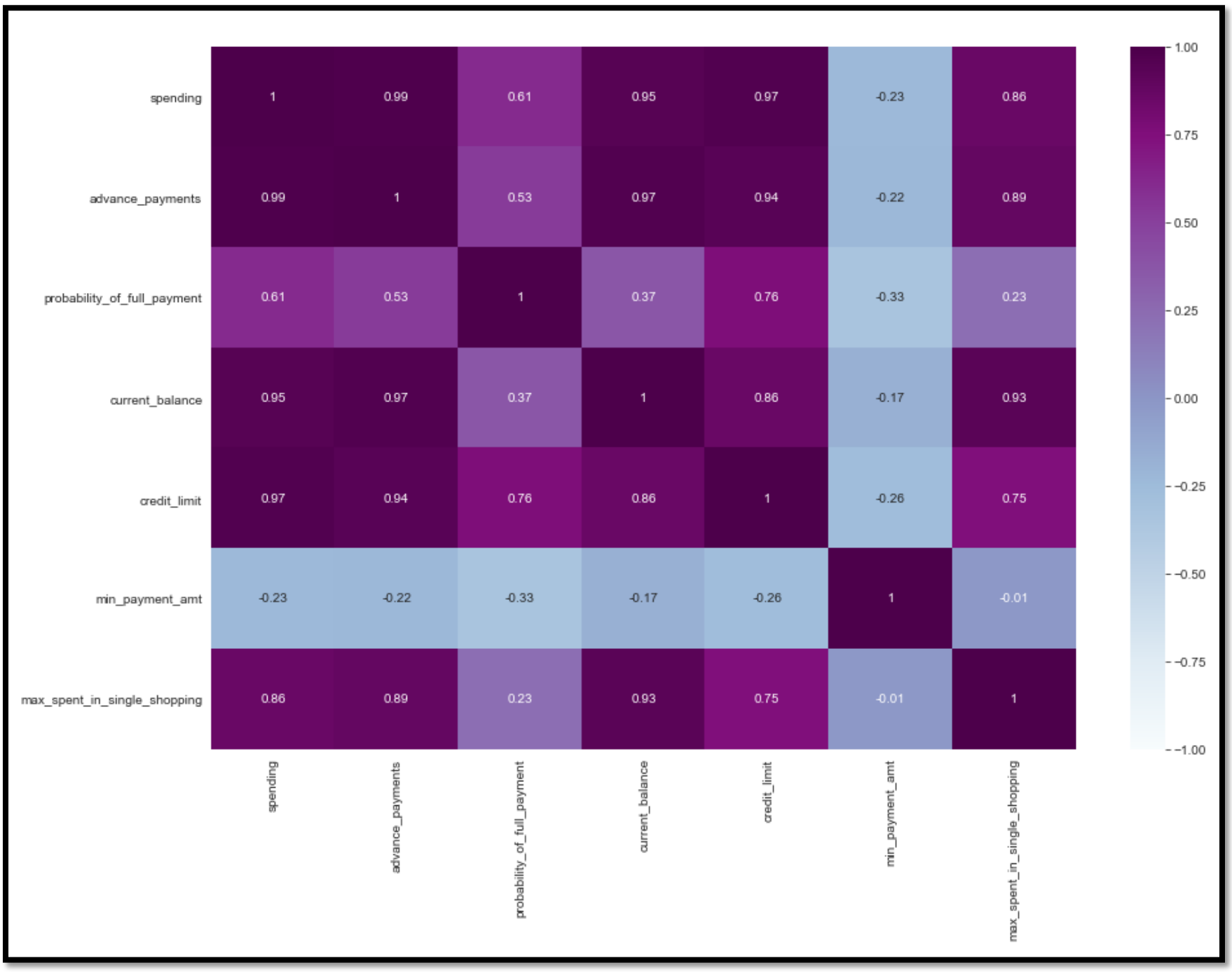

Figure 10. Correlation Heatmap

From the above correlation plot we can see that various aspects of credit card usage have high positive correlation with each other. Correlation values are always between 1 and -1. Those which are closer to 1 are positively correlated and those which near -1 are negatively correlated. Values near to 0 have no correlation.

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 basis 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.

Scaling or Standardization is an important step in data pre-processing. Most of the machine learning models use scaled data unless the data in hand is naturally scaled.

Let us see the variances between variables in the provided dataset.

spending	8.466351
advance_payments	1.705528
probability_of_full_payment	0.000558
current_balance	0.196305
credit_limit	0.142668
min_payment_amt	2.260684
max_spent_in_single_shopping	0.241553
dtype: float64	

Table 4. Variance

From the above table though there is not much variance between most of the variables, our target variable spending has a variance of 8.46 whereas other variables variance lie between 0 and 2. Hence scaling is necessary.

We will be using the Standard Scaler method for scaling our data. This method will calculate the z-score for each data point and then scale the data such that mean = 0 and variance/standard deviation = 1.

The standard score of a sample x is calculated as below:

$$z = \frac{x - \bar{x}}{s}$$

Figure 11. Z-score formula

Where **x-bar** is the mean and **s** is the standard deviation. The scaled data will be used for clustering models.

After using the Standard Scaler in sklearn package, below is the head of scaled data.

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

Table 5. Scaled 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.

Hierarchical Clustering

Hierarchical clustering is a technique of cluster analysis which performs hierarchy of clusters. There are mainly two types of hierarchical clustering:

1. Divisive: This is a top-to-down approach. It starts with all of the observations in the same cluster and then splits into smaller clusters.
2. Agglomerative: This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

Source: https://en.wikipedia.org/wiki/Hierarchical_clustering

Dendrogram

Dendrogram: is a tree-like diagram that summarizes the process of clustering in a visual format.

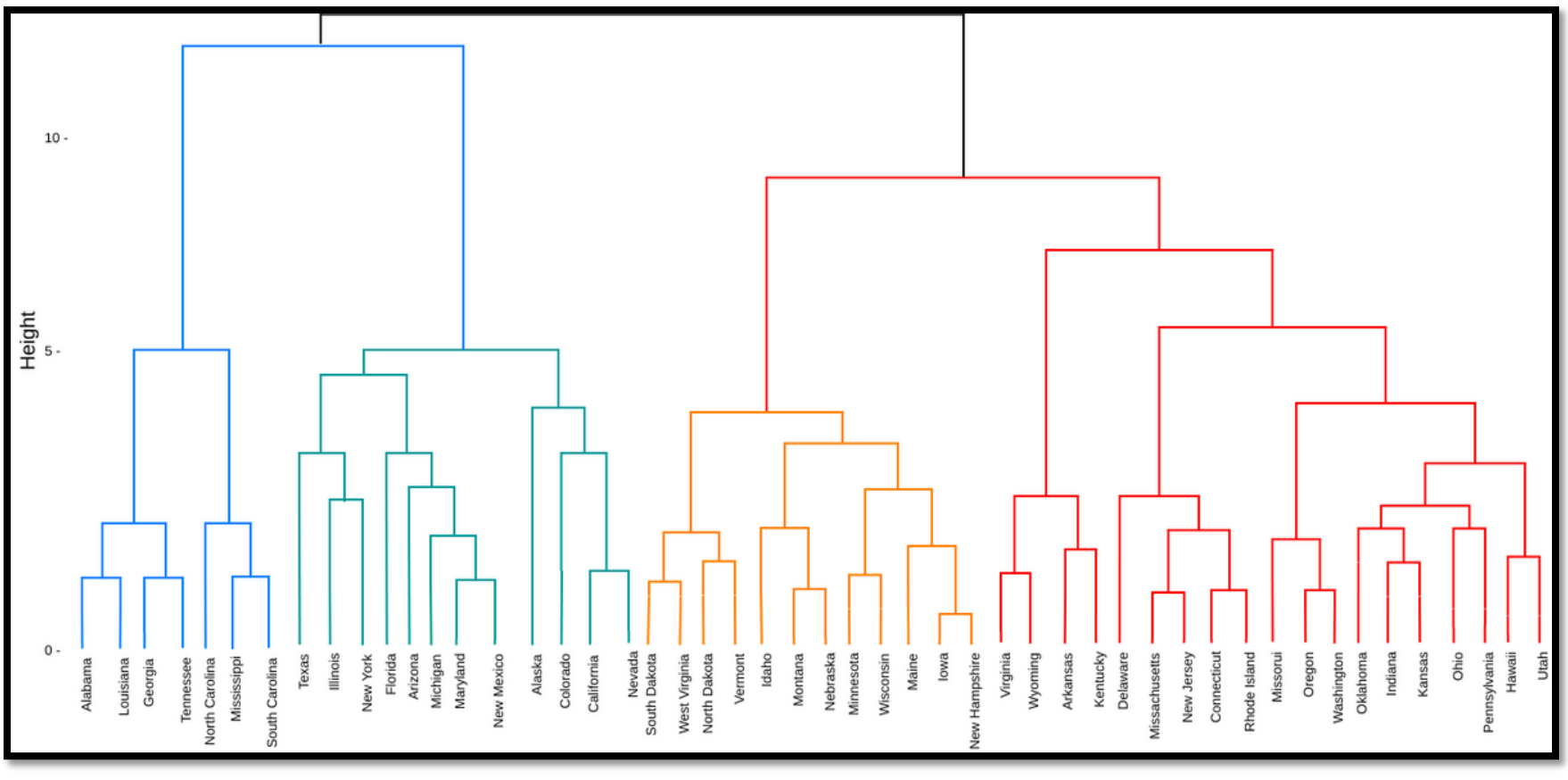

Figure 12. Example of Dendrogram

Source:
https://www.google.com/search?q=dendrogram&rlz=1C1GIWA_enIN900IN900&source=lnms&tbm=isch&sa=X&ved=2ahUKEwj2nvyMqYvzAhUUILcAHZEzB94Q_AUoAXoECAEQAw&biw=1920&bih=937&dpr=1#imgrc=npw5YAGtxrZh5M

Concept of Linkage

Once the clusters are formed, the distance between clusters are calculated using different linkage methods. The different linkage types are:

1. Single linkage
2. Complete linkage
3. Average linkage
4. Centroid linkage
5. Ward's method

For our dataset we will only look at single, complete and Ward's method linkages.

Single linkage: Distance between two clusters is defined as the shortest distance between two points in each cluster.

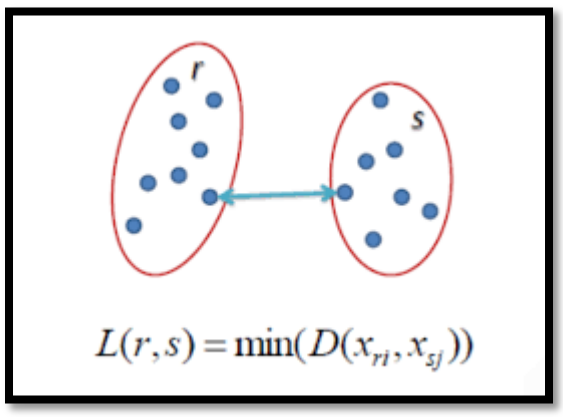

Figure 13. Single Linkage

Source: Google Images

Complete linkage: Distance between two clusters is defined as the longest distance between two points in each cluster.

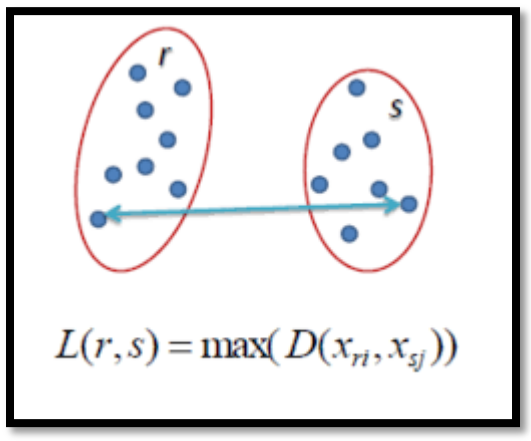

Figure 14. Complete Linkage

Source: Google Images

Ward's method: joins records and clusters together progressively to produce larger and larger clusters.

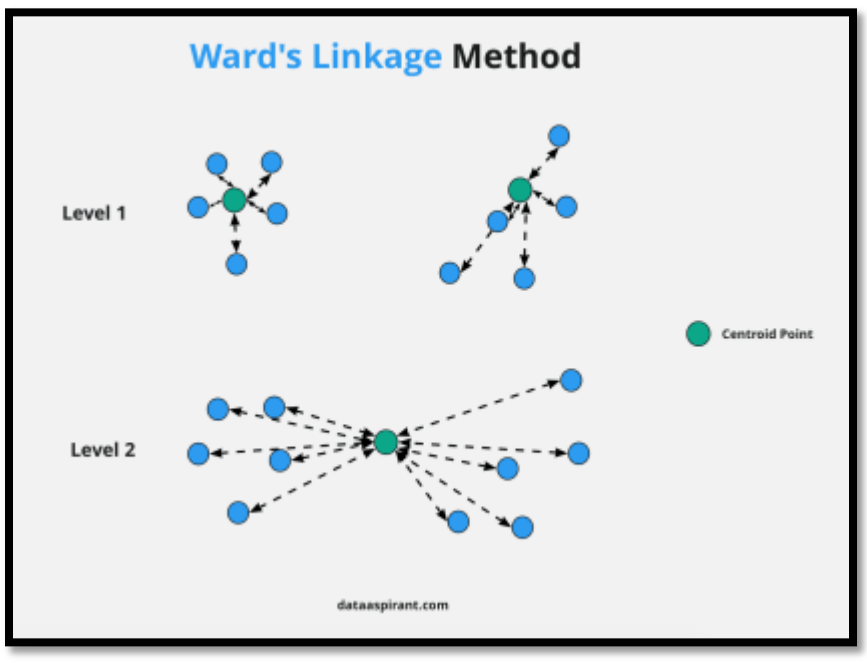

Figure 15. Ward's linkage method

Source: Google Images

Visualization is vital to understand different clusters. For our dataset, we have used both single and complete linkage with Euclidean and Manhattan distance calculations. We will use the scipy package for dendrogram and linkage.

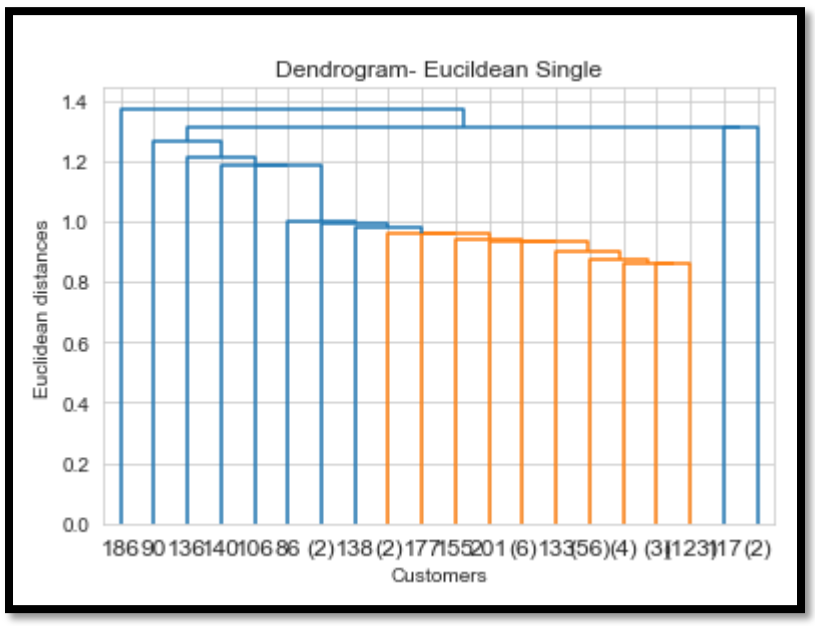

Figure 16. Dendrogram – Euclidean Single

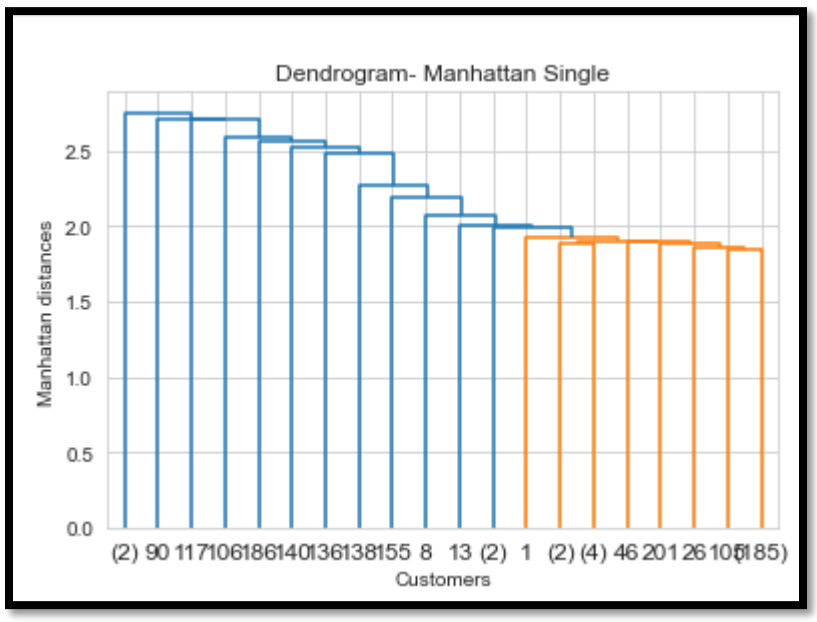

Figure 17. Dendrogram – Manhattan Single

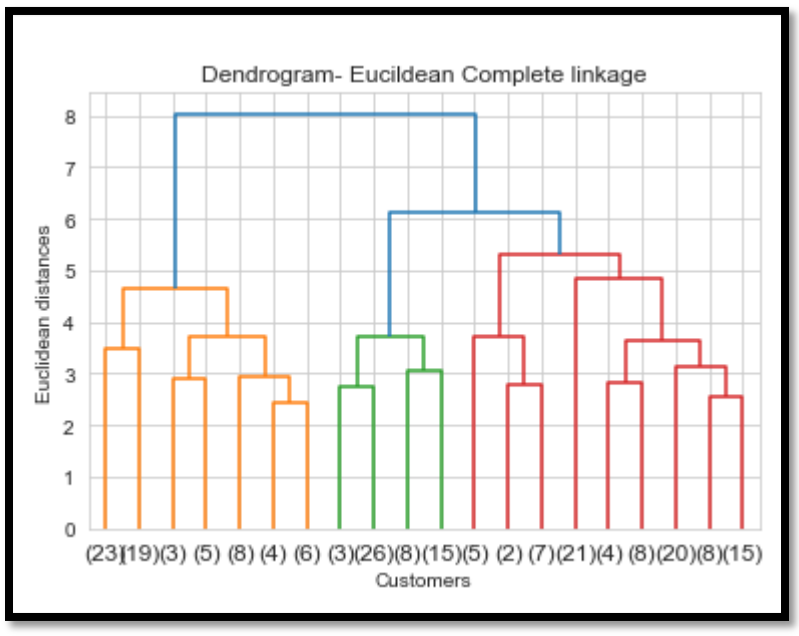

Figure 18. Dendrogram – Euclidean Complete

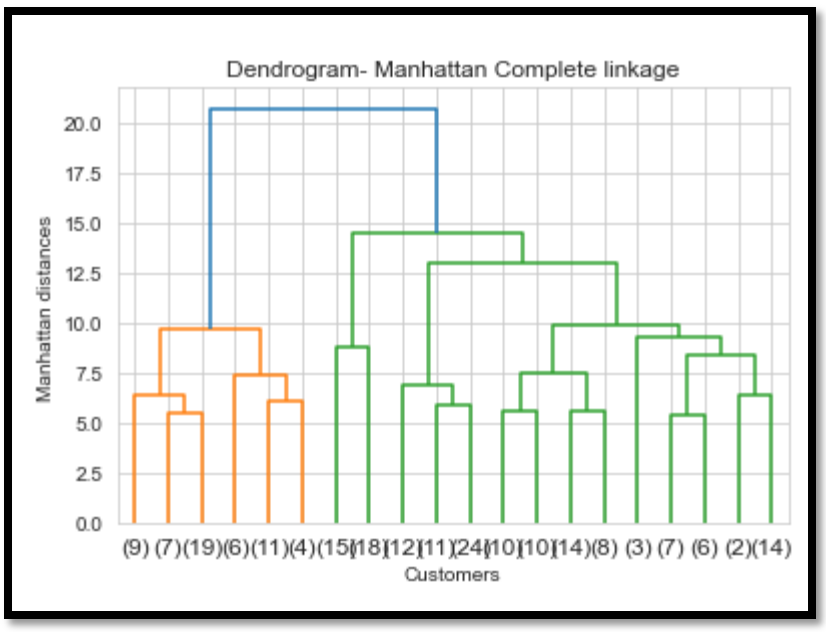

Figure 19. Dendrogram – Manhattan Complete

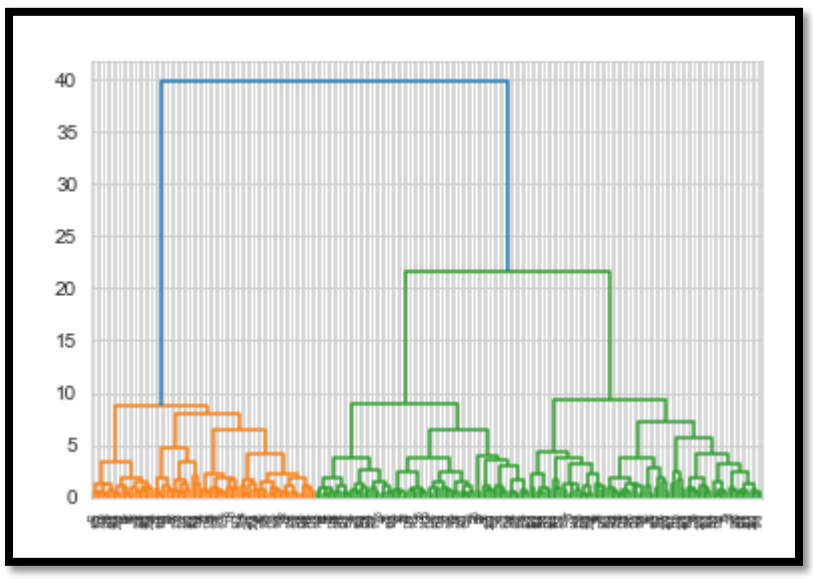

Figure 20. Dendrogram – Ward's Method

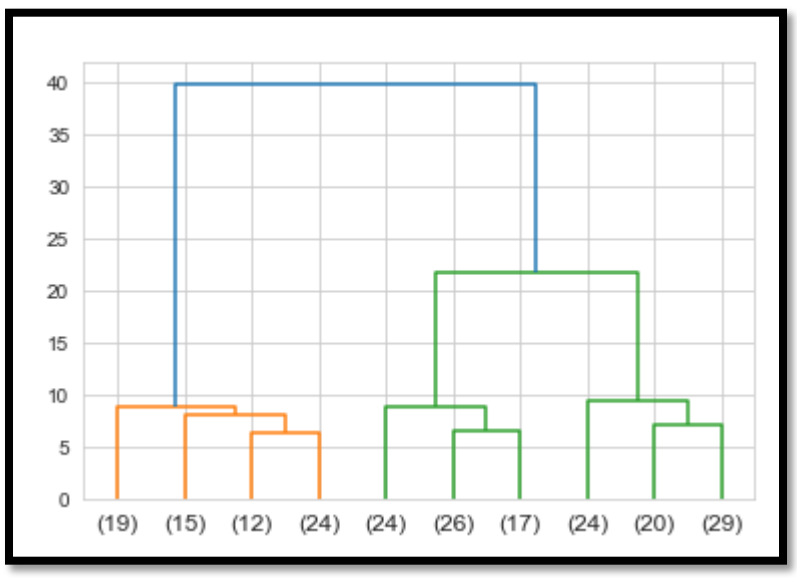

Figure 21. Dendrogram – Ward's Method (Truncated)

From the above Figure 20 we can see that the data has been segregated into three clusters by color (orange and green) using Ward's method. However since we are not able to see the data points, we have truncated the dendrogram.

Head of our dataset after merging the clusters:

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

Table 6. Dataset after merging clusters

Let us now see the relationship between clusters and each variable using scatterplots.

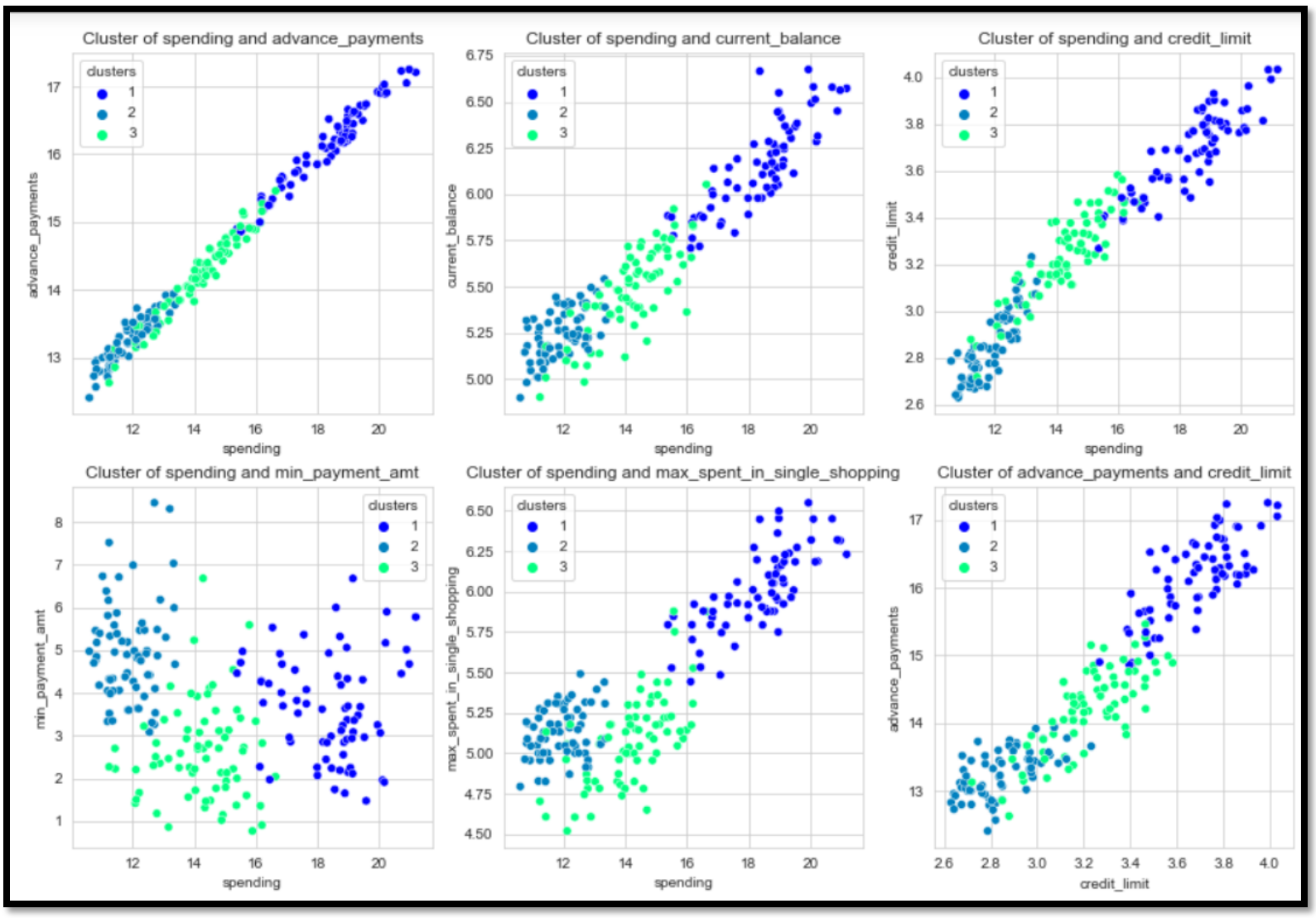

Figure 22. Hierarchical Clusters

Figure 18 shows a dendrogram with 3 clusters segmented with orange, green and red using the complete linkage method with Euclidean distance. Similarly we can see 3 clusters formed using the Ward's method. Hence we can conclude that 3 clusters would be the optimum number of clusters for the given dataset.

1.4 Apply K-Means clustering on scaled data and determine optimum clusters (2 pts). Apply elbow curve and silhouette score (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. Silhouette Score must be calculated for the same values of K taken above and commented on. Report must contain logical and correct explanations for choosing the optimum clusters using both elbow method and silhouette scores. Append cluster labels obtained from K-means clustering into the original data frame. Customer Segmentation can be visualized using appropriate graphs.

K-Means clustering is a unsupervised learning algorithm which tries to find groups or form clusters on the basis of their similarity. It was developed by researcher named James Macqueen in 1967.

Parameter K: k is the target variable which refers to the number of centroids in the given dataset.

Means: in K-Means clustering, '**Means**' refers to the averaging of data to find the centroid in a cluster.

There are techniques to find the optimal number of k values as below:

1. The Elbow method
2. The Silhouette method

Let us now apply K-means clustering on the scaled data followed by calculating WSS scores and plotting them to check the optimal k value using Elbow method. We will be using sklearn.cluster package to use Kmeans.

After we scaled the data, we will first try applying k-means clustering with number of clusters as 3. Below are the labels after applying k-means clustering.

```
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])
```

Table 7. K-means labels

However, when we try applying k-means clustering with k value starting from 1 to 10, we have the below inertia/WSS scores for each cluster.

```
[1469.9999999999998,
 659.171754487041,
 430.6589731513006,
 371.1846125351018,
 326.30618276116064,
 289.9983082056098,
 263.94786541626377,
 241.7618664018391,
 223.5788019487927,
 207.14125192816445]
```

Table 8. WSS Scores

From the above WSS scores, we can see that for cluster 1 the score is 1469.99 and the score for 2 clusters dropped to 659.17 which is a significant difference in the scores. For cluster 3, the score is 430.65 whereas from cluster 3 to cluster 10 we see that there is no significant decrease in the scores. Hence we arrive at the optimum no of clusters as 3.

Let us now see the Elbow method visually to understand the scores better.

Elbow Method

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

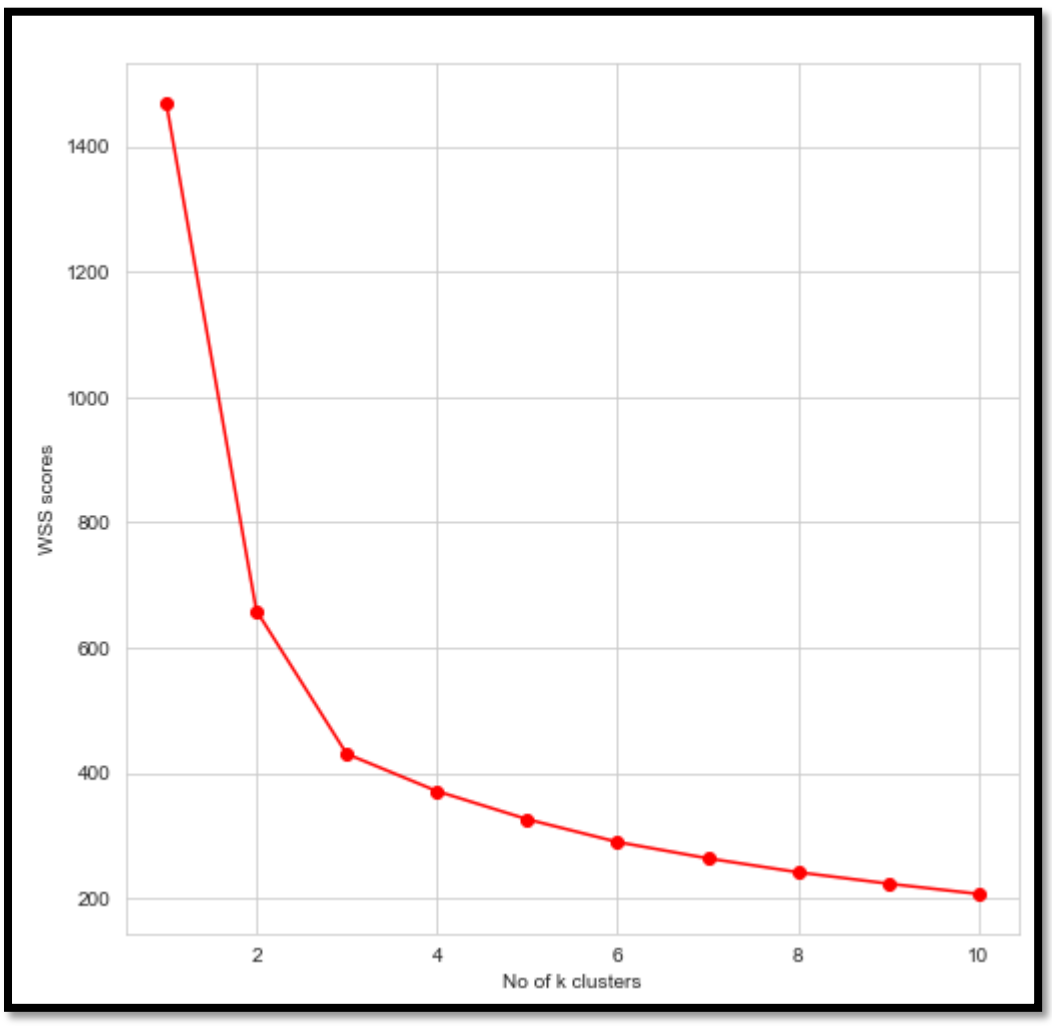

Figure 23. The Elbow method

We will now see the Silhouette scores to see if 3 clusters are the optimal clusters for the given dataset.

Silhouette Method

Silhouette is a different method to determine optimal number of clusters for given dataset. It defines as coefficient of measure of how similar an observation to its own cluster compared to that of other clusters. The range of silhouette coefficient varies between -1 to 1.

We will be using the sklearn package to calculate the Silhouette scores. The Silhouette scores for 3 clusters is **0.40072705527512986** which is a good coefficient score for the given dataset.

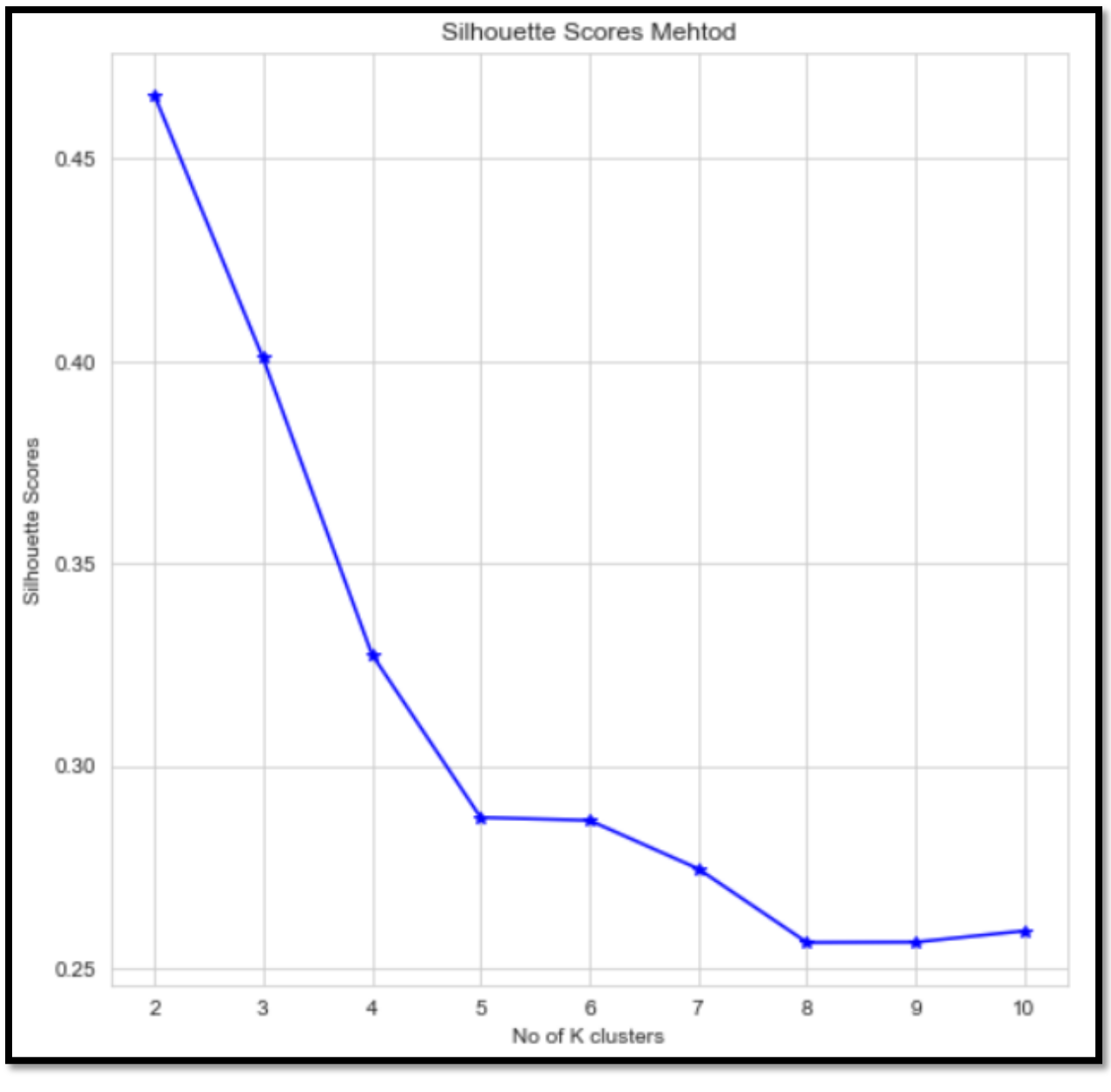

Figure 24. Silhouette Scores

Let us now append the original dataframe with the k_means clusters.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	kmeans_clusters
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	0
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	2
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	0
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	0

Table 9. K-means clusters merged with original dataframe

Visualizing the clusters for each variable using scatter plots. We can see that the data is segregated into 3 clusters separating each other with distinct colors (red, blue and green).

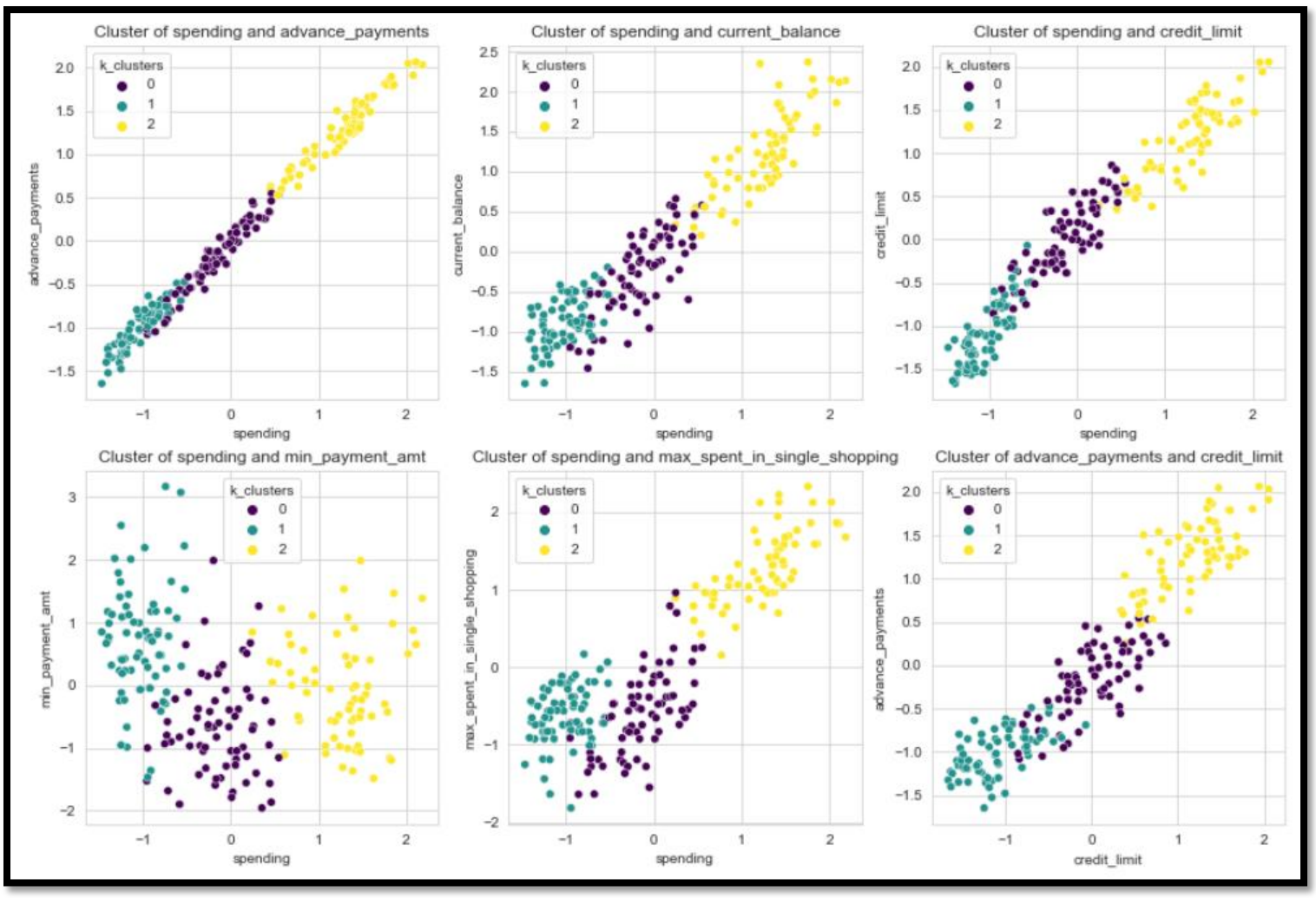

Figure 25. K-Means Clusters

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 dataframe, 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.

Recommendations for Hierarchical clustering

clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	frequency
1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371	70
2	11.872388	13.257015	0.848072	5.238940	2.848537	4.949433	5.122209	67
3	14.199041	14.233562	0.879190	5.478233	3.226452	2.612181	5.086178	73

Table 10. Hierarchical Clusters

When we look at the final clusters merged with original dataset and take the average values for the variables, below are the recommendations for each cluster profile.

Cluster 1: Platinum customers
Cluster 3: Gold customers
Cluster 2: Silver customers

Customers under cluster 1 have a high spending, current balance, credit_limit and max_spent_in_single_shopping which clearly shows that they are premium high-net worth customers who make expensive purchases on their credit cards.

Customers under cluster 3 have a relatively lesser spending, current balance, credit_limit and max_spent_in_single_shopping which indicate that they are upper middle class customers. The bank can provide promotional offers to this segment such that they increase their spending and are potential customers who can move into premium segments.

Customers under cluster 2 have the least spending and credit_limits compared to other clusters. This signifies that they are customers who have recently bought credit cards or youths who have started working recently. Bank can provide customized offers to this segment to promote more spending on credit cards.

Recommendations for K-Means clustering

s_clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	frequency
0	18.495373	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701	67
1	11.856944	13.247778	0.848253	5.231750	2.849542	4.742389	5.101722	72
2	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803	71

Table 11. K-Means Clusters

When we look at the final clusters merged with original dataset and take the average values for the variables, below are the recommendations for each cluster profile.

Cluster 0: Platinum customers
Cluster 2: Gold Customers
Cluster 1: Silver Customers

Customers under cluster 0 have a high spending, current balance, credit_limit and max_spent_in_single_shopping which clearly shows that they are premium high-net worth customers who make expensive purchases on their credit cards.

Customers under cluster 2 have a relatively lesser spending, current balance, credit_limit and max_spent_in_single_shopping which indicate that they are upper middle class customers. The bank can provide promotional offers to this segment such that they increase their spending and are potential customers who can move into premium segments.

Customers under cluster 1 have the least spending and credit_limits compared to other clusters. This signifies that they are customers who have recently bought credit cards or youths who have started working recently. Bank can provide customized offers to this segment to promote more spending on credit cards.

Problem 2: CART-RF-ANN

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

Introduction

The purpose of this whole exercise is to explore the dataset. Do the exploratory data analysis. Explore the dataset using central tendency and other parameters. The data consists of customer details who have availed tour insurance. We will perform exploratory data analysis to understand what the given data has to say and then use Decision Trees, Random Forest and Artificial Neural Network to build a model which predicts the claim status and provide recommendations to management with the business insights gained.

Data Dictionary for Market Segmentation

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

2.1 Read the data 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.

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
5	45	JZI	Airlines	Yes	15.75	Online	8	45.00	Bronze Plan	ASIA
6	61	CWT	Travel Agency	No	35.64	Online	30	59.40	Customised Plan	Americas
7	36	EPX	Travel Agency	No	0.00	Online	16	80.00	Cancellation Plan	ASIA
8	36	EPX	Travel Agency	No	0.00	Online	19	14.00	Cancellation Plan	ASIA
9	36	EPX	Travel Agency	No	0.00	Online	42	43.00	Cancellation Plan	ASIA

Table 12. Sample Dataset

Dataset has 10 variables. Target variable is Claimed attribute. We will be building the model keeping this as the dependent variable.

Exploratory Data Analysis

Information on 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 13. Information on the dataset

The given dataset has 3000 observations with 10 variables. We have 4 numeric variables and 6 object type features. It is clearly evident that we have no missing or null values in the dataset.

Duplicates in the dataset

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

139 rows × 10 columns

Table 14. Duplicates in the dataset

There are 139 rows of duplicates in the dataset. Hence we will be removing the duplicates from the dataset.

Descriptive Statistics

	count	mean	std	min	10%	25%	50%	75%	95%	max
Age	3000.0	38.091000	10.463518	8.0	26.000	32.0	36.00	42.000	60.000	84.00
Commision	3000.0	14.529203	25.481455	0.0	0.000	0.0	4.63	17.235	63.210	210.21
Duration	3000.0	70.001333	134.053313	-1.0	5.000	11.0	26.50	63.000	367.000	4580.00
Sales	3000.0	60.249913	70.733954	0.0	11.171	20.0	33.00	69.000	228.565	539.00

Table 15. Descriptive Statistics

From the above table below are the observations:

1. Age seems to be normally distributed. The minimum age is 8 yrs and max age is 84 yrs which shows that we might have the correct data on Age. Average age of customers in the dataset is 38 yrs.
2. Commission received for tour insurance is widely spread. The average commission is 14.52 however the minimum commission received is 0 and maximum commission received is 210.21 and 95% of data lies within 63.21 which indicates that there might be outliers.
3. Duration of the tour ranges from 134 days to 4580 days which means there are outliers. 95% of data lies within 367 days whereas maximum days shows 4580 days which is close to 12 years. It is surprising to see that someone would go for a tour for 12 years. Need to consult the business to validate the data. For now we will go with further analysis as is. The minimum duration shows -1 which cannot be the case in reality. Hence we might drop this row or impute it mean value to treat bad data.
4. Sales figures ranges from 0 to 539 (in 100's). The average sales is amounted to 60.24. There might be outliers as well.
5. Except for Age the other variables seem to be skewed with outliers present.

Treating Bad Data

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
1508	25	JZI	Airlines	No	6.3	Online	-1	18.0	Bronze Plan	ASIA

Table 16. Bad Data

From the above we can see that we have one observation in Duration column with value -1 which cannot be the case. Duration of a stay cannot be in negative numbers. Hence let us impute the bad data with mean value 72.14 days.

```
Age                25
Agency_Code       JZI
Type          Airlines
Claimed             No
Commision          6.3
Channel         Online
Duration       72.1202
Sales               18
Product Name  Bronze Plan
Destination          ASIA
Name: 1508, dtype: object
```

Table 17. Post Bad Data Treatment

Univariate Analysis

Boxplots

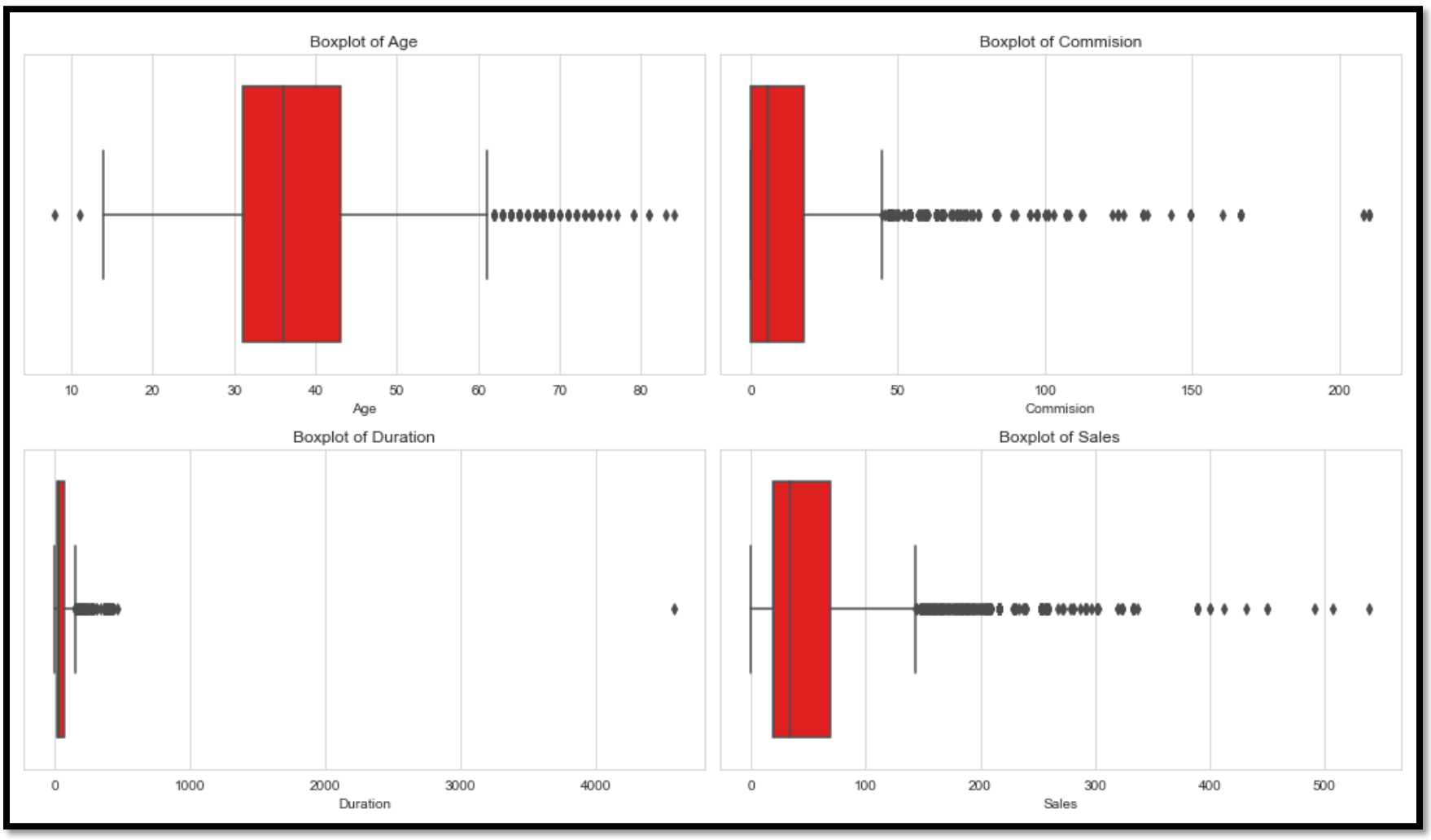

Figure 26. Boxplots

From the above boxplots, we can see that there are outliers present in the dataset for each numerical variable.

Distribution Plots

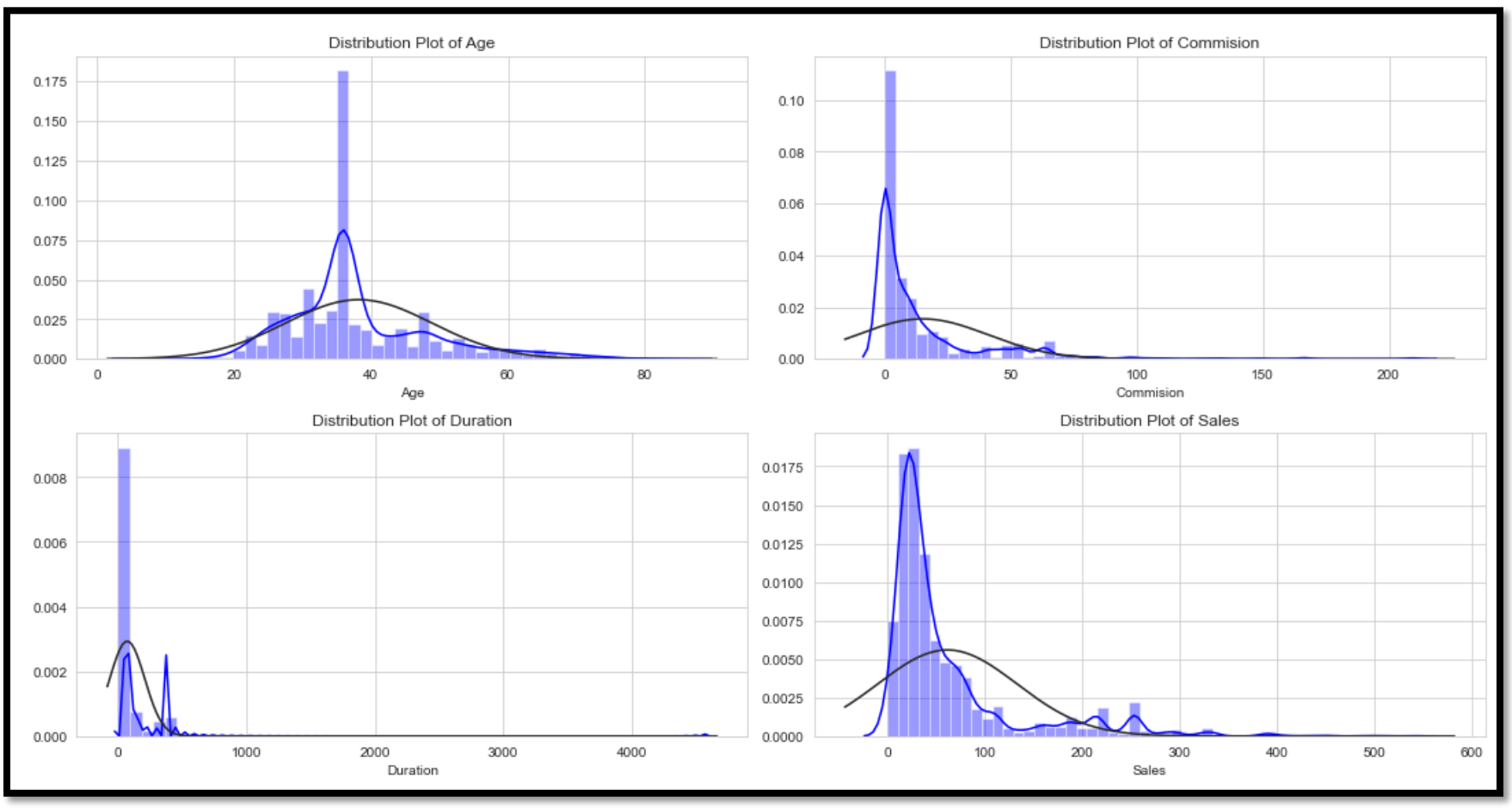

Figure 27. Distribution Plots

From the above distribution plots we can see that the variables are highly skewed except for Age. Let's validate the same with Skewness and Kurtosis scores.

```
Skewness of Age: 1.1
Kurtosis of Age: 1.44
Skewness of Commision: 3.1
Kurtosis of Commision: 13.59
Skewness of Duration: 13.79
Kurtosis of Duration: 422.71
Skewness of Sales: 2.34
Kurtosis of Sales: 5.97
```

Table 18. Skewness and Kurtosis

Plotting numerical variables w.r.t Claimed status

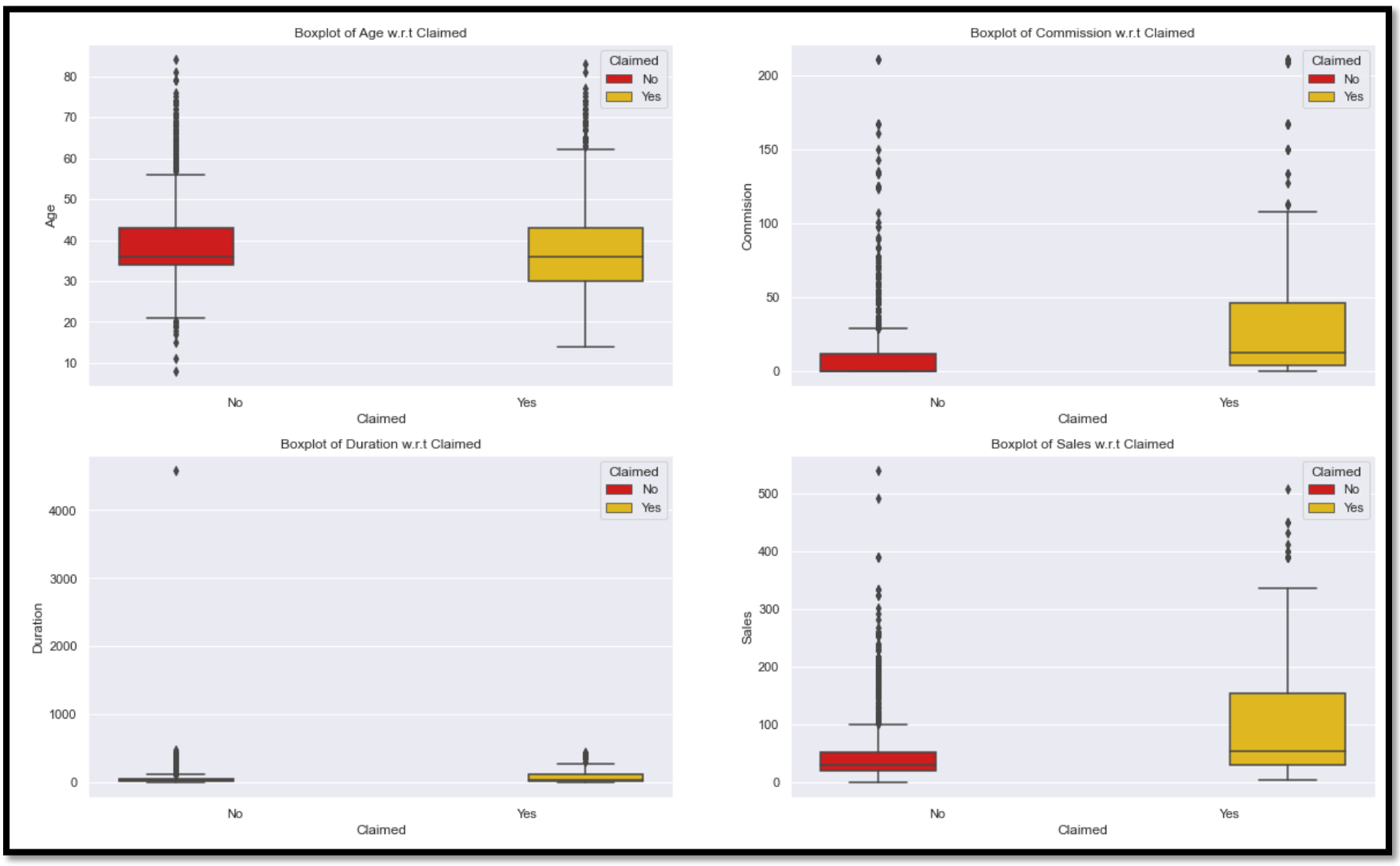

Figure 28. Boxplots of Numerical variables w.r.t Claimed status

From the above boxplots below are the observations:
Among the age, we see we have more customers who have claimed the insurance though we have more or less same spread for both categories.
When we see commission in relation to claimed status, we see that customers who have claimed are more compared to those who have not claimed the insurance.
We can hardly see any difference with respect to duration.
Sales are higher for those who have claimed compared to those who have not claimed.

Bivariate Analysis

Plotting Age with categorical variables w.r.t Claimed status

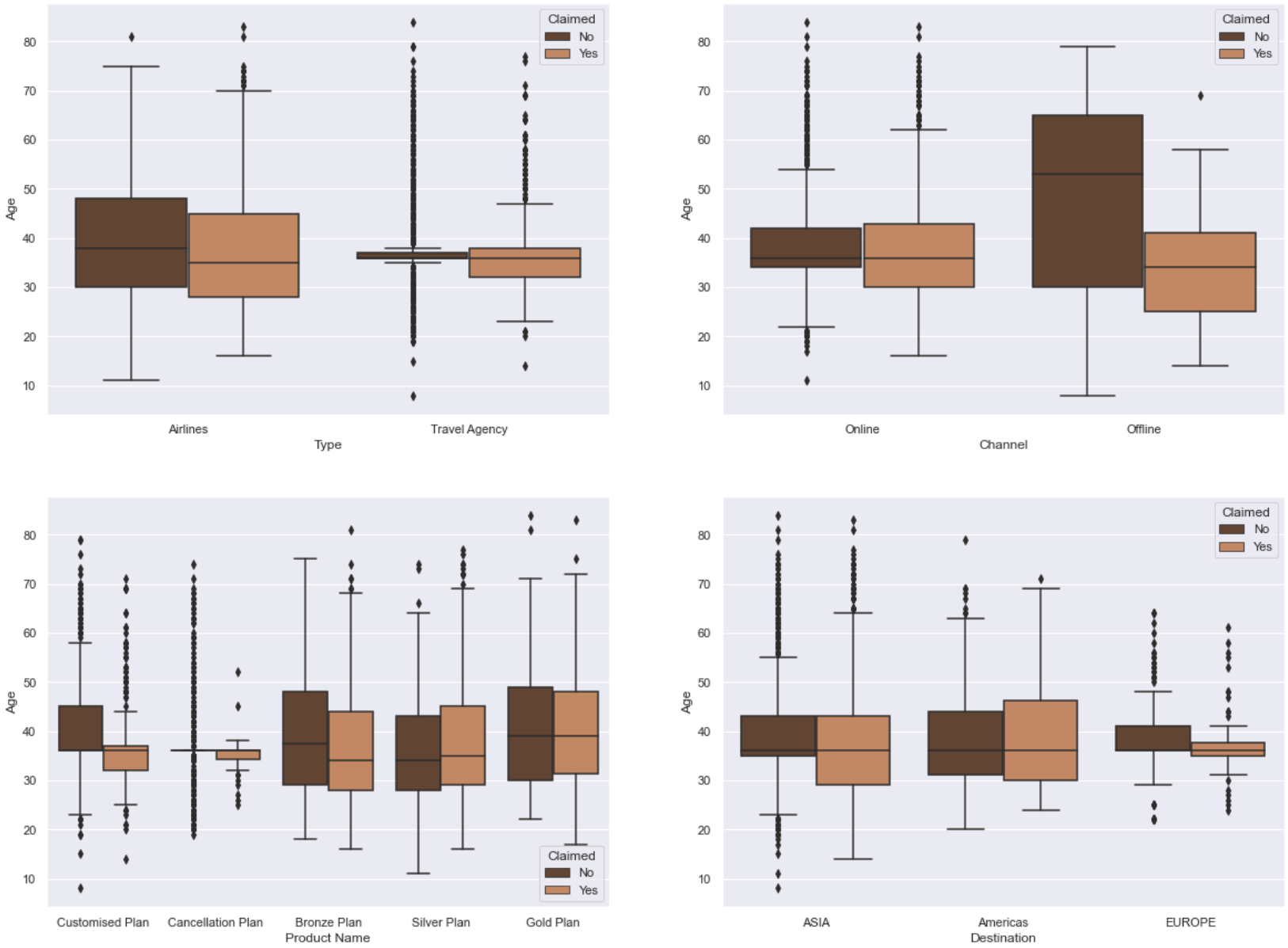

Figure 29. Boxplots of Age with Categorical variables w.r.t Claimed status

Plotting Commission with categorical variables w.r.t Claimed status

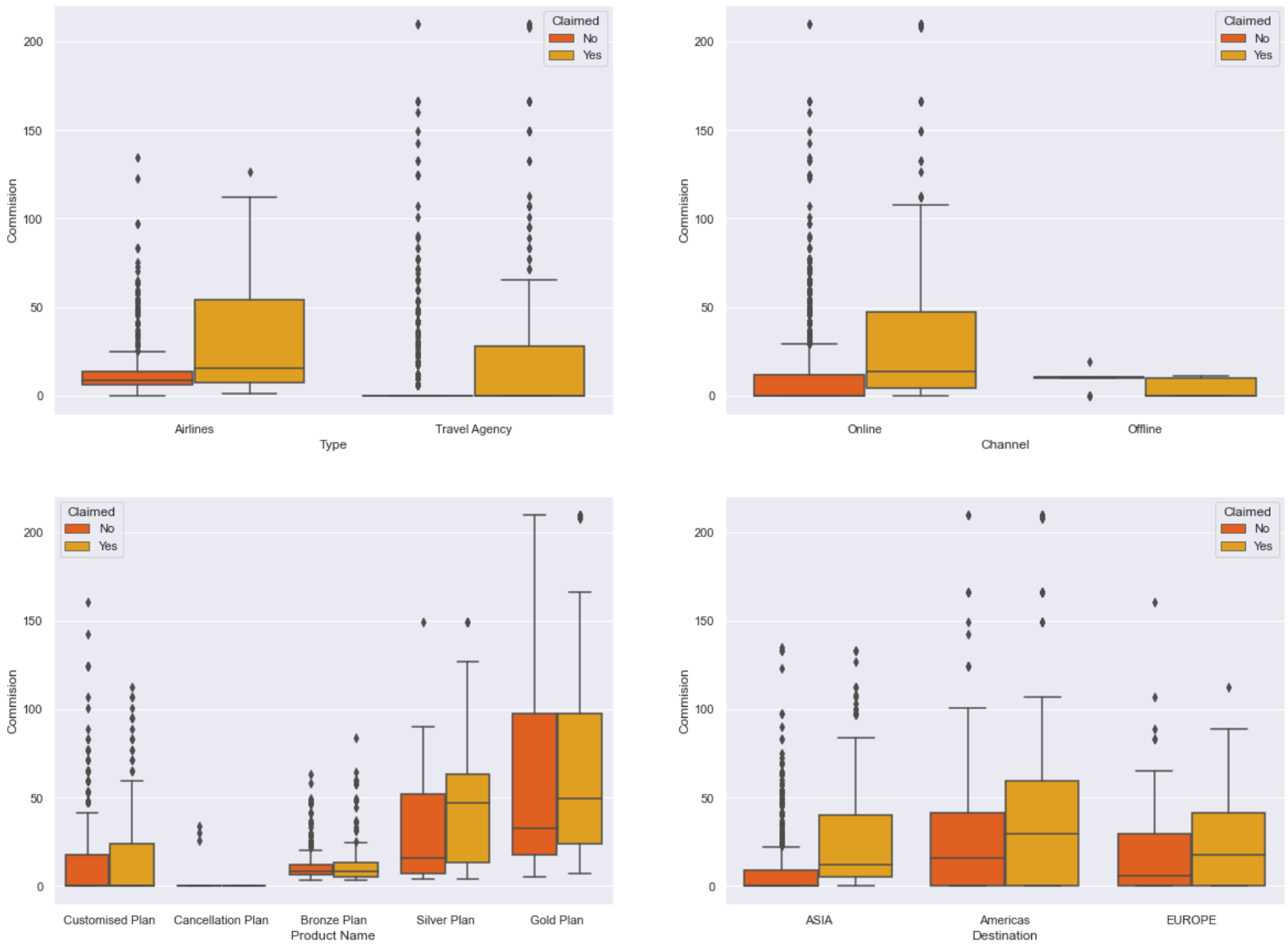

Figure 30. Boxplots of Commission with Categorical variables w.r.t Claimed status

Plotting Duration with categorical variables w.r.t Claimed status

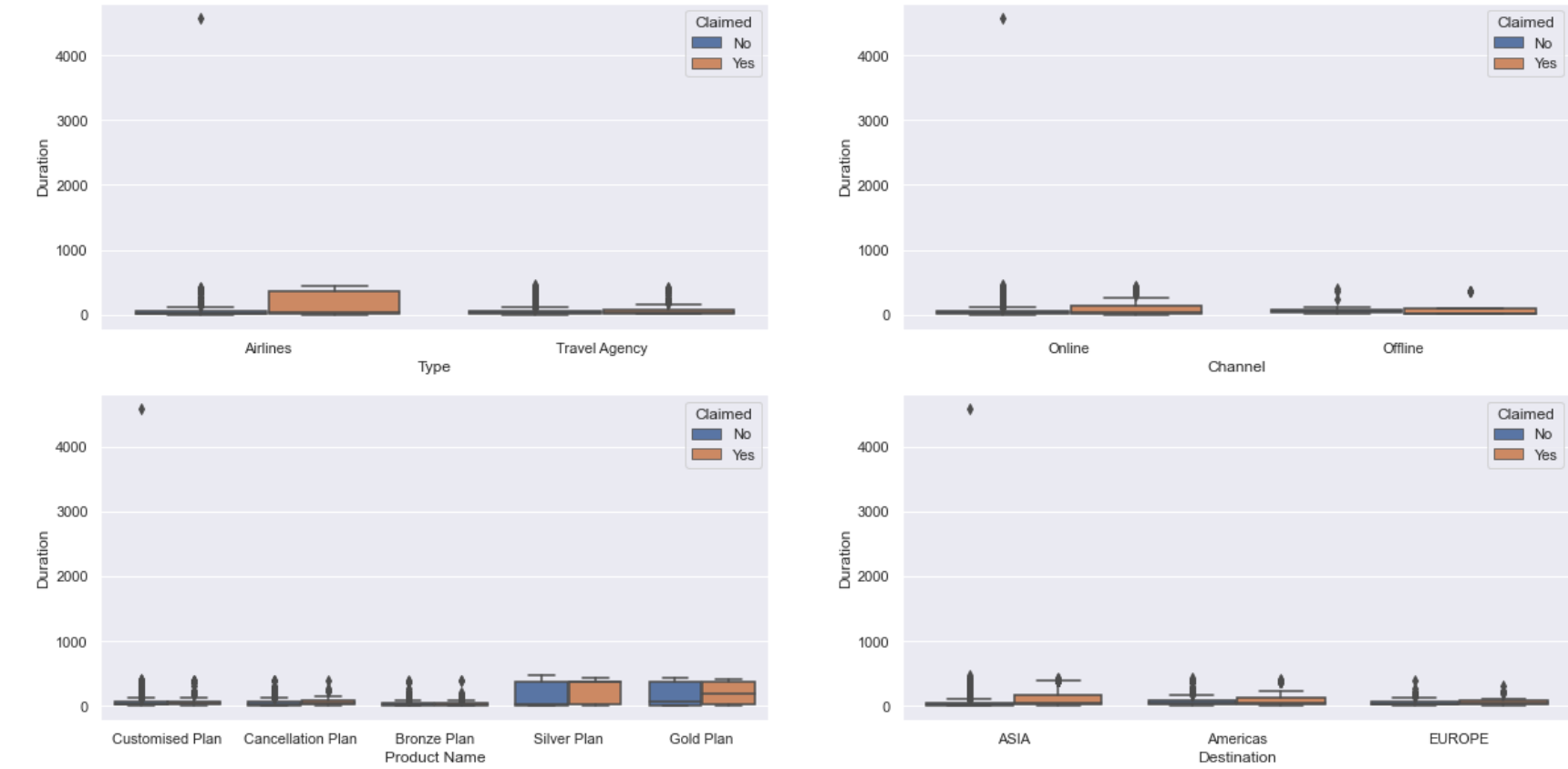

Figure 31. Boxplots of Duration with Categorical variables w.r.t Claimed status

Plotting Sales with categorical variables w.r.t Claimed status

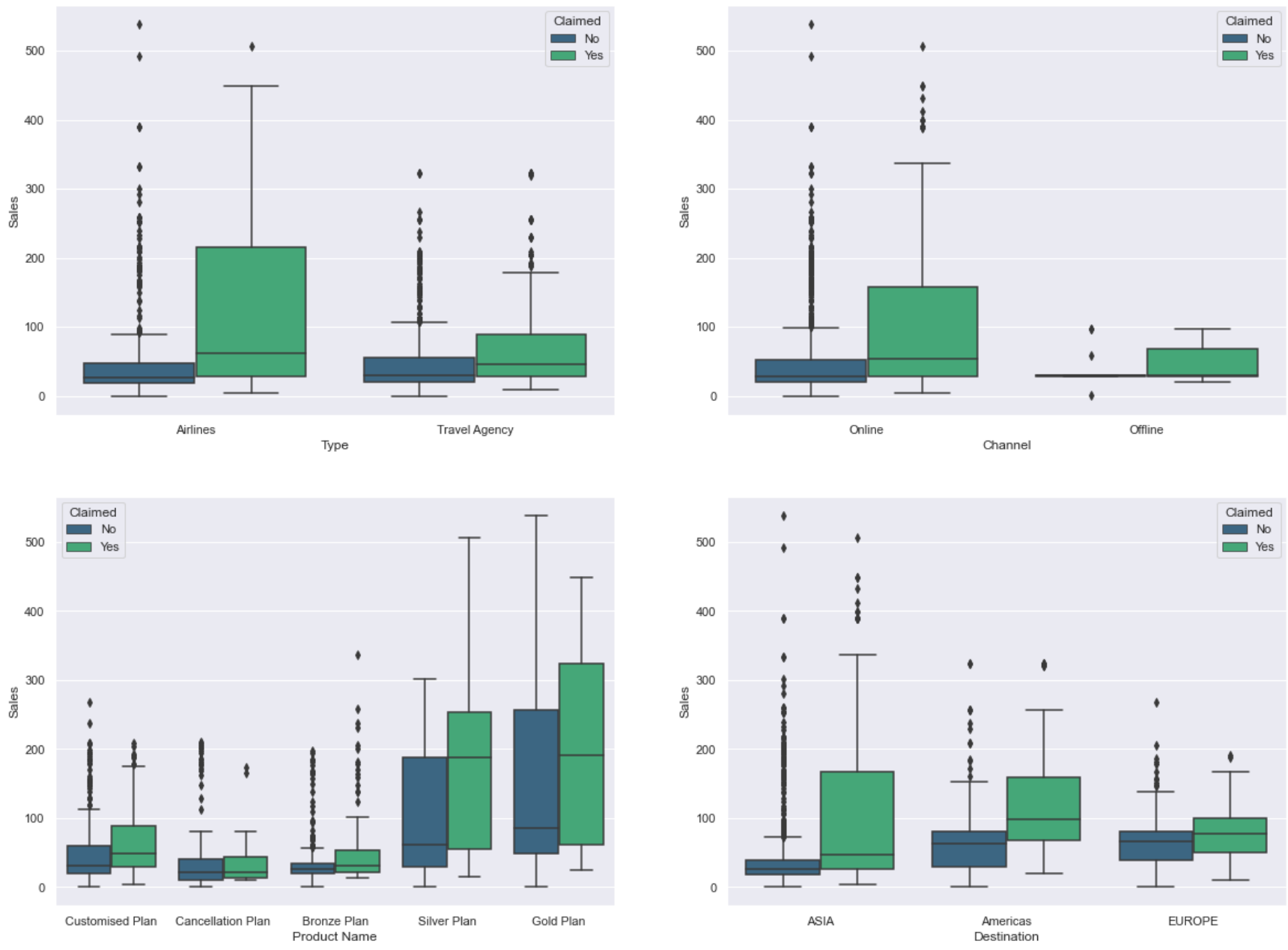

Figure 32. Boxplots of Sales with Categorical variables w.r.t Claimed status

Pairplot

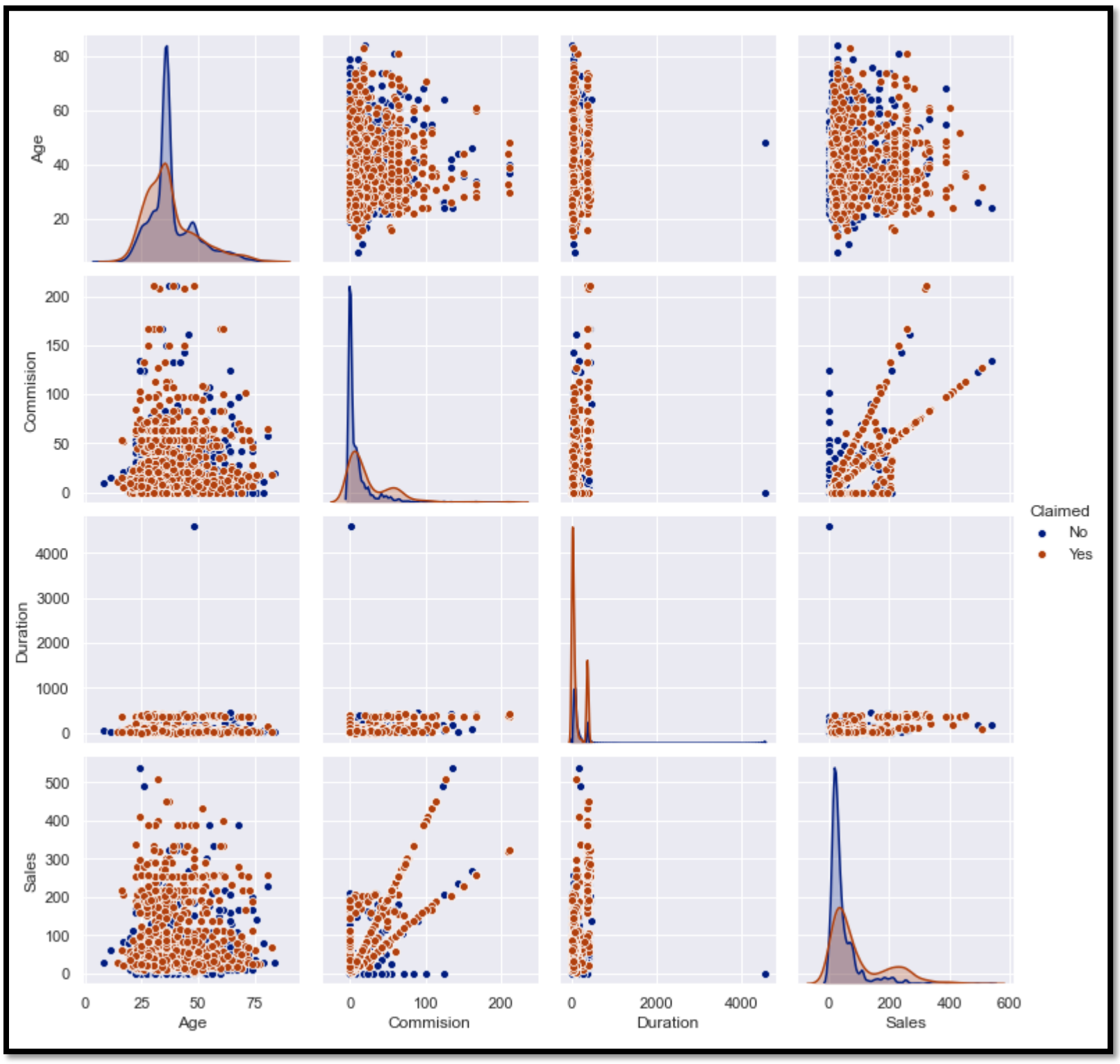

Figure 33. Pairplot of Numerical variables

From the above pairplot we can see that there is hardly any multicollinearity between the variables. Sales and Commission kind of show some positive relationship but need to check the correlation matrix to see how strong the relationship is.

Correlation Heatmap

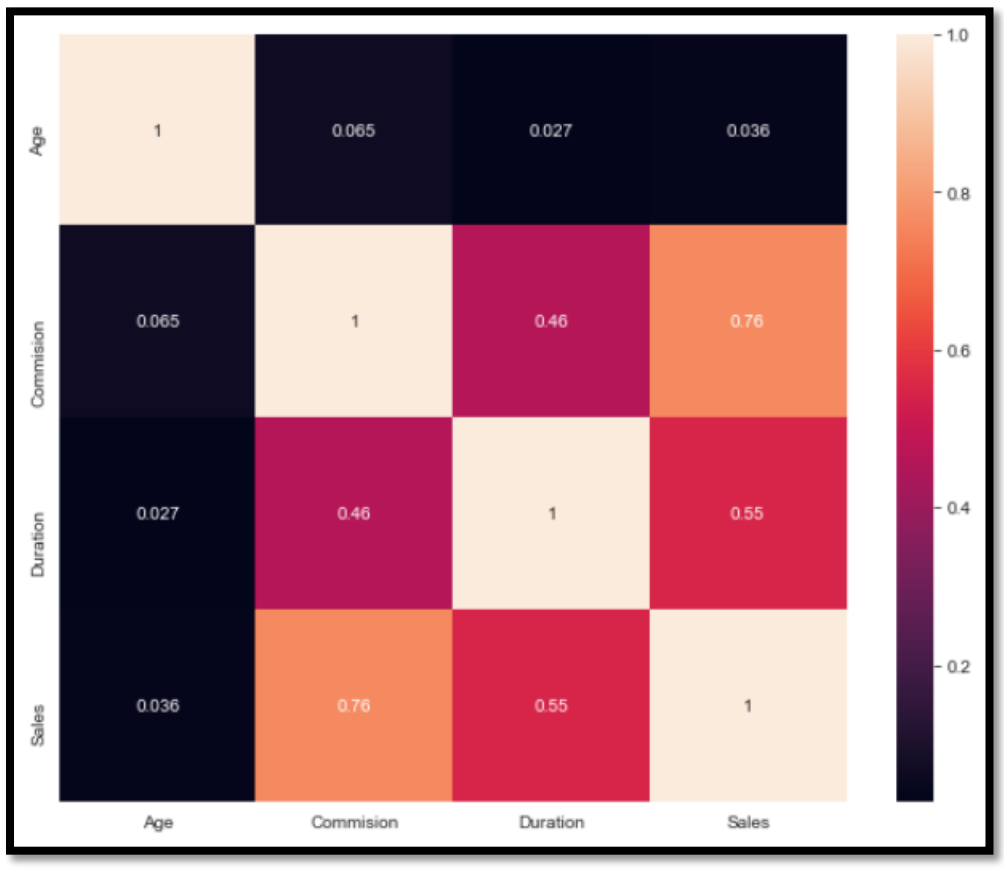

Figure 34. Correlation Heatmap

There is hardly any correlation between the variables. Sales and Commission have a positive correlation but they are not strong enough.

CART models only work on numerical data, hence let us encode the categorical variables into numerical codes.
After encoding, we can see that the data only has numeric data types which we will use for CART model building.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2861 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Age            2861 non-null   int64
 1   Agency_Code    2861 non-null   int8
 2   Type           2861 non-null   int8
 3   Claimed        2861 non-null   int8
 4   Commision      2861 non-null   float64
 5   Channel        2861 non-null   int8
 6   Duration       2861 non-null   float64
 7   Sales          2861 non-null   float64
 8   Product Name   2861 non-null   int8
 9   Destination    2861 non-null   int8
dtypes: float64(3), int64(1), int8(6)
```

Table 19. Information on the dataset

We can also see the head of the dataset.

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	0	0	0	0.70	1	7.0	2.51	2	0
1	36	2	1	0	0.00	1	34.0	20.00	2	0
2	39	1	1	0	5.94	1	3.0	9.90	2	1
3	36	2	1	0	0.00	1	4.0	26.00	1	0
4	33	3	0	0	6.30	1	53.0	18.00	0	0

Table 20. Head of the dataset

2.2 Data Split: Split the data into test and train(1 pts), build classification model CART (1.5 pts), Random Forest (1.5 pts), Artificial Neural Network(1.5 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.

CART and Random Forest (RF) models would only work on numerical data. Hence lets encode the categorical variables and transform them to numeric data.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2861 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Age            2861 non-null   int64
 1   Agency_Code    2861 non-null   int8
 2   Type           2861 non-null   int8
 3   Claimed        2861 non-null   int8
 4   Commision      2861 non-null   float64
 5   Channel        2861 non-null   int8
 6   Duration       2861 non-null   float64
 7   Sales          2861 non-null   float64
 8   Product Name   2861 non-null   int8
 9   Destination    2861 non-null   int8
dtypes: float64(3), int64(1), int8(6)
```

Table 21. Encoding Categorical variables to Numerical variables

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	0	0	0	0.70	1	7.0	2.51	2	0
1	36	2	1	0	0.00	1	34.0	20.00	2	0
2	39	1	1	0	5.94	1	3.0	9.90	2	1
3	36	2	1	0	0.00	1	4.0	26.00	1	0
4	33	3	0	0	6.30	1	53.0	18.00	0	0

Table 22. Head of Data post Encoding

Building CART / Decision Tree Model

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. It is a graphical representation of all possible solutions to a decision based on certain conditions.

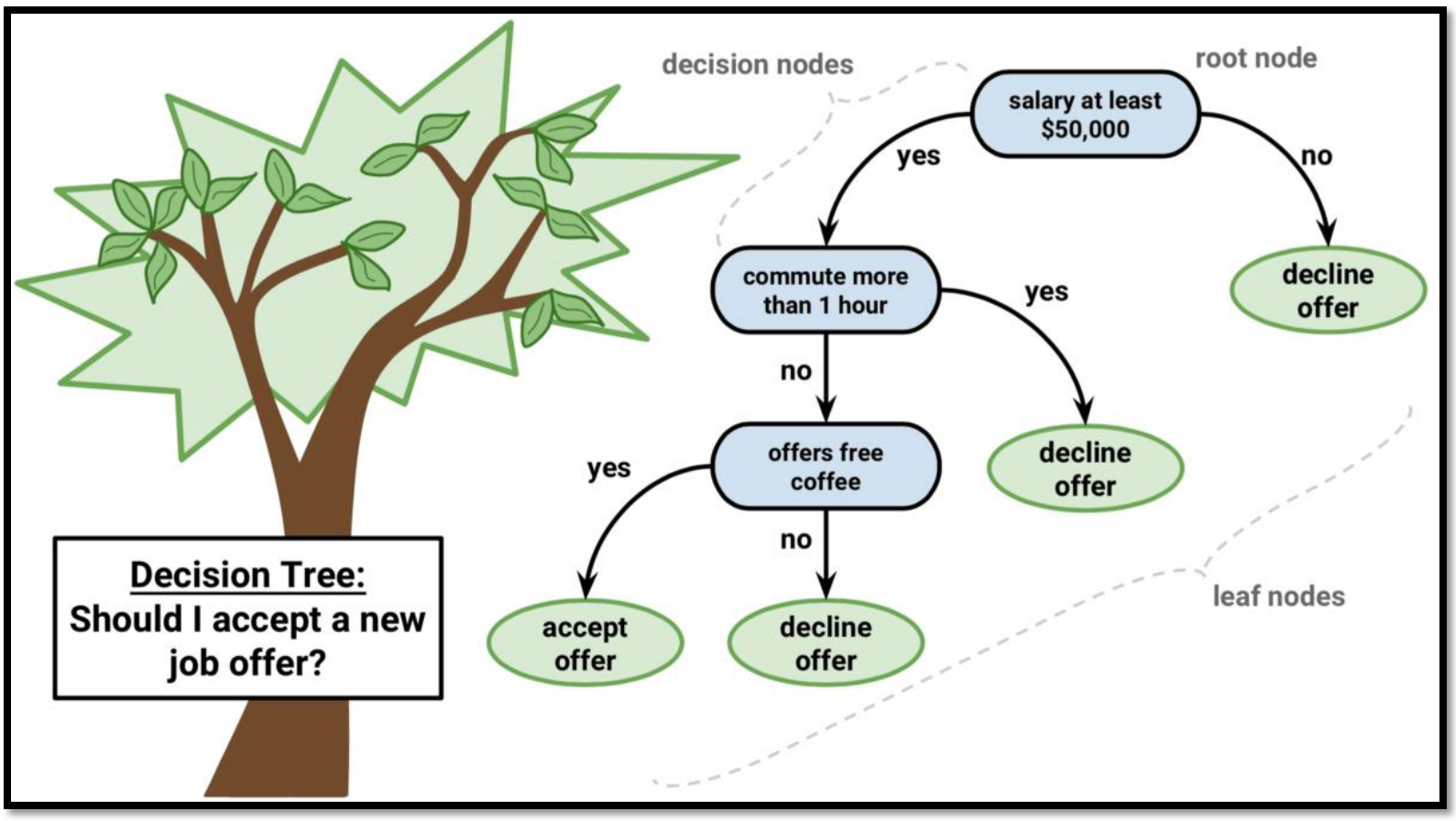

Figure 35. Example of a Decision Tree

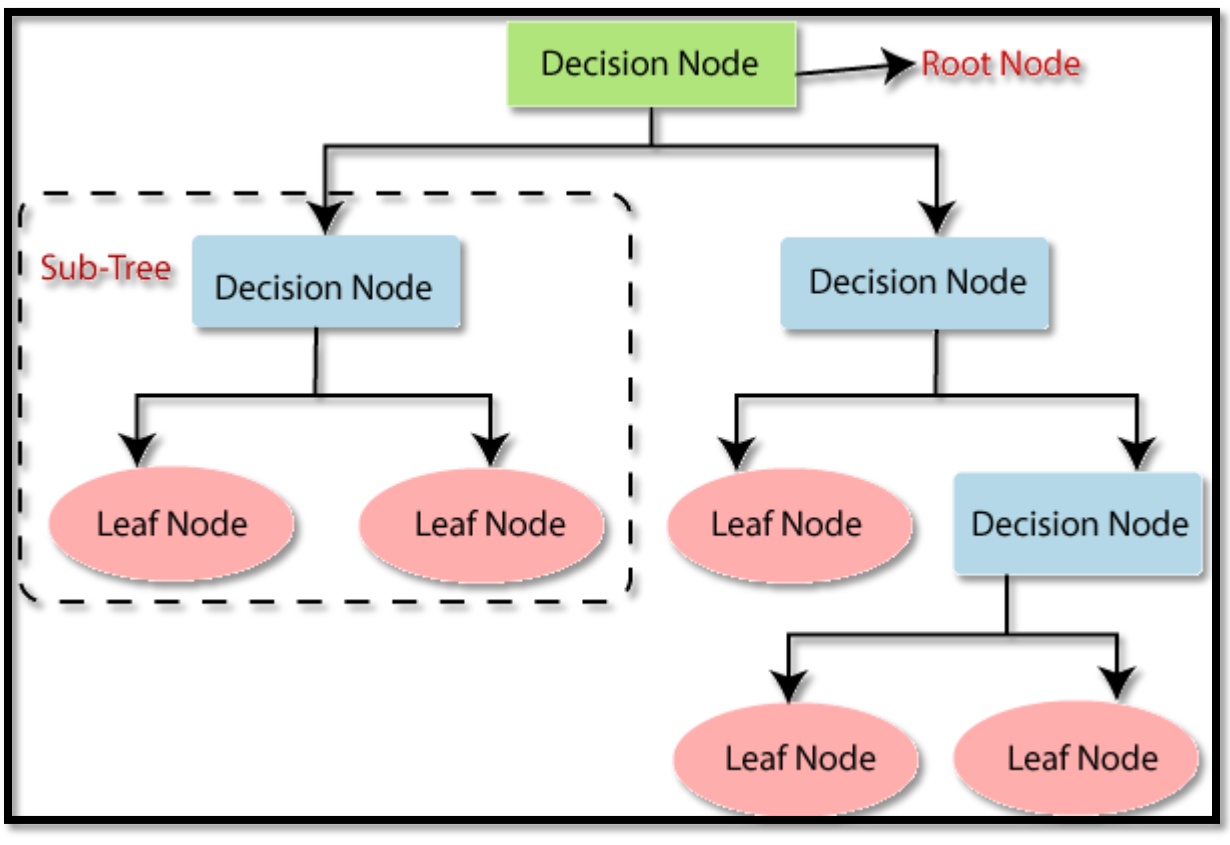

Figure 36. Decision Tree Terminology

Source: Wikipedia and Google Maps

Root Node: represent the entire population dataset which gets further divided into further nodes based on splitting decisions.
Decision Node: These are internal nodes of the tree.
Leaf Node: Nodes which do not split further are known as Leaf Node or Terminal Nodes.
Splitting: the process of dividing a node into one or more sub nodes.
Pruning: the reverse process of splitting where the sub nodes are removed.

Classification And Regression Tree (CART) is a binary tree. It uses Gini index as the calculating criteria.

Gini Index: It is calculated by subtracting the sum of squared probabilities of each class from one.

$$Gini\ Index = 1 - \sum (P(x=k))^2$$

Figure 37. Gini Index

Source: https://blog.clairvoyantsoft.com/entropy-information-gain-and-gini-index-the-crux-of-a-decision-tree-99d0cdc699f4

We will be using the sklearn model selection package to split the data into train and test data. We will use sklearn train_test_split package to split the data. We will use sklearn DecisionTreeClassifier and GridSearchCV for building Decision Tree model.

Train data will be split into 70% and test data will be split into 30%.

Hyperparameters for Decision Trees

To generate decision trees that will generalize to new problems well, we can tune different aspect about trees. We call these aspects of decision tree “hyperparameters”. Some of Important Hyperparameters used in decision trees are as follows:

Maximum Depth- The maximum depth of decision tree is simply the largest length between the root to leaf. A tree of maximum length k can have at most 2**k leaves.
Minimum number of samples per leaf- While splitting a node, one could run into the problem of having 990 samples in one of them, and 10 on the other. This will not take us too far in our process, and would be a waste of resources and time. If we want to avoid this, we can set a minimum for the number of samples we allow on each leaf.
Maximum number of feature- We can have too many features to build a decision tree. While splitting, in every split, we have to check the entire data-set on each of the features. This can be very expensive.
A solution for this is to limit the number of features that one looks for in each split. If this number is large enough, we're very likely to find a good feature among the ones we look for (although maybe not the perfect one). However, if it's not as large as the number of features, it will speed up our calculations significantly.

For the given dataset, we have taken the below hyper parameters:

Criterion: Gini
Max_Depth: 10, 13, 15
Min_Samples_Leaf: 10,50,100
Min_Samples_Split: 100,150,200
Cross Validation Iterations = 3

After applying Grid Search Cross Validation, the best parameters were as below:
Criterion: Gini
Max_Depth: 10
Min_Samples_Leaf: 50
Min_Samples_Split: 150
Now we have fit the best grid to the train and test. Let us see the feature importance.

	Importance
Agency_Code	0.609245
Sales	0.287959
Product Name	0.033844
Commision	0.029373
Duration	0.020696
Age	0.018884
Type	0.000000
Channel	0.000000
Destination	0.000000

Table 23. Feature Importance - CART

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.

Model Evaluation – CART

AUC and ROC for training data -CART

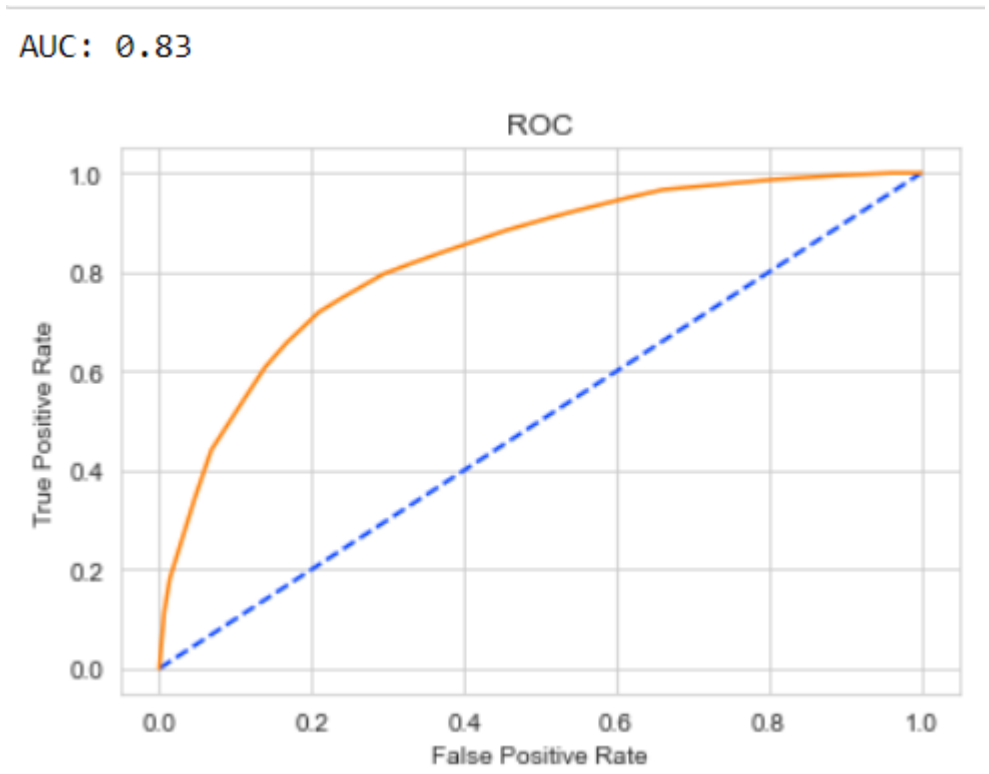

Figure 38. AUC training data - CART

AUC and ROC for testing data - CART

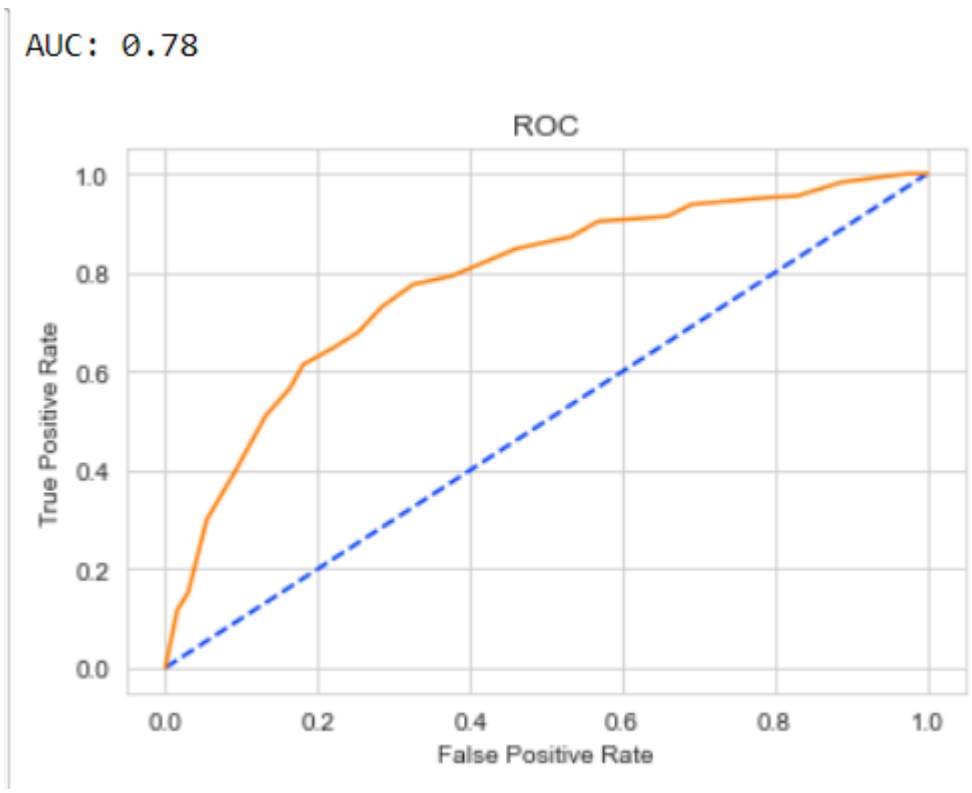

Figure 39. AUC testing data - CART

Confusion Matrix and Classification Report for training Data – CART

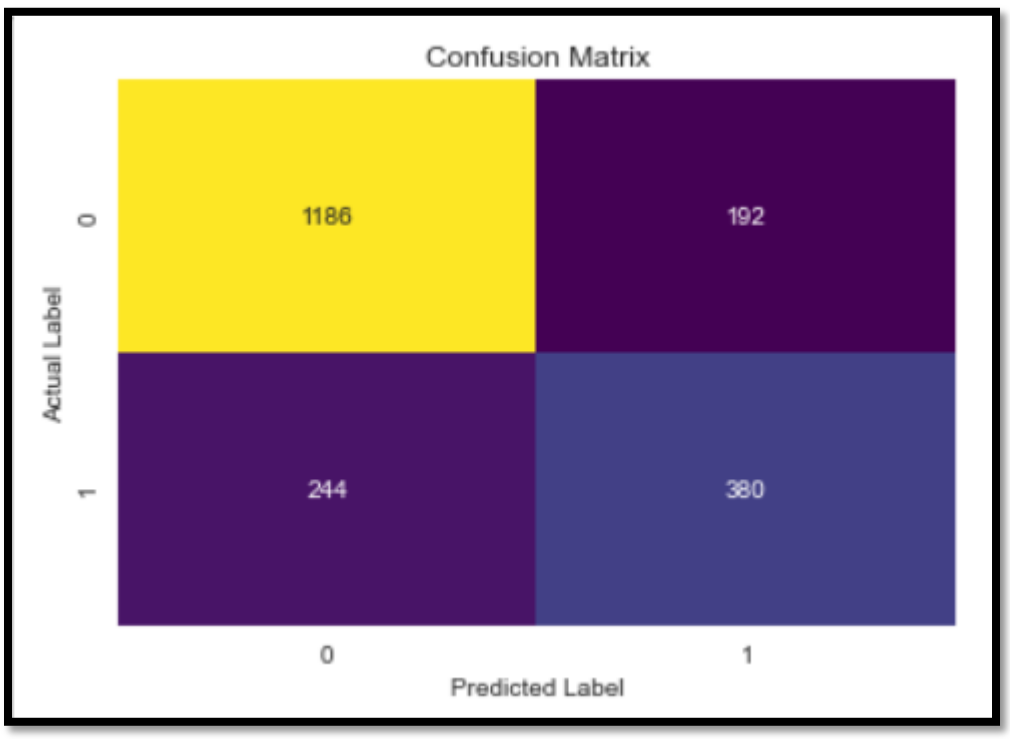

Figure 40. Confusion matrix training data – CART

	precision	recall	f1-score	support
0	0.83	0.86	0.84	1378
1	0.66	0.61	0.64	624
accuracy			0.78	2002
macro avg	0.75	0.73	0.74	2002
weighted avg	0.78	0.78	0.78	2002

Table 24. Classification report training data - CART

Confusion Matrix and Classification Report for testing data – CART

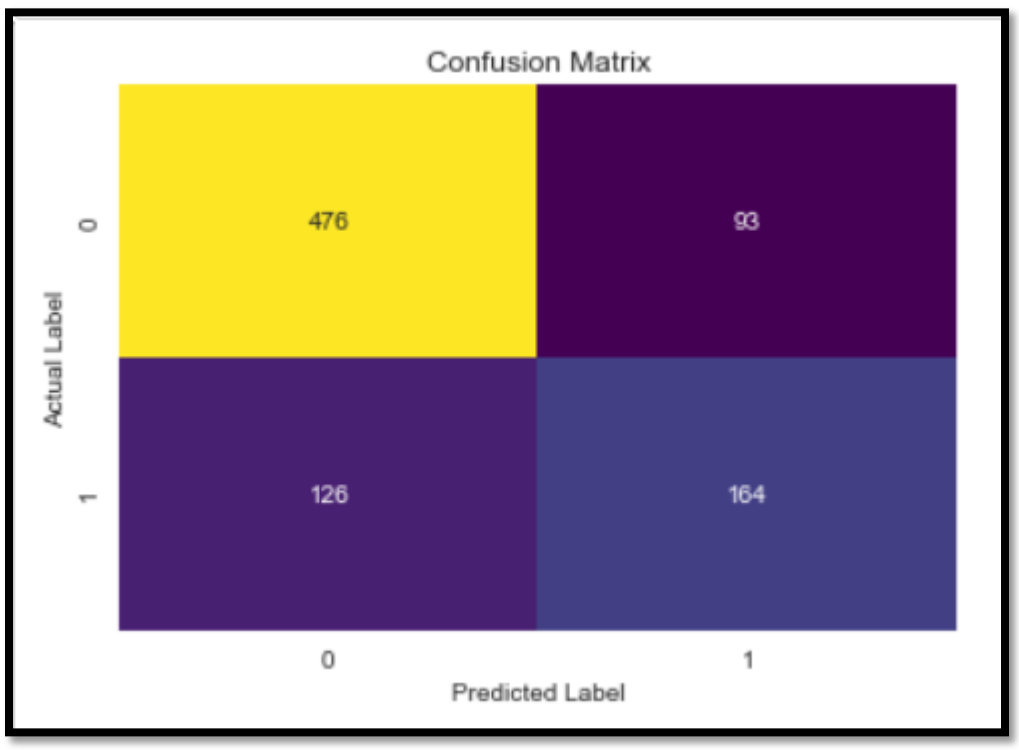

Figure 41. Confusion matrix testing data - CART

	precision	recall	f1-score	support
0	0.79	0.84	0.81	569
1	0.64	0.57	0.60	290
accuracy			0.75	859
macro avg	0.71	0.70	0.71	859
weighted avg	0.74	0.75	0.74	859

Table 25. Classification report testing data – CART

Building Random Forest Model

Random Forests is an ensemble machine learning technique that combines several base models in order to produce one optimal predictive model. Random Forests are a collection of decision trees. In random forests, each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set.
Furthermore, when splitting each node during the construction of a tree, the best split is found either from all input features or a random subset of size max_features.
The purpose of these two sources of randomness is to decrease the variance of the forest estimator. Indeed, individual decision trees typically exhibit high variance and tend to overfit. The injected randomness in forests yield decision trees with somewhat decoupled prediction errors. By taking an average of those predictions, some errors can cancel out. Random forests achieve a reduced variance by combining diverse trees, sometimes at the cost of a slight increase in bias. In practice the variance reduction is often significant hence yielding an overall better model.

Random Forest Algorithm

Draw multiplerandom samples, with replacement, from the data. This sampling approach is called the bootstrap. Using a random subset of predictors at each stage, fit a classification (or regression) tree to each sample (and thus obtaina “forest”). Combine the predictions/classifications from the individual trees to obtain improved predictions. Use voting for classification and averaging for prediction.

Out-Of-Bag (OOB) Dataset

When we create a bootstrapped dataset, 1/3 of the original data does not end up in the bootstrapped dataset. This is called Out-Of-Bag dataset. OOB samples are used to measure how accurate our random forest is.

For the given dataset, we have taken the below hyper parameters:

Criterion: Gini
Max_Depth: 10
Max_Features: 5
Min_Samples_Leaf: 10
Min_Samples_Split: 100
n_estimators: 201
Cross Validation Iterations = 3

After applying Grid Search Cross Validation, the best parameters were as below:
Max_Depth: 10
Max_Features: 5
Min_Samples_Leaf: 10
Min_Samples_Split: 100
n_estimators: 201
Cross Validation Iterations = 3

Now we have fit the best grid to the train and test. Let us see the feature importance.

	Importance
Agency_Code	0.369611
Product Name	0.201126
Sales	0.182622
Commision	0.090019
Duration	0.069473
Type	0.042939
Age	0.033528
Destination	0.007500
Channel	0.003182

Table 26. Feature Importance - RF

Model Evaluation – Random Forest

AUC and ROC for Training data - Random Forest

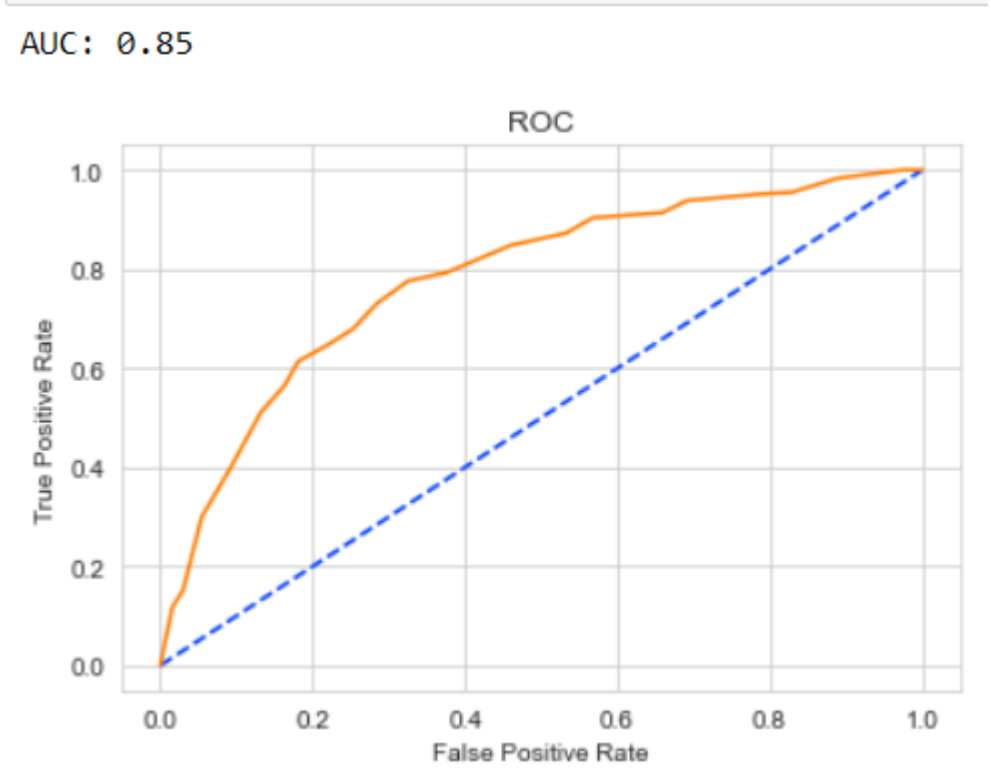

Figure 42. AUC training data – Random Forest

AUC and ROC for Testing data - Random Forest

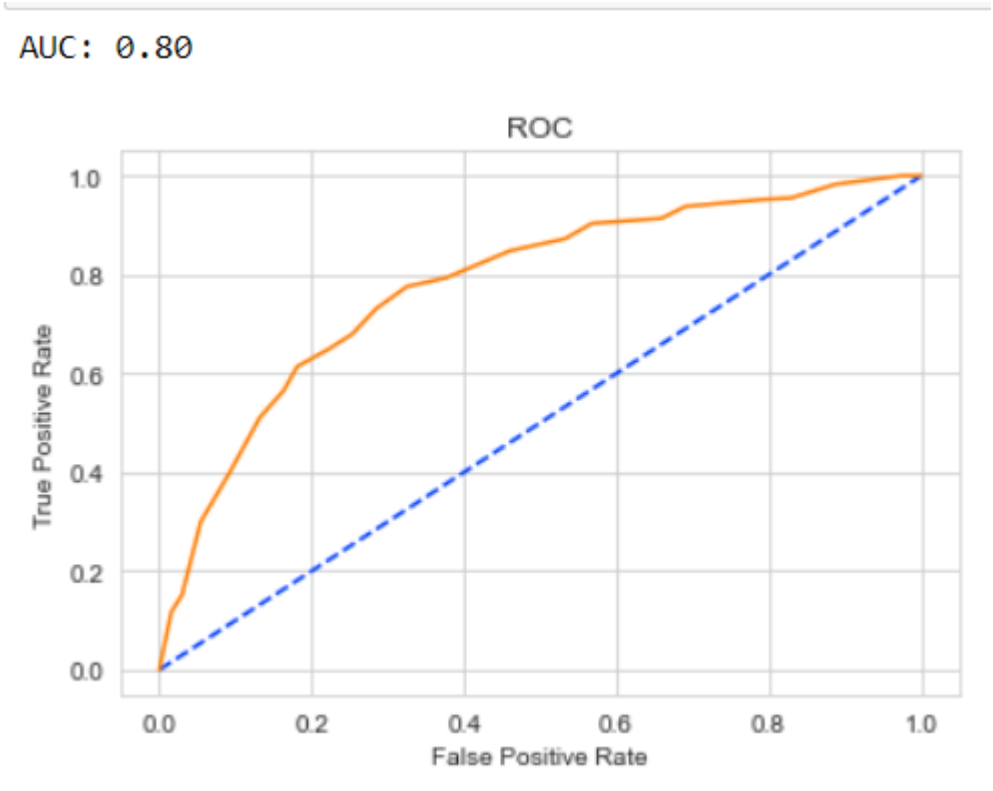

Figure 43. AUC testing data – Random Forest

Confusion Matrix and Classification Report for Training Data - Random Forest

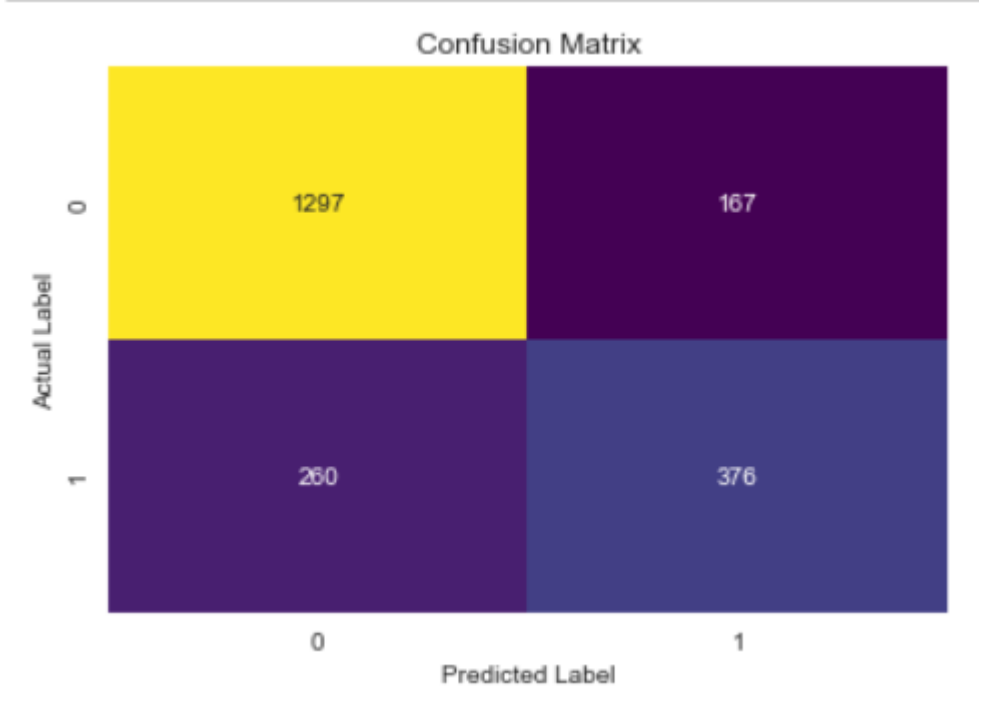

Figure 44. Confusion matrix training data – Random Forest

	precision	recall	f1-score	support
0	0.83	0.89	0.86	1464
1	0.69	0.59	0.64	636
accuracy			0.80	2100
macro avg	0.76	0.74	0.75	2100
weighted avg	0.79	0.80	0.79	2100

Table 27. Classification report training data – Random Forest

Confusion Matrix and Classification Report for Testing data - Random Forest

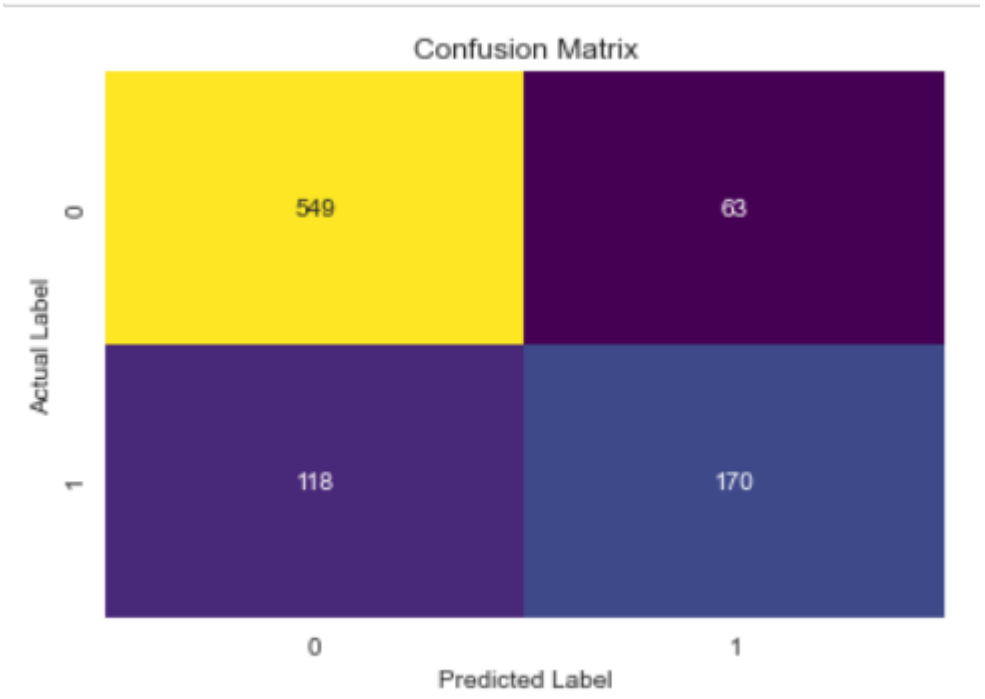

Figure 45. Confusion matrix testing data - Random Forest

	precision	recall	f1-score	support
0	0.82	0.90	0.86	612
1	0.73	0.59	0.65	288
accuracy			0.80	900
macro avg	0.78	0.74	0.76	900
weighted avg	0.79	0.80	0.79	900

Table 28. Classification report testing data – Random Forest

Building an Artificial Neural Network (ANN) Model

Artificial Neural Network (ANN) is machine learning algorithm that is roughly modeled around what is currently known about how the human brain functions. It models the relationship between a set of input signals and an output similar to a biological brain response to stimuli from sensory inputs. The brain uses a network of interconnected cells called neurons to provide learning capability. ANN uses a network of artificial neurons or nodes to solve challenging learning problems. Below is the comparison between brain network and ANN.

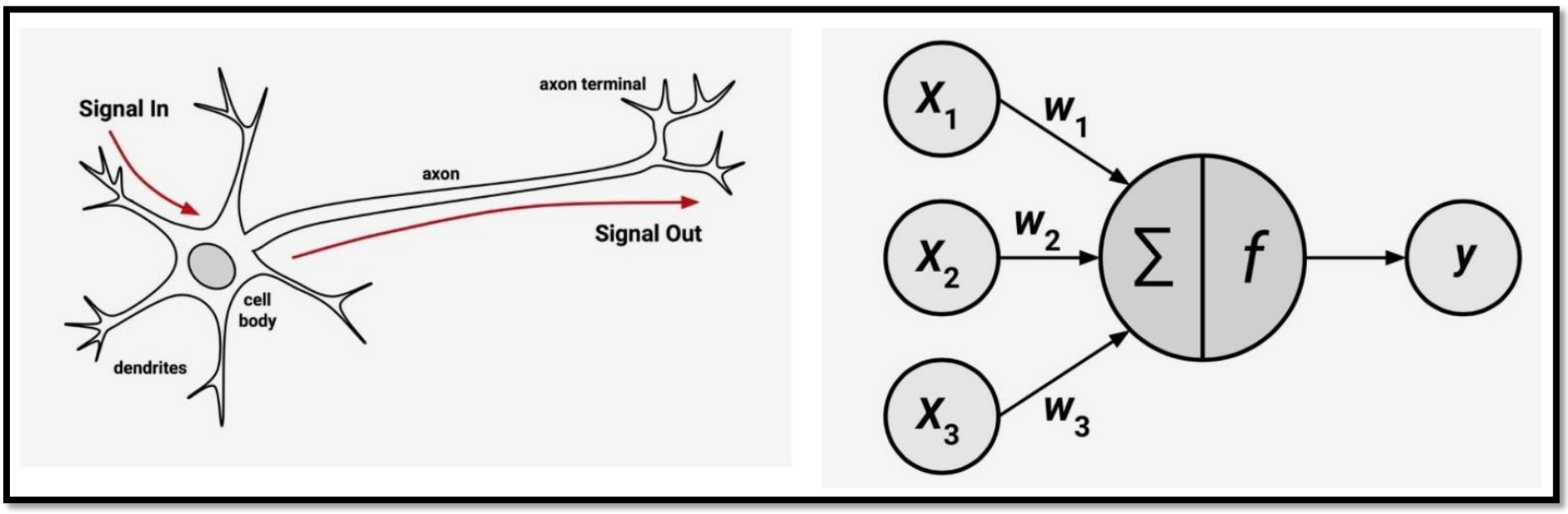

Figure 46. Artificial Neural Network (ANN)

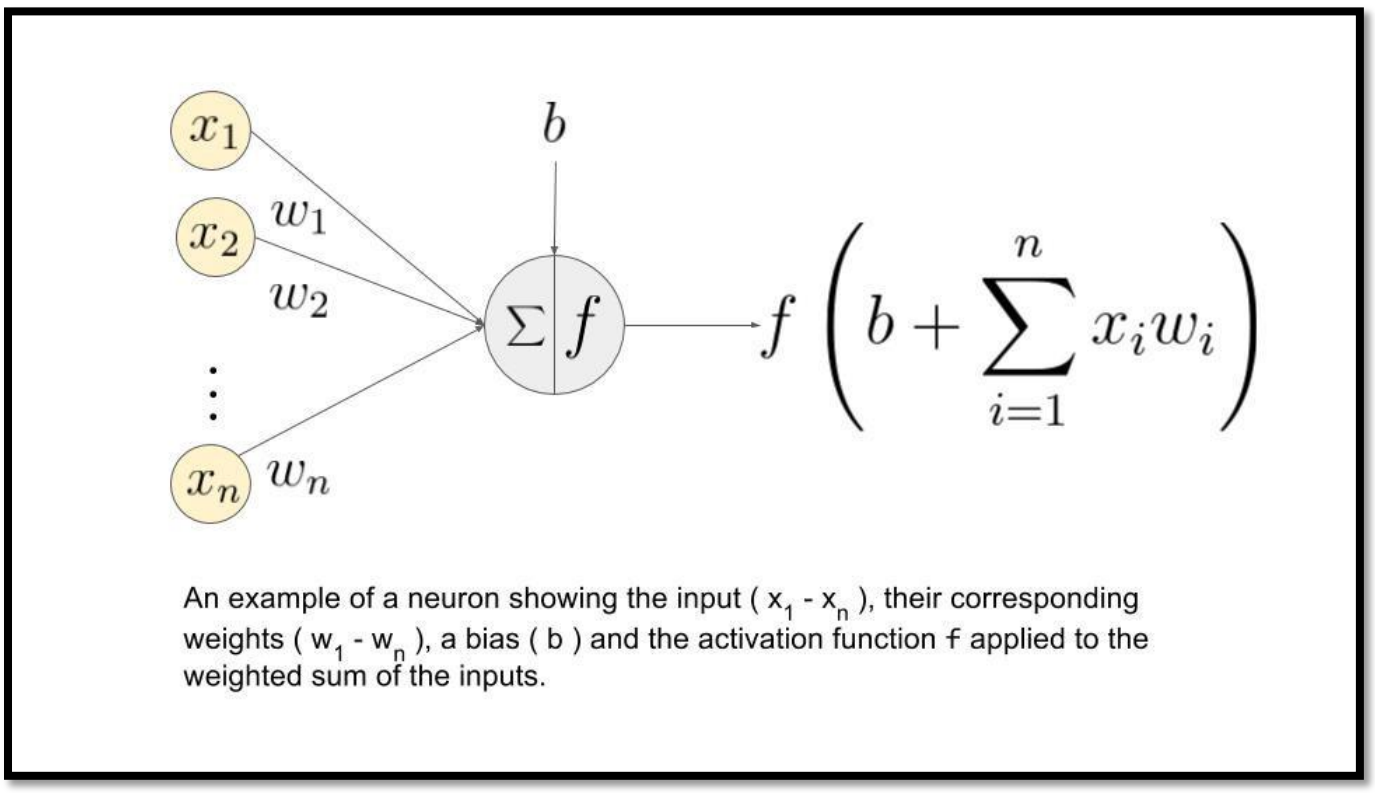

Figure 47. ANN Formula

Source: Google Images

ANN Architecture

ANN architecture is made of layers with many interconnected nodes (neurons).

There are three main layers, specifically
InputLayer
HiddenLayer (Hidden Layer can be one or more)
OutputLayer

ANN Neurons

A neuron is an information-processing unit that is fundamental to the operation of a neural network.

Three basic elements of the neuron model:

Synaptic weights
Combination (Addition) function
Activation function

External input bias to increase or lower the net input to the Activation function.

Activation Function

It is a mechanism by which the artificial neuron processes incoming information and passes it throughout the network. It is a threshold activation function as it results in an output signal only once a specified input threshold has been attained.
The different types of Activation Function are as below:

Unit Step Activation function
Sigmoid Activation function
Hyperbolic Tangent Activation function
Rectified Linear Unit Activation function (RELU)

Learning Rate

Learning Rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated.
-Choosing the learning rate is challenging.
-a value too small may result in a long training process that could get stuck.
-a value too large may result in learning a sub-optimal set of weights too fast or an unstable training process.

We will be using the sklearn neural network Multi-Layer Perceptron Classifier to build this model. ANNs are effective when the data is scaled. Hence we have scaled the data using sklearn Standard Scalar package.

We have used the below parameters while building the model.
Hidden Layer Size: 3,5,7
Max_Iteration: 2500
Solver: adam, sgd, lbfgs
Tolerance: 0.01, 0.001

After applying Grid Search Cross Validation, the best parameters were as below:
Hidden Layer Size: 7
Max_Iteration: 2500
Solver: lbfgs
Tolerance: 0.01

Model Evaluation - ANN

AOC and ROC for Training data - ANN

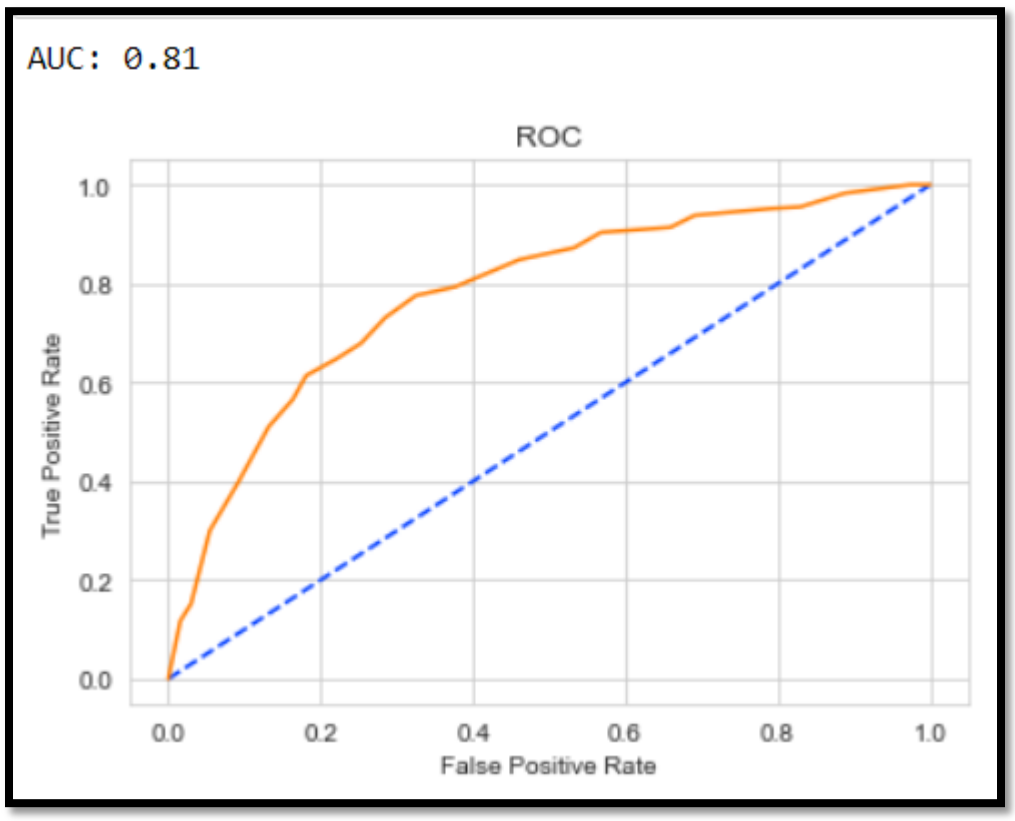

Figure 48. AUC training data - ANN

AOC and ROC for Testing data - ANN

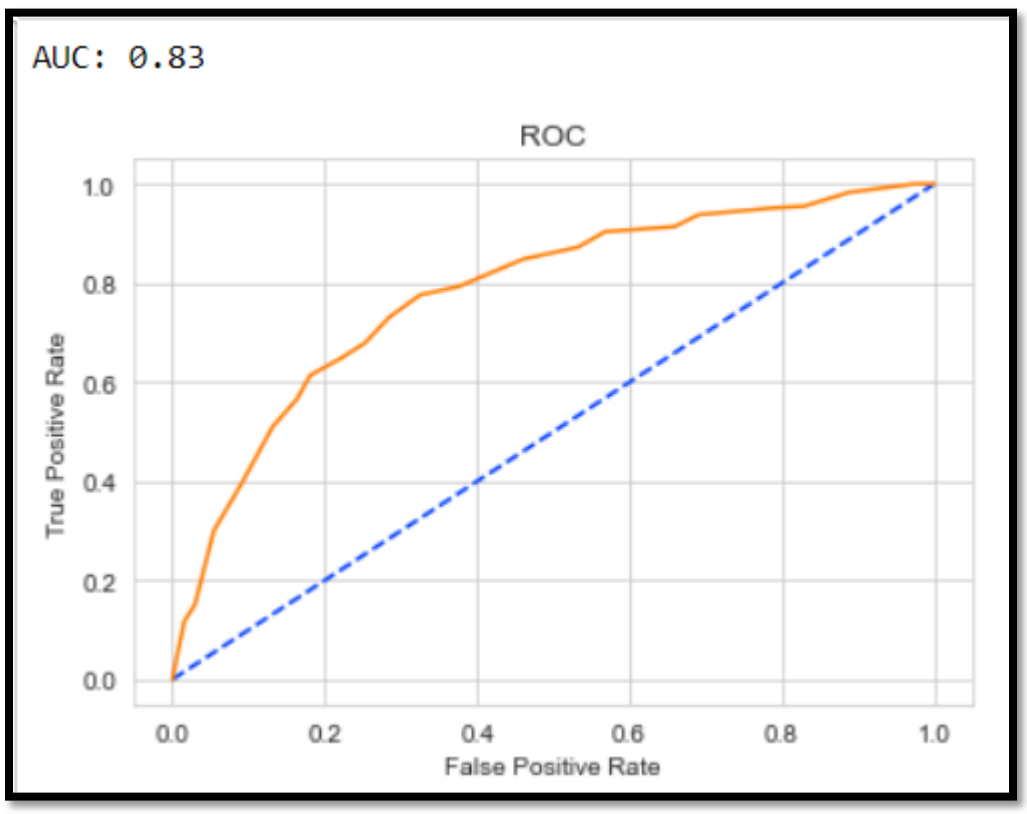

Figure 49. AUC testing data - ANN

Confusion Matrix and Classification Report for Training data - ANN

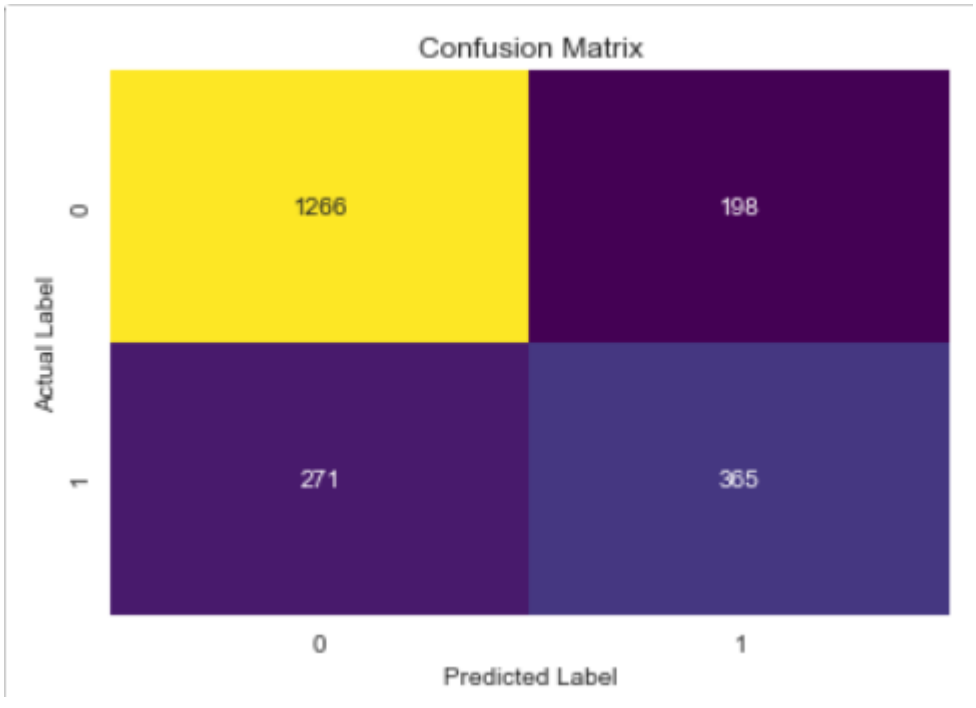

Figure 50. Confusion matrix training data - ANN

	precision	recall	f1-score	support
0	0.82	0.86	0.84	1464
1	0.65	0.57	0.61	636
accuracy			0.78	2100
macro avg	0.74	0.72	0.73	2100
weighted avg	0.77	0.78	0.77	2100

Table 29. Classification report training data – ANN

Confusion Matrix and Classification Report for Testing data - ANN

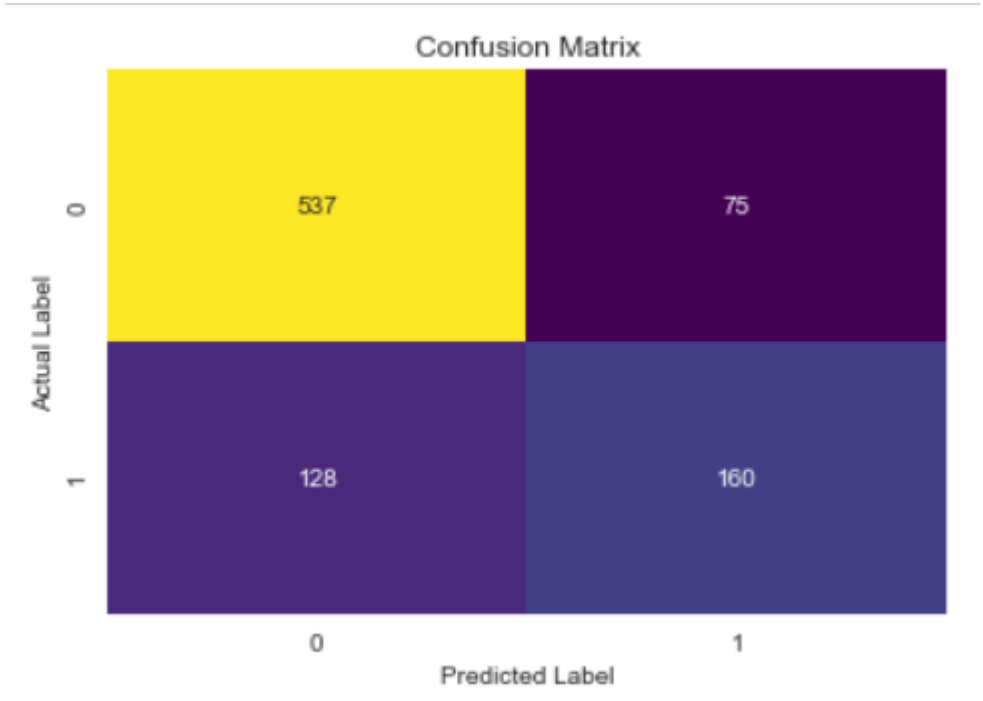

Figure 51. Confusion matrix testing data - ANN

	precision	recall	f1-score	support
0	0.81	0.88	0.84	612
1	0.68	0.56	0.61	288
accuracy			0.77	900
macro avg	0.74	0.72	0.73	900
weighted avg	0.77	0.77	0.77	900

Table 30. Classification report testing data – ANN

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.

Below is the comparison of all the 3 models with their recall, precision, accuracy, f1-score and AUC (Area Under the Curve).

	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.78	0.75	0.80	0.80	0.78	0.77
AUC	0.83	0.78	0.85	0.85	0.81	0.83
Recall	0.61	0.56	0.59	0.59	0.57	0.56
Precision	0.66	0.68	0.69	0.73	0.65	0.68
F1 Score	0.64	0.61	0.64	0.65	0.61	0.61

Table 31. Comparison of 3 models

ROC Curve for the 3 models on the Training data

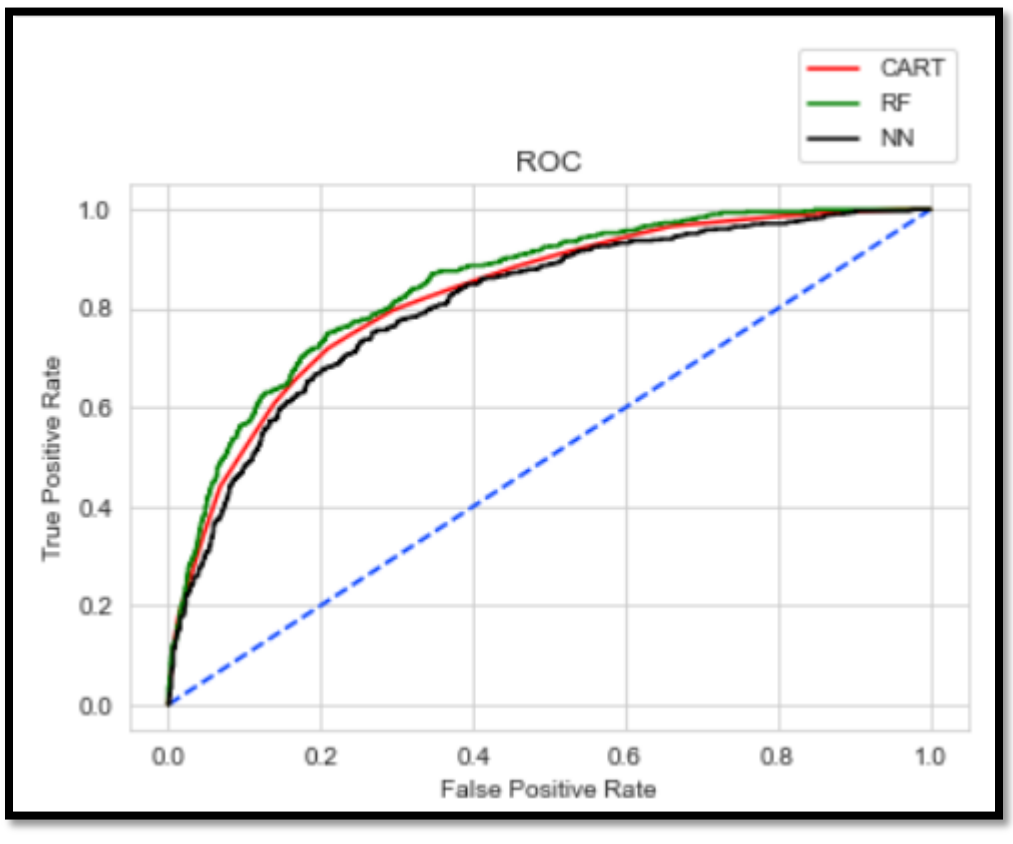

Figure 52. ROC for 3 models training data

ROC Curve for the 3 models on the Testing data

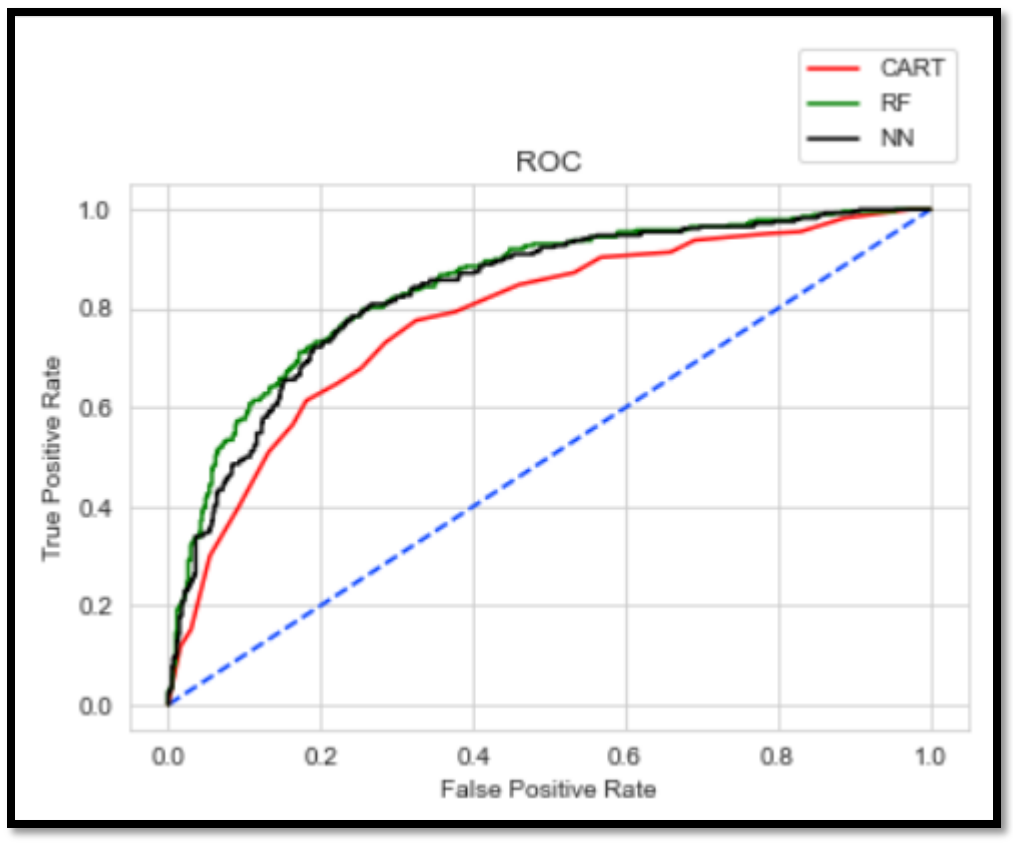

Figure 53. ROC for 3 models testing data

Out of the 3 models, Random Forest has slightly better performance than the CART and ANN model.

Overall all the 3 models are reasonably stable enough to be used for making any future predictions. From CART and Random Forest Model, the variable Agency_Code is found to be the most useful feature amongst all other features for predicting if a customer has claimed the insurance or not.

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.

Based on the above 3 models, below are the recommendations.

1. Out of 2861 observations after removing duplicates, the proportion of claimed status “No” is 1947 whereas “Yes” is 914 which clearly shows that No is 68.05% of the total data and Yes is 31.94% of the total data. Hence there is a clear imbalance in the proportion of class labels.
2. Business needs to collect more data in order to balance the proportions and thereby build an effective model.
3. The business should focus on Agency_Codes which is the main attribute on which these models are built. They should look at different Agency_Codes and see if there is a pattern hidden in terms of insurance claimed.

THE END !!!