Data Mining Project

by

Kajal Dusseja

PGP DSBA

Febuary'22

19-06-2022

greatlearning
Learning for Life

Table of Contents

Problem 1

Problem Statement:

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

Dataset Dictionary:

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

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

Solution:

Dataset is imported using read_csv function from pandas library.

Information of 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)
memory usage: 11.6 KB
```

Figure 1.1.1: Information of Bank Dataset

Summary of Dataset:

	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

Figure 1.1.2: Summary of Bank Dataset

Dimension of dataset:

The dataset dimension is checked using the shape property of dataset. The dataset has 210 rows and 7 columns.

```
Out[6]: (210, 7)
```

Figure 1.1.3: Dimension of Bank Dataset

Null Value Check:

Null or missing value check needs to be performed because the null values can cause errors or disparities in results. The null values need to be dropped if any. The check is performed using isnull() function of pandas library

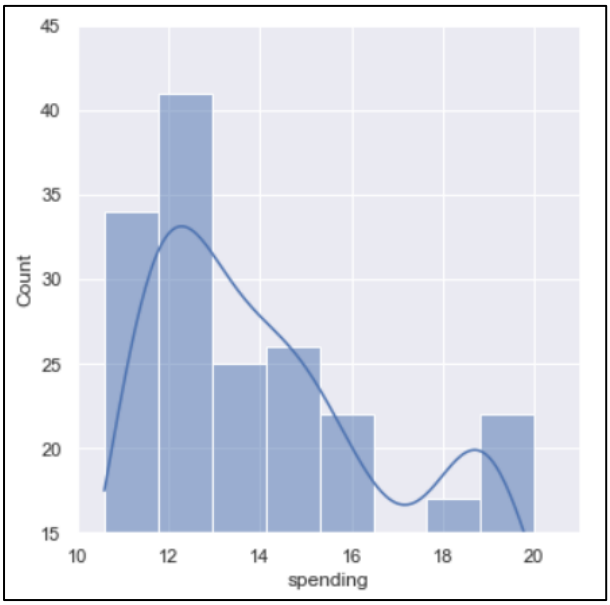
```
Out[7]: spending                      0
        advance_payments              0
        probability_of_full_payment   0
        current_balance               0
        credit_limit                  0
        min_payment_amt               0
        max_spent_in_single_shopping  0
        dtype: int64
```

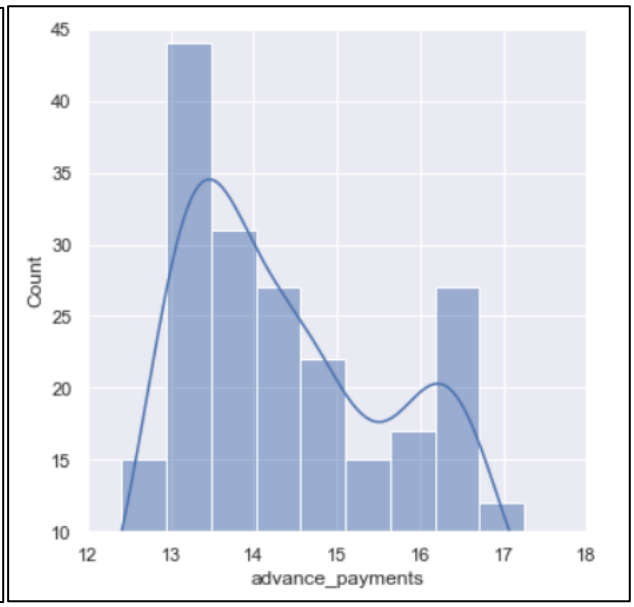
Figure 1.1.4: Null or missing values in Bank Dataset

As can be seen from the output, there are no null or missing values in dataset.

Univariate Analysis:

Histogram is plotted for all features of dataset for Univariate analysis. Boxplot to check for the outliers in dataset, outliers if exist need to be removed otherwise leads to misinterpretation of data.





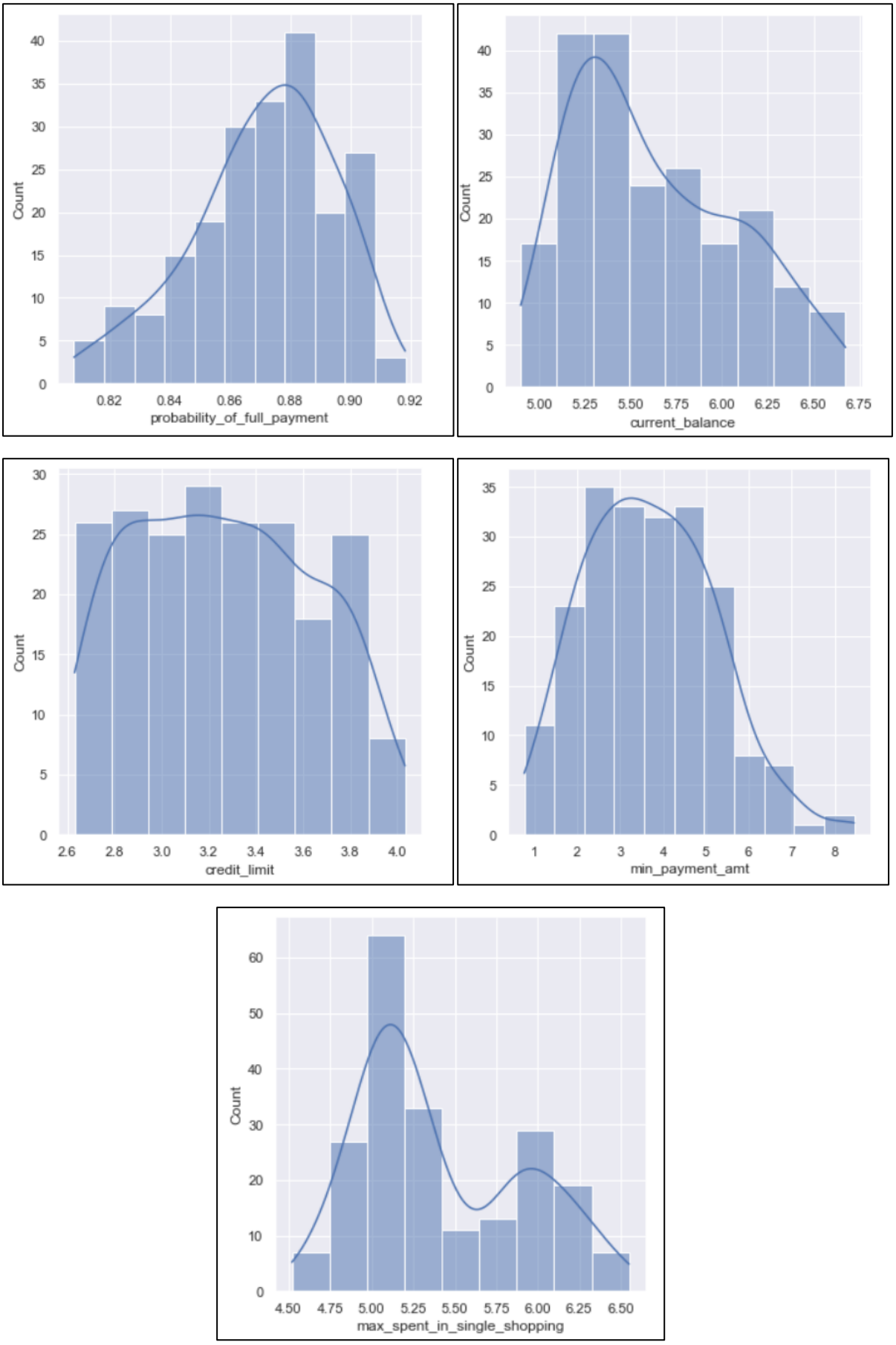


Figure 1.1.5: Histogram plot of all features of Bank Dataset

Boxplot for Outliers:

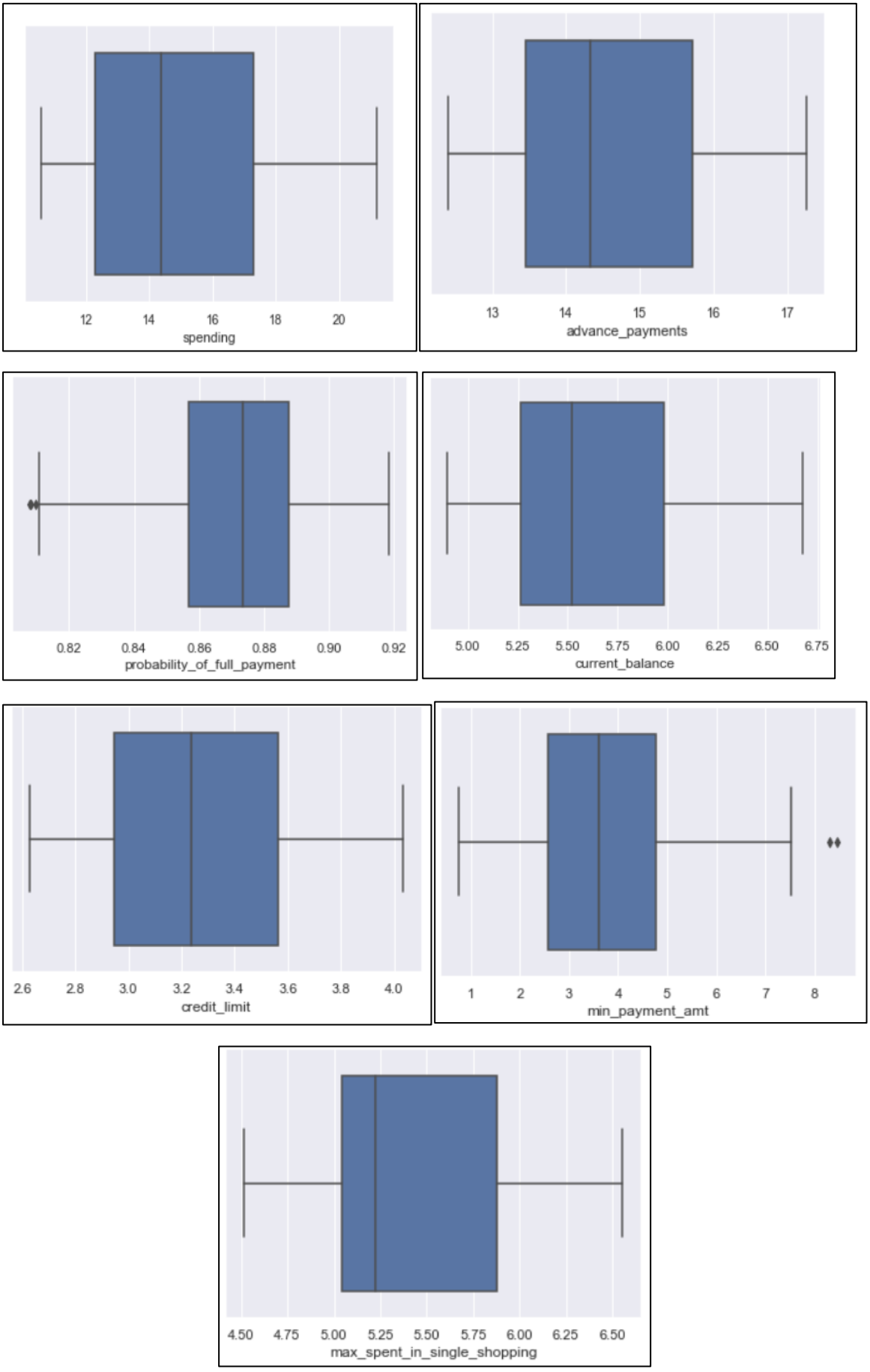


Figure 1.1.6: Boxplot to check outliers in Bank Dataset

Univariate Inference:

From histogram plots of the features, it can be seen that

- Most of the customers spend in the range of 10-13K per month. The max spent by high end customers is 20K per month.
- Majority of the customers pay to bank in advance and in the range of 12-17K.
- Probability of full payment by the customers to bank is above 80%.
- Most customers have current account balance in range of 5 to 5.5L.
- Customers have a credit limit of up to 40K.
- Minimum payment of up to 8K is done by customers to bank.
- Customers are spending up to 6K in shopping with majority customers spending up to 5K.

Overall, it can be interpreted from plots that customers repayment to bank shows a good score and most customers have higher spending capacity, higher current bank balance and customers spent heavily on shopping.

From boxplots, we can see outliers only exist for minimum payment amount which shows that there are few customers whose repayment falls on higher side. Since one out of seven features has outliers there is no need to treat the outliers as such small value will not create a huge difference.

Bivariate Analysis:

Pairplot is used for Bivariate analysis, to analyze and interpret behaviors, relationships of two features from dataset.

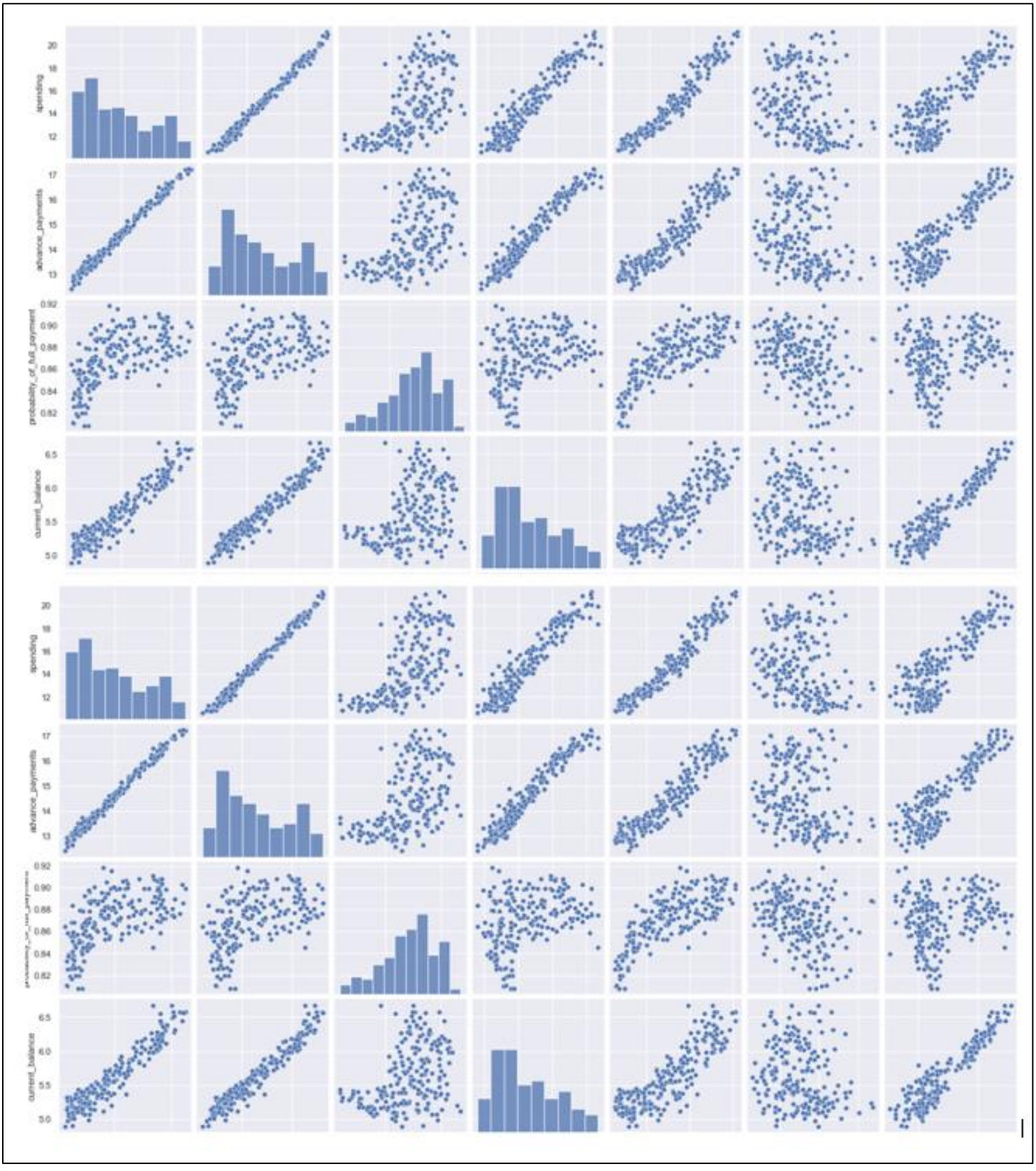


Figure 1.1.7: Pairplot for Bivariate Analysis of Dataset

Multivariate Analysis:

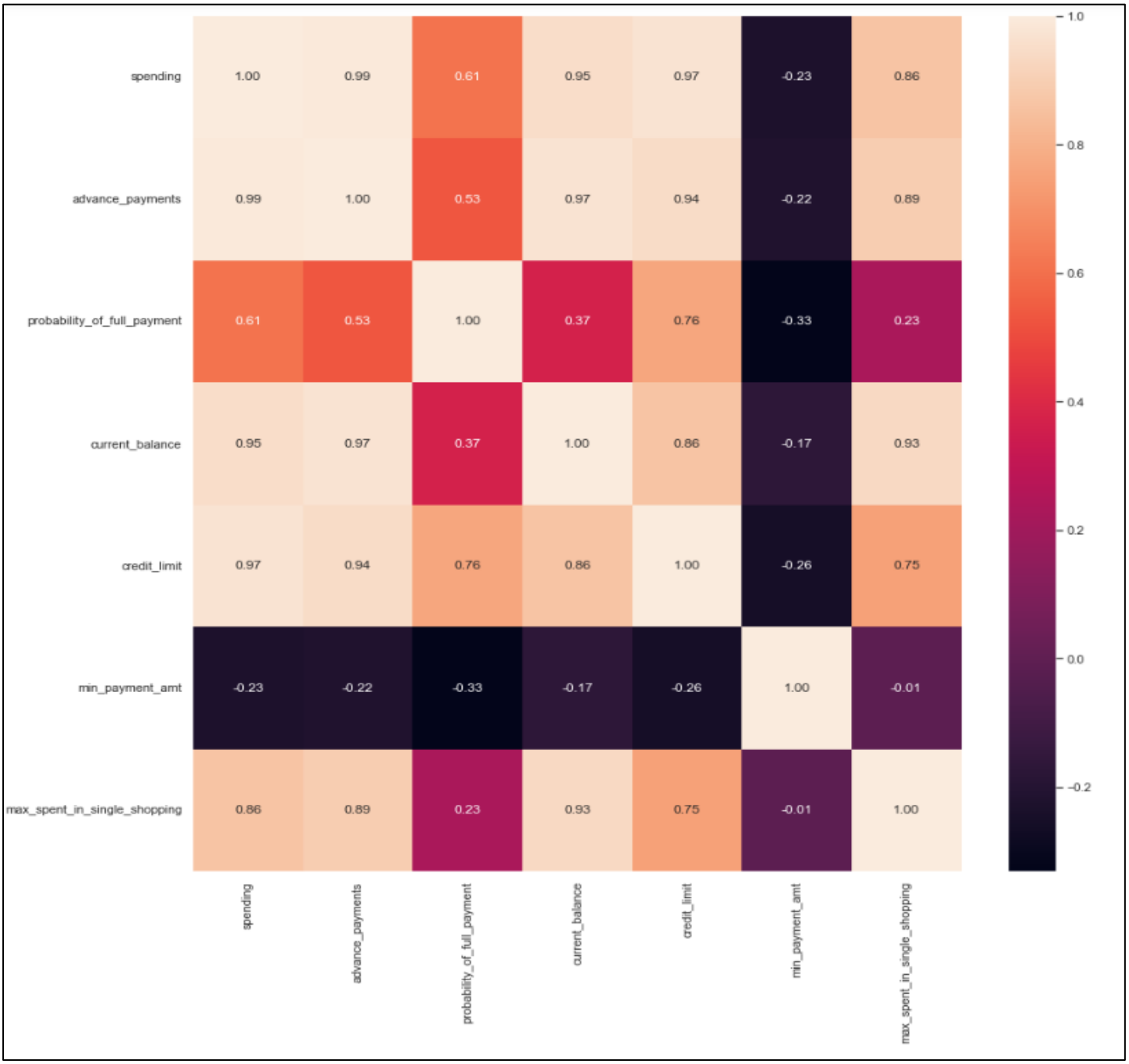


Figure 1.1.8: Heatmap of dataset

Multivariate Inference:

As per the Heat Map, we can see that following variables are highly correlated,

- Spending and advance payments, spending and current balance, spending and credit limit
- Advance payment and current balance, advance payment and credit limit
- Current balance and max spent in single shopping

By this we can conclude that the customers who are spending very high have a higher current balance and high credit limit. Advance payments and maximum expenditure done in single shopping are done by majority of those customers who have high current balance in their bank accounts. Probability of full payments are higher for those customers who have a higher credit limit. Minimum payment amount is not correlated to any of the variables, hence, it is not affected by any changes in the current balance or credit limit of the customers.

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

Scaling is done to pre-process the data, to bring down the data to one scale. If scaling is ignored then algorithms tend to weigh the higher values as higher and lower values as lower even though the unit is same.

The customer data provided has different units for variables, spending is in unit of 1000's, credit limit in 10000, advance payments in 100's and probabilities of full payment in decimal. Probability of full payment will be outweighed by other features hence it's important to scale the data using Standard Scaler so the means of data is 0 and standard deviation 1. Following is the output of dataset after scaling.

```
array([[ 1.75435461,  1.81196782,  0.17822987, ...,  1.33857863,
        -0.29880602,  2.3289982 ],
       [ 0.39358228,  0.25383997,  1.501773  , ...,  0.85823561,
        -0.24280501, -0.53858174],
       [ 1.41330028,  1.42819249,  0.50487353, ...,  1.317348  ,
        -0.22147129,  1.50910692],
       ...,
       [-0.2816364 , -0.30647202,  0.36488339, ..., -0.15287318,
        -1.3221578 , -0.83023461],
       [ 0.43836719,  0.33827054,  1.23027698, ...,  0.60081421,
        -0.95348449,  0.07123789],
       [ 0.24889256,  0.45340314, -0.77624835, ..., -0.07325831,
        -0.70681338,  0.96047321]])
```

Figure 1.2.1: Output of scaled bank data

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

Solution:

Clustering is unsupervised learning technique and is divided in Hierarchical and K-Means Clustering. Hierarchical clustering groups similar data into set of clusters and the end results is set of clusters with distinct features but the objects within cluster have same features. There are two types of Hierarchical clustering, Agglomerative and Divisive.

For the dataset provided, Agglomerative clustering is used to identify clusters and create dendograms.

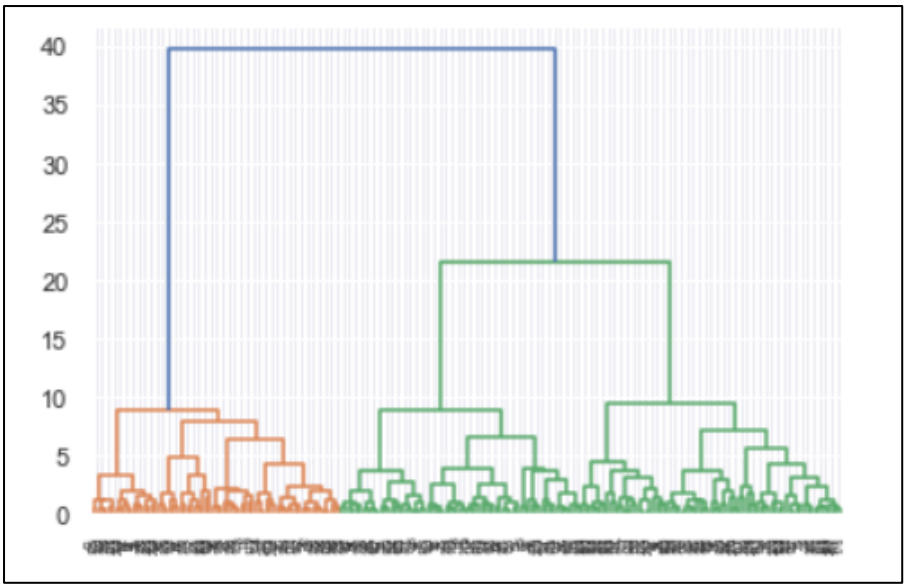


Figure 1.3.1: Dendrogram with all linkages

The above figure shows the initial dendrogram created with all the linkages. As can be seen two clusters are created for the data.

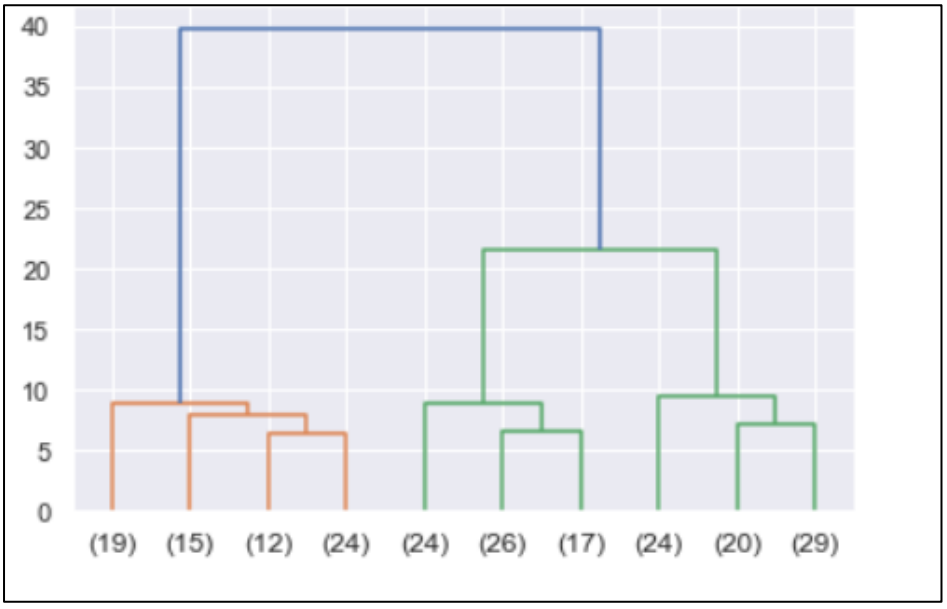


Figure 1.3.2: Dendrogram with last 10 linkages

Using the 'maxclust' and 'distance' from fcluster function, the creation of above two clusters is checked.

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

Figure 1.3.3: Cluster creation using maxclust criteria of fcluster

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

Figure 1.3.4: Cluster creation using distance criteria of fcluster

The above two figures show the assignment of each customer to a cluster. The data is segregated into two clusters using agglomerative clustering.

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

Solution:

K-Means clustering is also an unsupervised learning algorithm. K-Means clustering focuses on creating centroids and distributing the objects around the centroids. The algorithm aims to keep the centroid as small as possible for better clustering.

For the dataset provided, K-Means is applied on scaled dataset to form clusters and use those clusters to devise tools for each group.

We will plot two curves to determine optimal number of clusters to be used. The two methods being within sum of squares(wss) and average silhouette scores

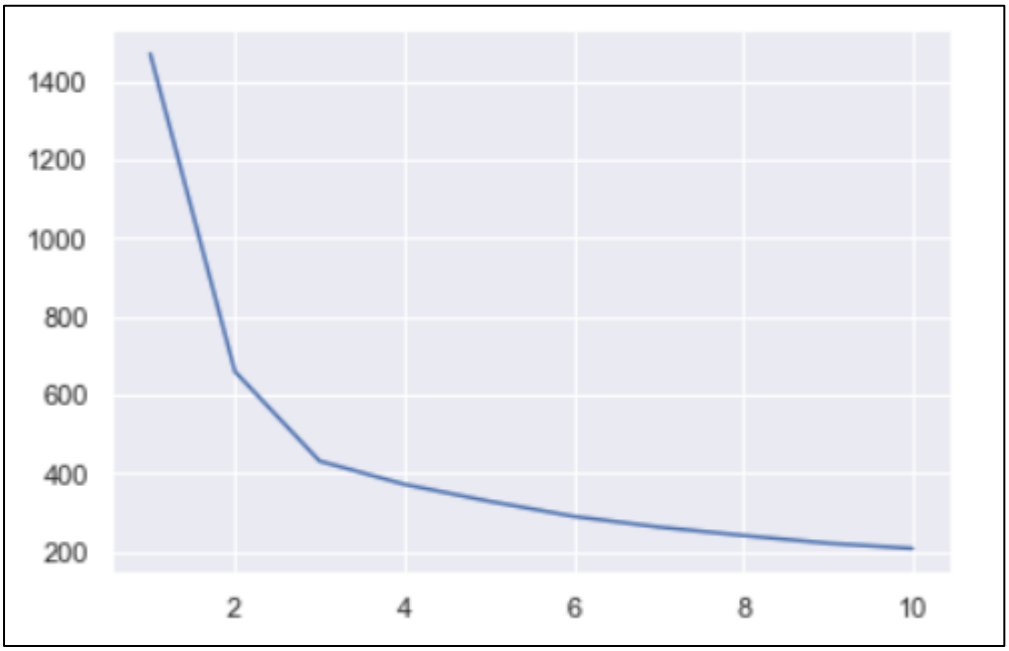


Figure 1.4.1: K-Means wss curve

As seen from the above curve plotted through within sum of square(wss) method, the optimal number of clusters is 3 as the graph starts to decline and become almost flat after 3.

```
For cluster = 2  The average silhouette score is  0.5717487774387302
For cluster = 3  The average silhouette score is  0.6070196300166342
For cluster = 4  The average silhouette score is  0.5246588046460366
For cluster = 5  The average silhouette score is  0.4240344250329259
For cluster = 6  The average silhouette score is  0.3526583283998142
```

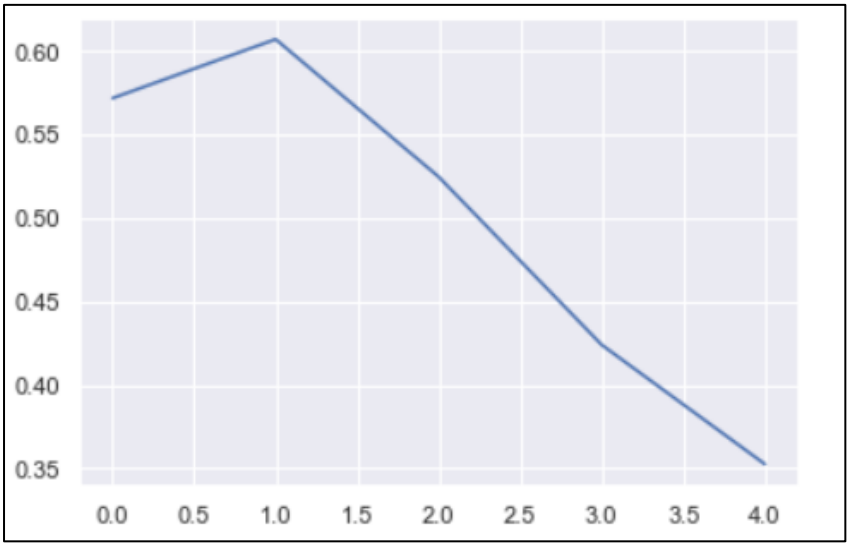


Figure 1.4.2: K-Means silhouette score method

As seen can be seen from figures above, the silhouette score for 3 clusters is the highest. Hence by wss method and silhouette score method 3 clusters will be formed using K-Means algorithm.

```
The silhouette score is  0.40072705527512986
```

```
The silhouette width is 0.0027130893476783 76
```

Figure 1.4.3: Silhouette score and silhouette width

The average silhouettes score is 0.400 and minimum silhouette score is 0.002. The silhouette score ranges from -1 to +1 and higher the silhouette score better the clustering.

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

Solution:

The last step is to plot the clusters that we have created with Hierarchical and K-Means clustering for market segment analysis and devise promotional strategies. We have identified 2 clusters using Hierarchical clustering and 3 using K-Means clustering. Further analysis need to be done to determine the best clustering approach.

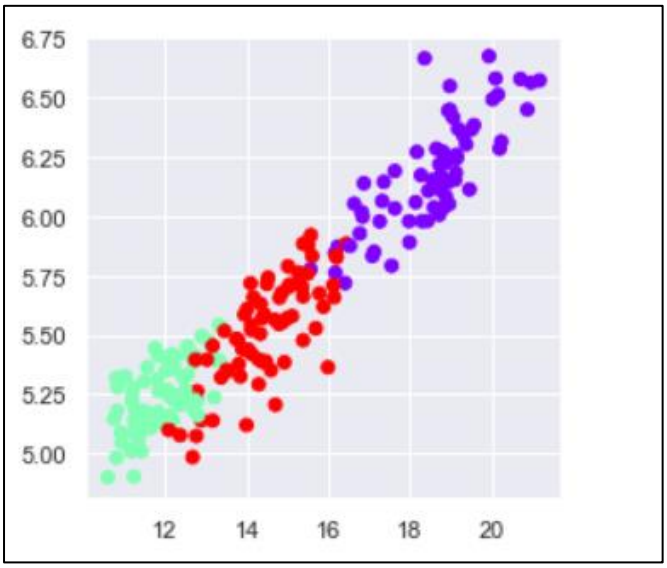


Figure 1.5.1: Hierarchical clustering

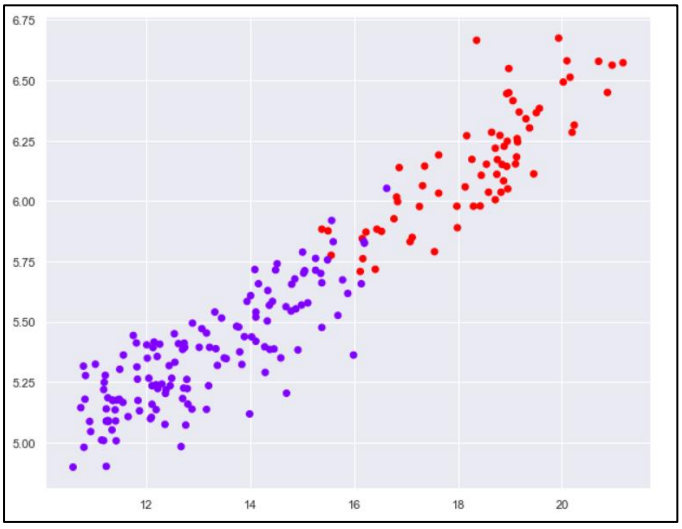


Figure 1.5.2: K-Means Clustering

Variables	Spending	Advance Payments	Probability of full payment	Current Balance	Credit Limit	Min Payment Amt	Max Spent in single shopping
Hierarchical (Cluster 1)	18.62	16.26	0.88	6.19	3.71	3.66	6.06
Hierarchical (Cluster 2)	13.23	13.83	0.87	5.39	3.07	3.71	5.13
K-Means (Cluster 1)	11.86	13.25	0.85	5.23	2.85	4.74	5.1
K-Means (Cluster 2)	18.5	16.2	0.88	6.18	3.7	3.63	6.04
K-Means (Cluster 2)	14.43	14.33	0.88	5.51	3.26	2.7	5.12

Table 1.5.1: Means of features as per clusters

Cluster Inference:

Hierarchical Cluster 1: This segment has higher spending per month, high current balance and credit limit. This is the upper class with majorly higher income. This segment can be targeted using various offers such as cards with rewards or loyalty points for every spent.

Hierarchical Cluster 2: This segment has lower spending per month with low current balance and lower credit limit. This segment can be targeted with cards that have lower interest rates so as to encourage more spending.

K-means Cluster 0: This segment has the lowest spending per month, lowest current balance and credit limit. This segment can be targeted with cards with offers such as zero annual charges and lurking them with benefits such as free coupons or tickets and waivers on a variety of places.

K-means Cluster 1: This segment has higher spending per month, high current balance and credit limit. This segment can be targeted using various offers such as cards with rewards and loyalty points for every spent.

K-means Cluster 2: This segment has must lower spending per month with low current balance and lower credit limit. This segment can be targeted with cards that have lower interest rates so as to encourage more spending.

Problem 2:

Problem Statement:

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

Attribute Information:

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

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

Solution:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Age            3000 non-null   int64
 1   Agency_Code    3000 non-null   object
 2   Type           3000 non-null   object
 3   Claimed        3000 non-null   object
 4   Commision      3000 non-null   float64
 5   Channel        3000 non-null   object
 6   Duration       3000 non-null   int64
 7   Sales          3000 non-null   float64
 8   Product Name   3000 non-null   object
 9   Destination    3000 non-null   object
dtypes: float64(2), int64(2), object(6)
memory usage: 234.5+ KB
```

Figure 2.1.1: Information of insurance dataset

	Age	Commision	Duration	Sales
count	3000.000000	3000.000000	3000.000000	3000.000000
mean	38.091000	14.529203	70.001333	60.249913
std	10.463518	25.481455	134.053313	70.733954
min	8.000000	0.000000	-1.000000	0.000000
25%	32.000000	0.000000	11.000000	20.000000
50%	36.000000	4.630000	26.500000	33.000000
75%	42.000000	17.235000	63.000000	69.000000
max	84.000000	210.210000	4580.000000	539.000000

Figure 2.1.2: Summary of insurance dataset

```
(3000, 10)
```

Figure 2.1.3: Dimension of dataset

Null Value Check:

The null or missing values needs to be checked, if null values are not treated can lead to error or misinterpretation of data. The following is the output of null value check.

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

Figure 2.1.3: Null or missing value check

Dropping non-important columns:

The 'Agency code' column won't be used for analysis. Hence dropping the column. Dataset head after dropping 'Agency code'.

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

Figure 2.1.4: Dataset after dropping column

Univariate Analysis:

Histogram is plotted for continuous features in dataset. Bar plot for categorical features. These plots help in analysing the variables independently.

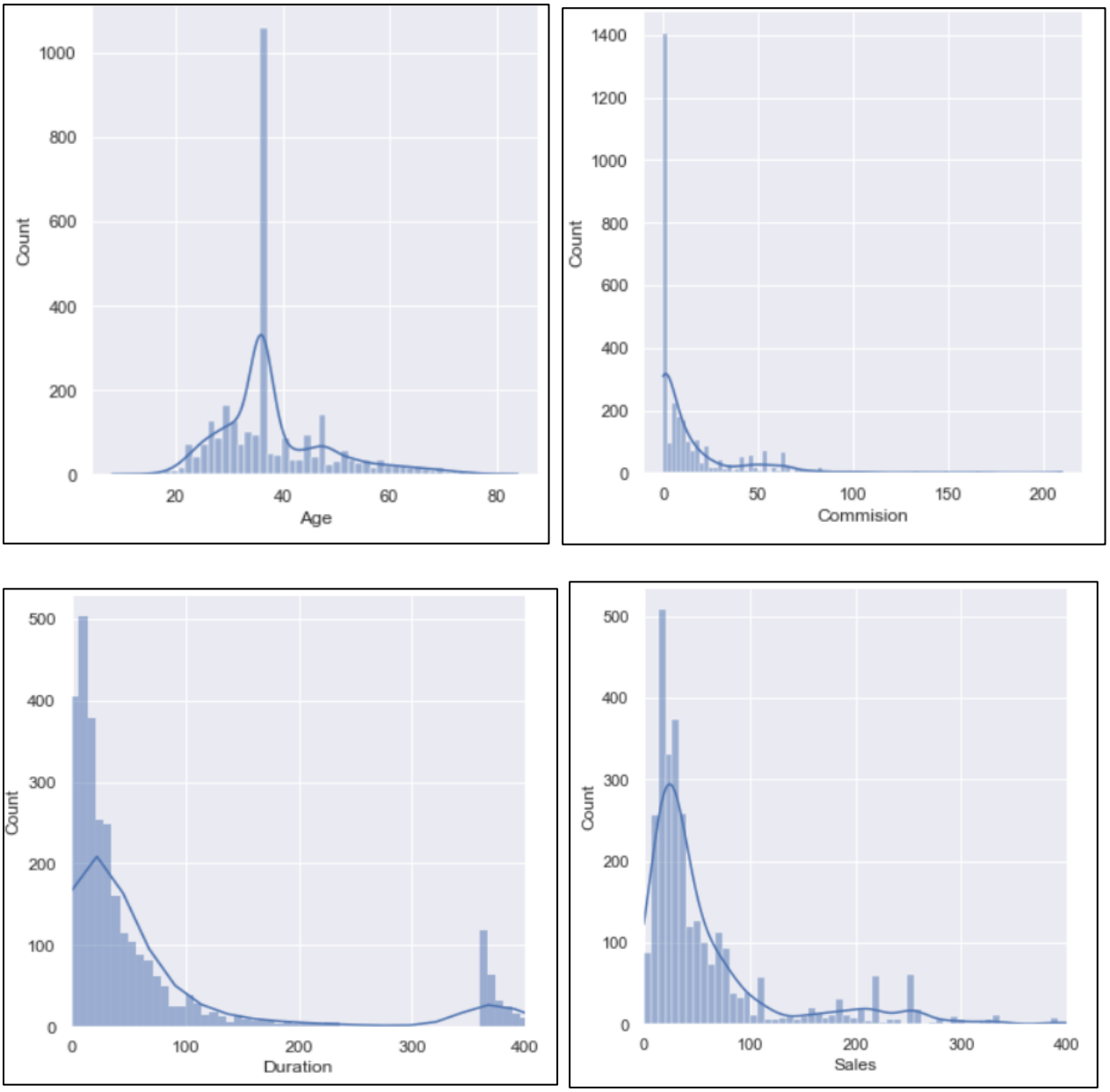
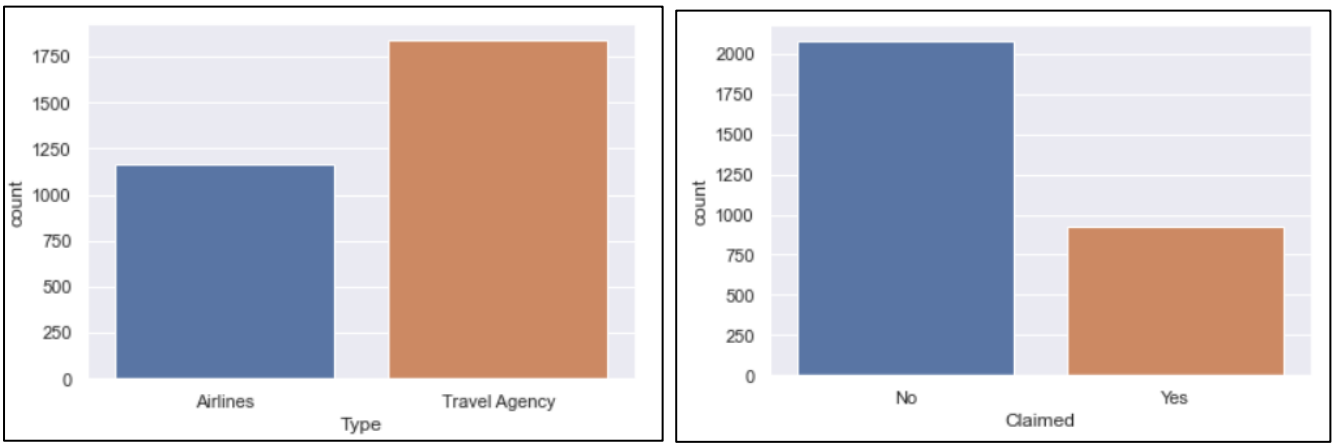


Figure 2.1.5: Histogram plot of dataset

Countplot for Categorical Variables:



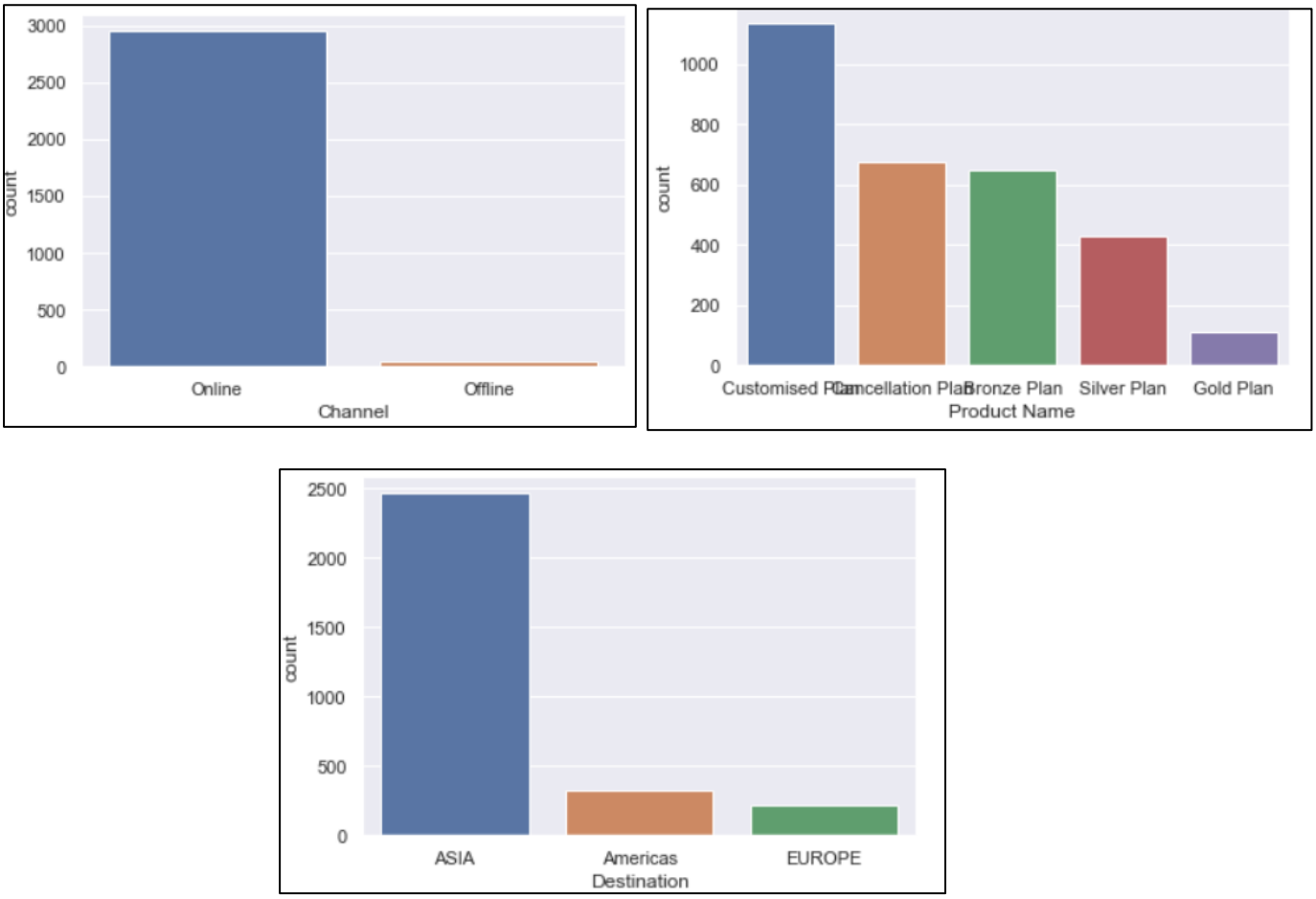


Figure 2.1.6: Countplot of dataset

Boxplot for Outliers:

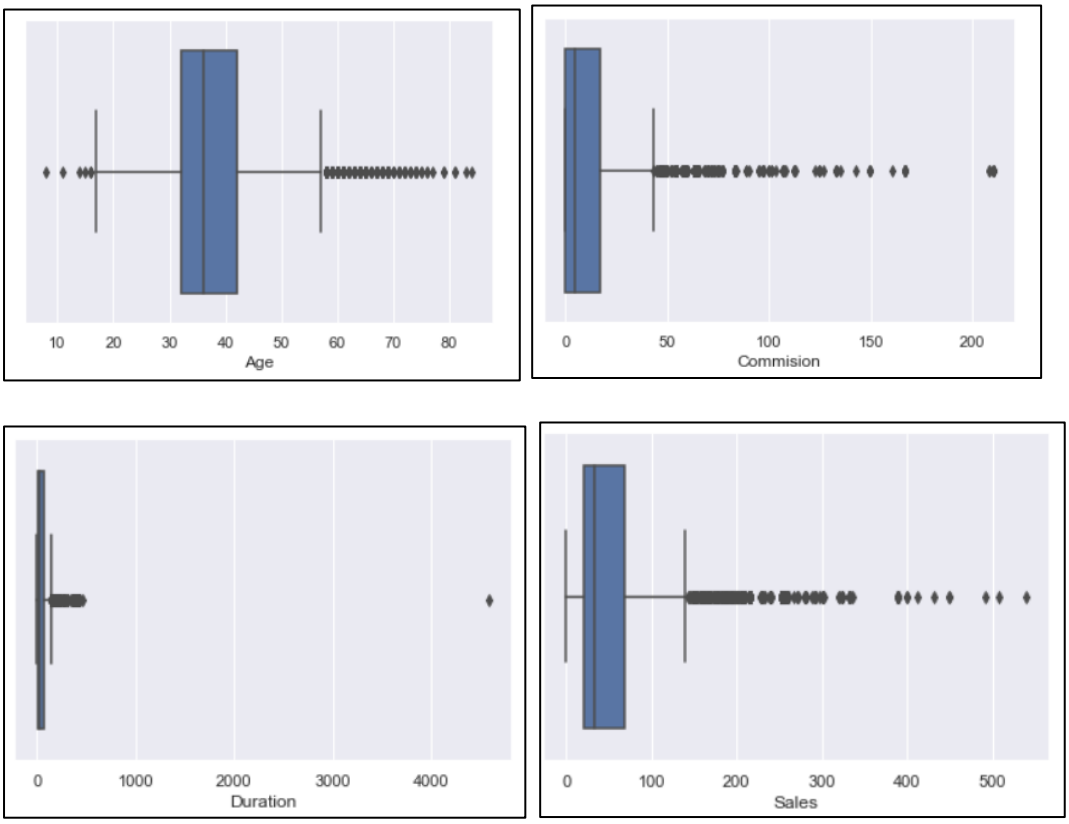


Figure 2.1.7: Boxplot for outliers of continuous variables

The outliers exist for all continuous variables and needs to be removed before further analysis. The outliers in data can cause misinterpretation of data hence needs to be treated. Using the formula for Inter-Quartile Range outliers are removed from features of age, commission, duration, sales.

The dimension of dataset after treating the outliers.

```
(2278, 9)
```

Figure 2.1.8: Dimension of dataset after treating the outliers

Univariate Inference:

From univariate analysis, it can be seen that majority of customers are in age of 0-40, for type being 'Travel Agency' and country ASIA.

<u>Multivariate Analysis:</u>

Heatmap or correlation matrix is plotted to get the relation between different continuous variables. As it can be seen in below correlation map, there is no or low correlation between variables.

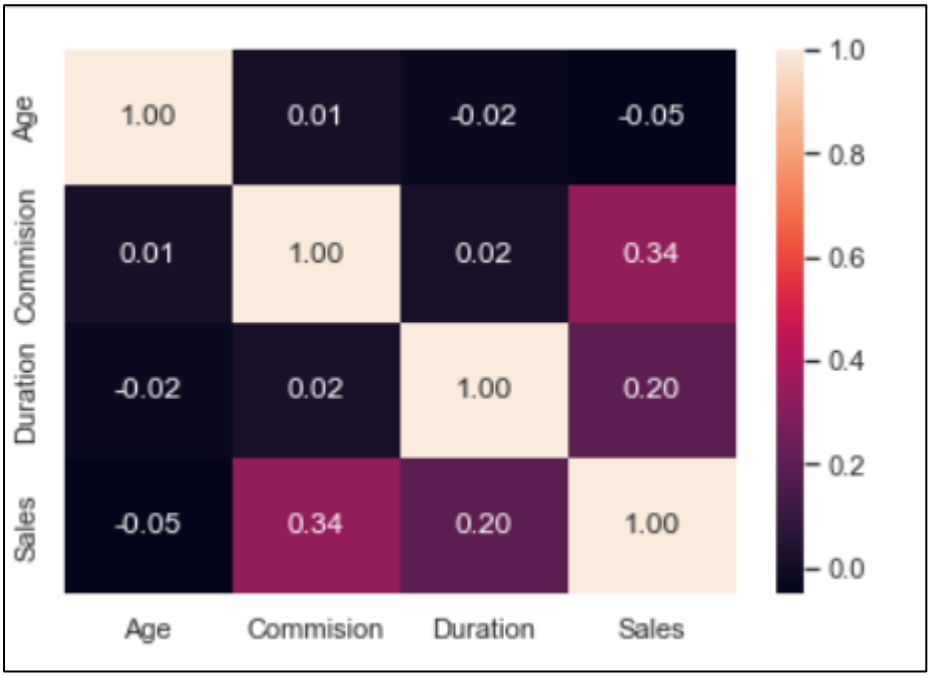


Figure 2.1.9: Heatmap for multivariate analysis of insurance claim dataset

Pair plot of all variables:

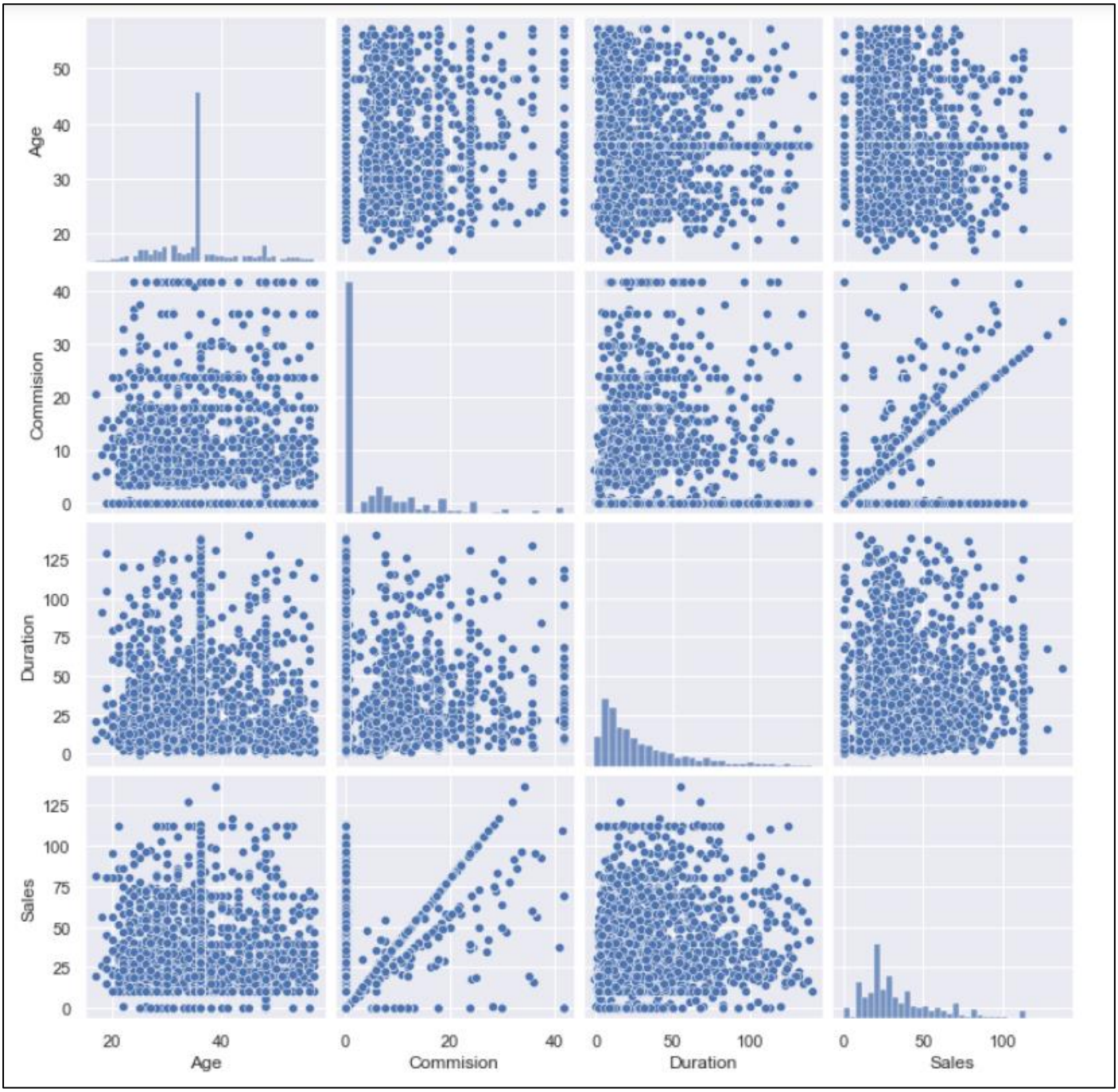


Figure 2.1.10: Pair plot of all variables

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

Solution:

Before splitting the dataset into train and test dataset, object datatypes need to be converted into int datatype. There are five categorical variables type, claim, channel, product name, destination.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2278 entries, 0 to 2999
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Age            2278 non-null   int64
 1   Type           2278 non-null   int8
 2   Claimed        2278 non-null   int8
 3   Commision      2278 non-null   float64
 4   Channel        2278 non-null   int8
 5   Duration       2278 non-null   int64
 6   Sales          2278 non-null   float64
 7   Product Name   2278 non-null   int8
 8   Destination    2278 non-null   int8
dtypes: float64(2), int64(2), int8(5)
memory usage: 164.6 KB
```

Figure 2.2.1: Dataset after converting object datatype into int

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

Figure 2.2.2: Head of dataset after datatype conversion

Now to split the data into training and testing dataset, the 'Claimed' column needs to be removed and popped from the dataset as that is the target column.
The dataset is split into ratio of 70:30, the training dataset is stored in X_train and testing dataset in X_test. The dimension of training and testing dataset is as below.

```
(1594, 8)
```

Figure 2.2.3: Training set dimension

```
(684, 8)
```

Figure 2.2.4: Testing set dimension

CART MODEL

Classification or Regression Tree (CART) is a type of supervised learning technique is either a discrete or class (classification) of the dataset or the outcome is of continuous or numerical in nature(regression).

Using the training set, the model is trained and testing dataset is used to test the model.

DecisionTressClassifier and Tree packages from sklearn are used for creating the CART model. Decision tree is created using DecisionTreeClassifier and using the 'gini' criterion. The 'tree' package is used to create the dot file to help visualize the tree.
Below are the feature importance values to build the tree

	Imp
Age	0.168935
Type	0.095787
Commision	0.120000
Channel	0.001813
Duration	0.275545
Sales	0.270128
Product Name	0.035941
Destination	0.031850

Figure 2.2.5: Feature importance values

Using the GridSearchCV package identified the best parameters to build regularised decision tree.

```
GridSearchCV(cv=3, estimator=DecisionTreeClassifier(),
             param_grid={'max_depth': [8, 9, 10],
                         'min_samples_leaf': [15, 20, 25],
                         'min_samples_split': [45, 60, 75]})
```

Figure 2.2.6: GridSearchCV output

```
{'max_depth': 8, 'min_samples_leaf': 25, 'min_samples_split': 45}
```

Figure 2.2.7: Best params from Grid search

```
DecisionTreeClassifier(max_depth=8, min_samples_leaf=25, min_samples_split=45)
```

Figure 2.2.8: Regularised decision tree params

The regularised decision tree is build using the best params from grid search and the tree is stored in dot file for analysing the data.

<u>Random Forest Model:</u>

Random Forest is another supervised learning algorithm which consists of multiple decision trees and predictions are made for individual trees and one best output is selected.

RandomForestClassifier package is used from sklearn ensemble and a random forest is build using the training dataset and further testing dataset is used on model.

Using GridSearchCV the best params are obtained for building the random forest classifier

```
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [6, 7], 'max_features': [4, 5],
                         'min_samples_leaf': [25, 30],
                         'min_samples_split': [20, 30],
                         'n_estimators': [101, 301]})
```

Figure 2.2.9: Best params from grid search CV for Random Forest

```
RandomForestClassifier(max_depth=6, max_features=5, min_samples_leaf=25,
                       min_samples_split=30, n_estimators=301)
```

Figure 2.2.10: Params for Random Forest classifier

Using these params a Random Forest Classifier is built which is used for further performance evaluation.

Artificial Neural Network (ANN):

ANN is computational model that consists several processing elements that receive inputs and gives output based on predefined functions.

MLPClassifier package is from sklearn neural network which is used to built the ANN model. Using the GridSearchCV first the best params are identified and MLP is created using those params with training data.

```
GridSearchCV(cv=3, estimator=MLPClassifier(),
             param_grid={'activation': ['logistic', 'relu'],
                         'hidden_layer_sizes': [(500, 500, 500)],
                         'max_iter': [10000], 'solver': ['sgd', 'adam'],
                         'tol': [1, 0.1, 0.01]})
```

Figure 2.2.11: Best params from Grid search CV for MLP

```
{'activation': 'relu',
 'hidden_layer_sizes': (500, 500, 500),
 'max_iter': 10000,
 'solver': 'adam',
 'tol': 0.01}
```

Figure 2.2.12: Best params from Grid

```
MLPClassifier(hidden_layer_sizes=(500, 500, 500), max_iter=10000, tol=0.01)
```

Figure 2.2.13: MLP built using best params

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

Solution:

For the models CART, Random Forest and ANN, few of the model evaluators are classification report, confusion matrix, AUC score and ROC plot. The evaluators are calculated using packages of classification report, confusion matrix, auc_roc_score from sklearn metrics. The evaluators are calculated first for training dataset and then testing dataset.

CART Model:

	precision	recall	f1-score	support
0	0.85	0.92	0.88	1204
1	0.67	0.51	0.58	390
accuracy			0.82	1594
macro avg	0.76	0.71	0.73	1594
weighted avg	0.81	0.82	0.81	1594

Figure 2.3.1: Classification report of training dataset

	precision	recall	f1-score	support
0	0.80	0.90	0.85	497
1	0.61	0.41	0.49	187
accuracy			0.77	684
macro avg	0.70	0.66	0.67	684
weighted avg	0.75	0.77	0.75	684

Figure 2.3.2: Classification report for testing dataset

```
array([[1107,   97],
       [ 192,  198]], dtype=int64)
```

Figure 2.3.3: Confusion matrix for training dataset

```
array([[447,  50],
       [110,  77]], dtype=int64)
```

Figure 2.3.4: Confusion matrix for testing dataset

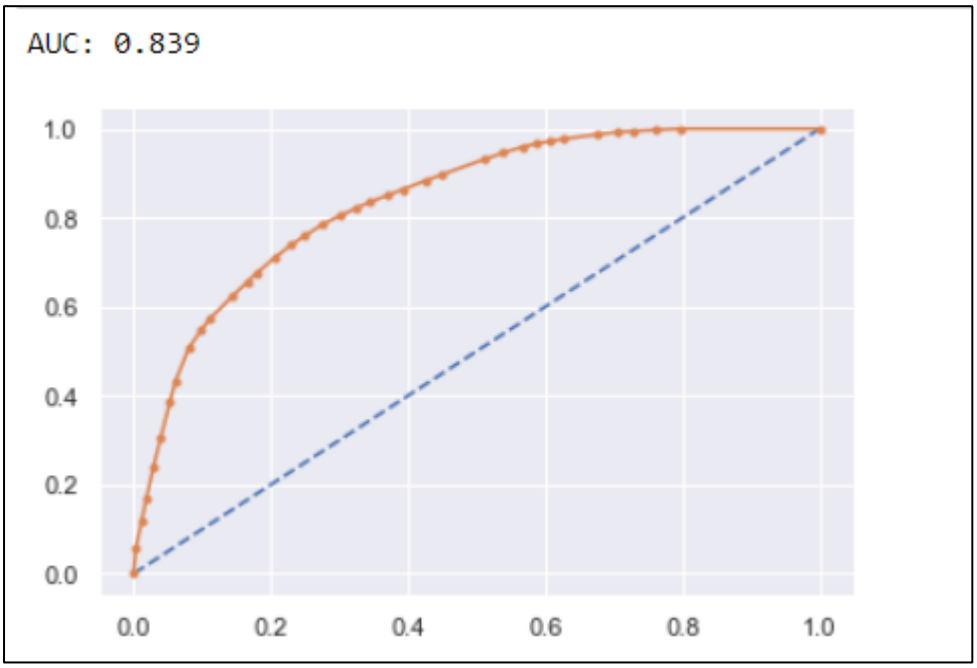


Figure 2.3.5: AUC score and ROC curve for training data

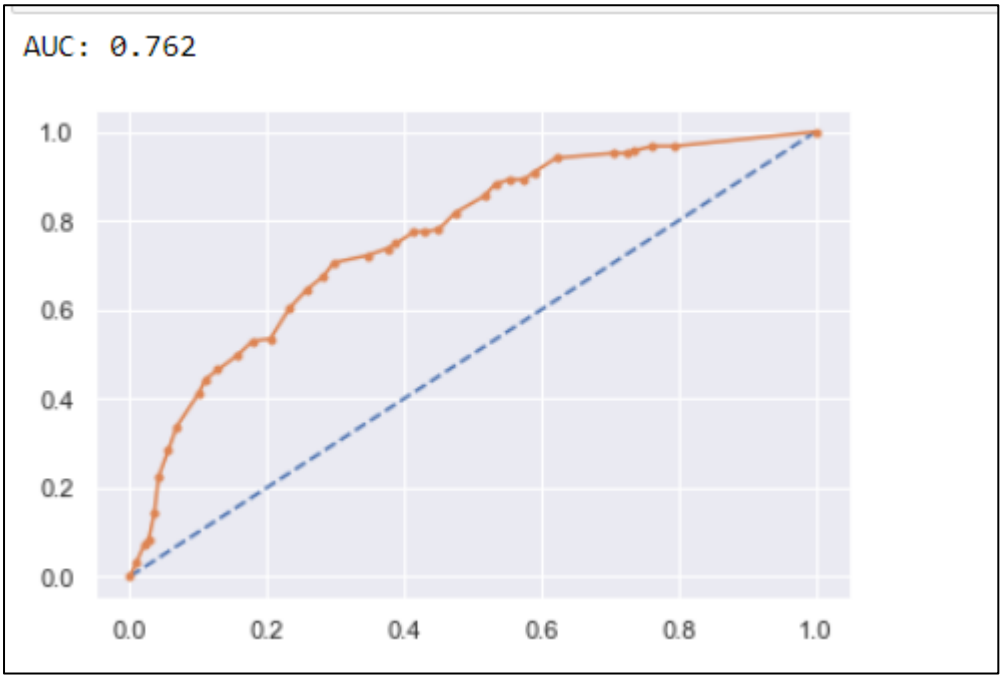


Figure 2.3.6: AUC score and ROC curve for testing dataset

Model score for training dataset:

0.8186951066499373

Model score for testing dataset:

0.7660818713450293

Random Forest Model:

	precision	recall	f1-score	support
0	0.84	0.94	0.89	1204
1	0.71	0.43	0.53	390
accuracy			0.82	1594
macro avg	0.77	0.68	0.71	1594
weighted avg	0.80	0.82	0.80	1594

Figure 2.3.7: Classification report for training dataset

	precision	recall	f1-score	support
0	0.79	0.93	0.85	497
1	0.64	0.33	0.44	187
accuracy			0.77	684
macro avg	0.71	0.63	0.64	684
weighted avg	0.75	0.77	0.74	684

Figure 2.3.8: Classification report for testing dataset

```
array([[1135,   69],
       [ 224,  166]], dtype=int64)
```

Figure 2.3.9: Confusion matrix for training dataset

```
array([[462,  35],
       [125,  62]], dtype=int64)
```

Figure 2.3.10: Confusion Matrix for testing dataset

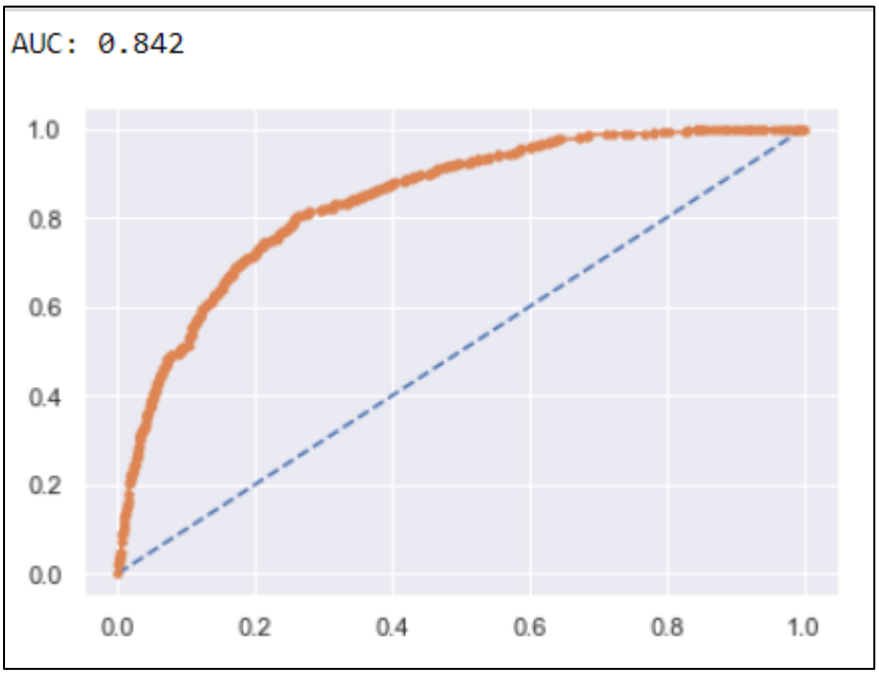


Figure 2.3.11: AUC score and ROC curve for training dataset

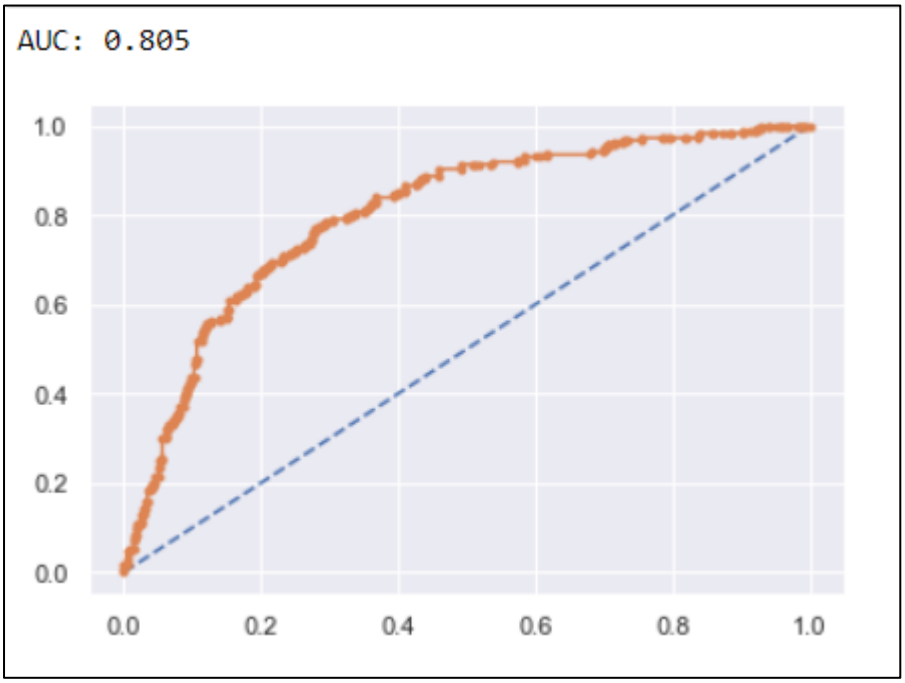


Figure: 2.3.12: AUC score and ROC curve for testing dataset

Artificial Neural Network Model (ANN):

	precision	recall	f1-score	support
0	0.82	0.95	0.88	1204
1	0.68	0.34	0.46	390
accuracy			0.80	1594
macro avg	0.75	0.65	0.67	1594
weighted avg	0.78	0.80	0.77	1594

Figure 2.3.13: Classification report for training dataset

	precision	recall	f1-score	support
0	0.79	0.92	0.85	497
1	0.62	0.33	0.43	187
accuracy			0.76	684
macro avg	0.70	0.63	0.64	684
weighted avg	0.74	0.76	0.74	684

Figure 2.3.14: Classification report for testing dataset

```
array([[1141,   63],
       [ 256,  134]], dtype=int64)
```

Figure 2.3.15: Confusion matrix for training dataset

```
array([[459,  38],
       [125,  62]], dtype=int64)
```

Figure 2.3.16: Confusion matrix for testing dataset

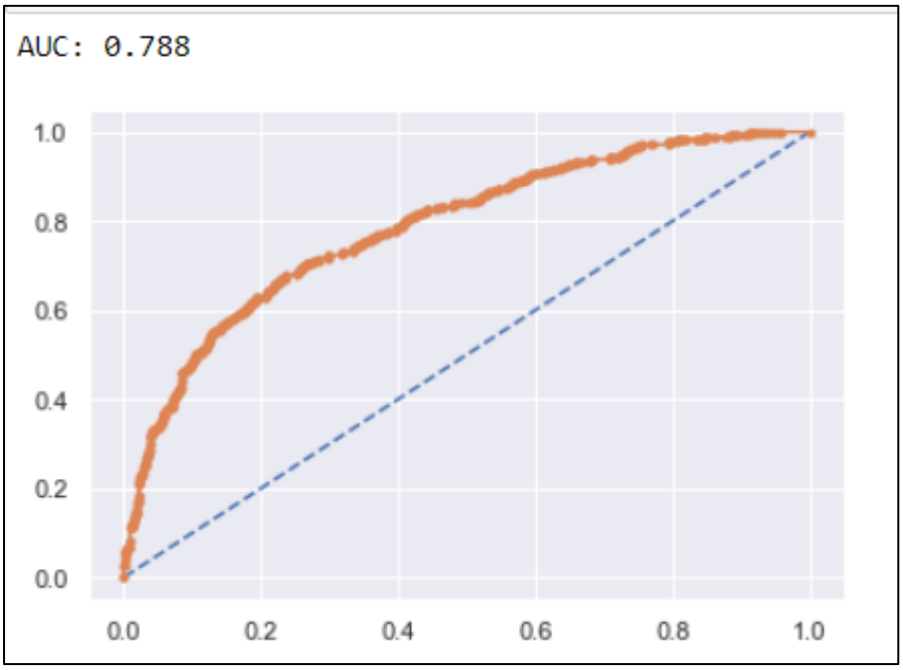


Figure 2.3.17: AUC score and ROC curve for training dataset

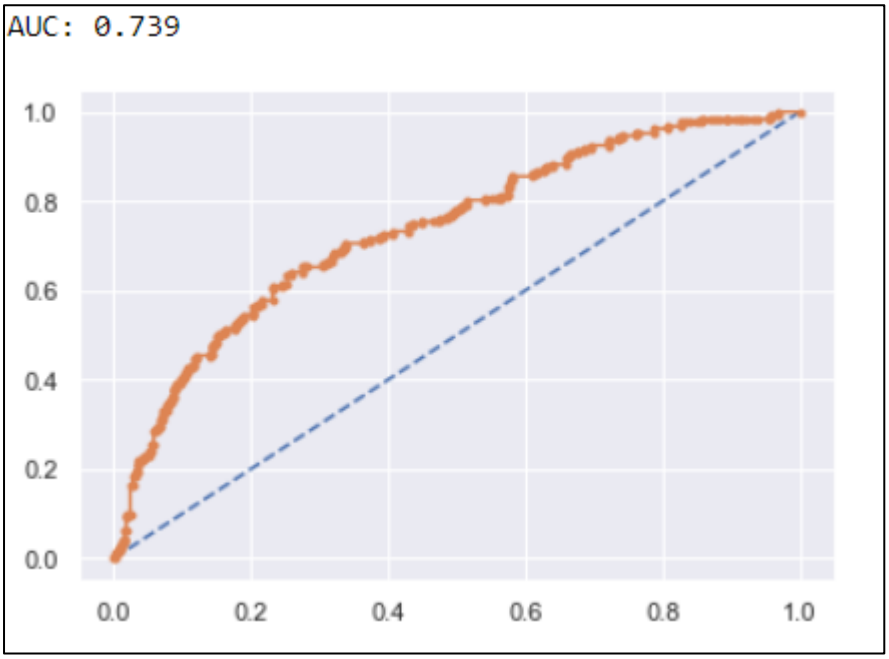


Figure 2.3.18: AUC score and ROC curve for testing dataset

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

Solution:

The performance evaluators of all three models CART, Random Forest and ANN is in following table:

Model	Precision	F1 Score	AUC Score
CART Model			
Train Data	0.67	0.82	0.84
Test Data	0.61	0.77	0.76
Random Forest			
Train Data	0.71	0.82	0.84
Test Data	0.64	0.77	0.8
Neural Network			
Train Data	0.68	0.8	0.78
Test Data	0.6	0.76	0.74

Table 2.4.1: Performance Evaluators of different models

Inference:
As it can be seen from above comparison table, the precision for training data is highest for Random Forest model, and so for testing dataset. The AUC score for CART and RF model is same for training dataset but it's highest for testing dataset. The Random Forest is best option as this will exhibit less variance as compared to CART and ANN model.

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

Solution:

For this business problem, we had to build a model which predicts the claim status for the insurance company. The different models built and analysed for this problem are CART model, Random Forest and ANN model. These models are built on training dataset and evaluated on testing dataset.

The different parameters are evaluated for each of the models, classification reports, confusion matrix, AUC score and ROC curve, which are further compared to get the best suited model for predicting the claim status. All models performed well but to predict the claim status for the insurance company Random Forest performed the best. The Random Forest model has highest AUC score for testing dataset.

Random Forest model has less variance and is best suited for this business model.