

# CLUSTERING & MODELING DATA ANALYSIS

Mohamed Rifaz Ali K S  
PGP-DSBA Online  
October' 21

## Contents

Problem 1: Clustering.....	8
Executive Summary.....	8
Introduction .....	8
Data Description .....	8
Sample of the Salary Data dataset.....	9
Let us check the types of variables in the data frame .....	9
Check for missing values in the dataset.....	9
Check for the duplicate records in the dataset.....	10
Summary of the Dataset .....	10
Observations .....	10
Boxplot to identify the Outliers .....	11
Histograms and Boxplots .....	11
Pair plot and Correlation Plot: .....	16
Inferences from correlation plots: .....	17
Summary of the Dataset before Scaling .....	18
Summary of the Dataset after Scaling .....	18
Scaled Data.....	19
Ward's Method .....	20
<i>Method 1 – maxclust criterion (ward's method)</i> .....	21
<i>Method 2 – distance criterion (ward's method)</i> .....	22
<i>Cluster profiling for Ward's Method</i> .....	23
Average Method .....	24
<i>Method 1 – maxclust criterion (Average Method)</i> .....	25
<i>Method 2 – distance criterion (Average Method)</i> .....	25
<i>Cluster profiling for Average Method</i> .....	26
Customer Segmentation for Hierarchical Clustering .....	27
Observations .....	27

K Means Technique.....	28
<i>Elbow curve</i> .....	29
Customer Segmentation for K Means Clustering.....	31
Cluster profiling for K Means Technique .....	32
Problem 2: CART-RF-ANN .....	34
Executive Summary.....	34
Introduction .....	34
Sample of the Salary Data dataset.....	35
Data Description .....	35
Let us check the types of variables in the data frame. ....	35
Summary of the Dataset .....	36
Observations .....	36
Check for duplicate values in the dataset.....	37
Outliers Proportion in numerical columns.....	37
Check the null values .....	38
Histograms and Boxplots .....	38
Correlation Plots .....	41
Pair plot.....	41
Heat Map .....	41
Observations .....	42
Categorical Variables .....	42
Value counts of each column to see if anomalies are present in the dataset.....	48
Converting object data types to categorical codes.....	49
Target variable counts and percentage for Train & Test .....	51
Feature Importance Check.....	51
Grid Search to find out the optimal hyper parameters for CART training set.....	52
Regularizing the tree using the best grid parameters .....	53
Feature Importance's.....	53

Predicting on Training and Test dataset .....	53
Building the Random Classifier Model.....	53
Boxplot to verify after treating the outliers.....	54
Correlation Map .....	54
Grid Search to find out the optimal hyper parameters for RF training set .....	55
Feature Importance's.....	56
Predicting on Training and Test dataset .....	56
Building the Artificial Neural Network Model.....	57
Grid Search to find out the optimal hyper parameters for ANN training set .....	57
Predicting on Training and Test dataset .....	58
Decision Tree Predictions.....	59
Random Classifier Model Predictions .....	61
ANN Predictions .....	63
ROC Curve for DT, RF and ANN .....	66
Insights & Recommendations .....	67

## List of Figures

Figure 1. Histogram for all numerical variables to find the distribution .....	11
Figure 2. Source: <a href="https://www.simplypsychology.org/boxplots.html">https://www.simplypsychology.org/boxplots.html</a> .....	11
Figure 3. Histogram & Boxplot for Spending .....	12
Figure 4. Histogram & Boxplot for Advance Payments.....	12
Figure 5. Histogram & Boxplot for Probability of full payment .....	13
Figure 6. Histogram & Boxplot for current balance .....	13
Figure 7. Histogram & Boxplot for credit limit.....	14
Figure 8. Histogram & Boxplot for payment amount .....	14
Figure 9. Histogram & Boxplot for max_spent_in_single_shopping .....	15
Figure 10. Pair plot for all the numerical variables.....	16
Figure 11. Heat Map for all the numerical variables .....	17

Figure 12. Source: Towards Data Science .....	19
Figure 13.Dendrogram for ward's method .....	20
Figure 14.Truncated Dendrogram for ward's method (maxclust criterion) .....	21
Figure 15.Cluster mapping using 'maxclust' criterion .....	21
Figure 16.Truncated Dendrogram for ward's method (distance criterion) .....	22
Figure 17.Cluster mapping using 'distance' criterion .....	22
Figure 18.Cluster counts .....	23
Figure 19. Dendrogram for average method .....	24
Figure 19.Truncated Dendrogram for average method (maxclust criterion) .....	24
Figure 20.Cluster mapping using 'maxclust' criterion .....	25
Figure 21.Truncated Dendrogram for average method (distance criterion) .....	25
Figure 22.Cluster mapping using 'distance' criterion .....	26
Figure 23.Cluster segmentation for Hierarchical clustering (ward's method) .....	27
Figure 24. Cluster mapping using K Means algorithm .....	28
Figure 25. Inertia values from K=1 to K-10 .....	29
Figure 26. Elbow curve from K=1 to K-10 .....	29
Figure 27.Cluster counts for K Means .....	30
Figure 28.Cluster Segmentation for K Means .....	31
Figure 27.Data Types.....	35
Figure 30. Histogram for all numerical variables to find the distribution .....	37
Figure 31.Null Values check .....	38
Figure 32.Histogram and Box plot for Age .....	38
Figure 33.Histogram and Box plot for Commission .....	39
Figure 34.Histogram and Box plot for Duration.....	40
Figure 35.Histogram and Box plot for Sales.....	40
Figure 36.Pair plot to see the correlation between the variables .....	41
Figure 37.Heat Map to see the correlation between the variables.....	41
Figure 38.Count plot for Agency Code .....	42

Figure 39.Count plot for Agency Code Vs Sales .....	43
Figure 40.Count plot for Type .....	43
Figure 41.Count plot for Type Vs Sales .....	44
Figure 42.Count plot for Channel.....	44
Figure 43.Count plot for Channel Vs Sales.....	45
Figure 44.Count plot for Product Name .....	45
Figure 45.Count plot for Product Name Vs Sales.....	46
Figure 46.Count plot for Destination .....	46
Figure 47.Count plot for Destination Vs Sales .....	47
Figure 48.Count plot for Claimed.....	47
Figure 48.Verify data types after conversion.....	50
Figure 49.Feature Importance Check.....	51
Figure 50.Sales leaf node .....	52
Figure 51.Feature importance for CART .....	53
Figure 52.Box plots after treating the outliers.....	54
Figure 53.Heat Map after treating the outliers .....	54
Figure 53.Feature importance's for RF Model .....	56
Figure 54.ROC Curve for all the Models.....	66

## List of Tables

Table 1. Bank Dataset Sample.....	9
Table 2.Bank Data Types.....	9
Table 3. Summary of the Bank Dataset.....	10
Table 4. Summary of the Bank Dataset before scaling .....	18
Table 5. Summary of the Bank Dataset after scaling .....	18
Table 6. Scaled data after applying z-score .....	19
Table 9. Cluster mapping for Average Method to the dataset .....	26
Table 10. Cluster profiling for Average Method .....	26

Table 11. Cluster mapping for K Means .....	30
Table 12. Cluster profiling for K Means.....	32
Table 13. Insurance Dataset Sample.....	35
Table 14. Insurance Dataset Summary .....	36
Table 15. Anomalies in Duration variable.....	37
Table 16. Sample Dataset after encoding .....	50
Table 17. Sample Dataset after encoding .....	57
Table 18. Comparison of CART, RF and ANN Models .....	66

## Problem 1: Clustering

### Executive Summary

A leading bank wants to develop a customer segmentation to give promotional offers to its customers. They collected a sample that summarizes the activities of 210 users during the past few months. Based on the credit card usage, we are asked to group the customers so that the bank will provide the customized promotional offers to the groups instead of common offers to everyone.

### Introduction

The purpose of this problem is to segment the users based on the credit card usage. There are users who use the cards high, moderate and low. Using 'clustering' technique, we will group the users based on the credit card usage.

Clustering is a part of unsupervised learning. It is the technique of grouping objects, with heterogeneity between groups and homogeneity within the groups. It can follow Agglomerative, Divisive or Partitioning approach. Distance calculations are done to find similarity and dissimilarity in Clustering problems. We use ward's & average method as part of hierarchical clustering and k-means(partitioning) part of non-hierarchical clustering for our business problem.

**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.**

There are 210 records and 7 columns present in the Bank provided dataset.

### Data Description

1. spending: Amount spent by the customer per month (in 1000s)
2. advance\_payments: Amount paid by the customer in advance by cash (in 100s)
3. probability\_of\_full\_payment: Probability of payment done in full by the customer to the bank
4. current\_balance: Balance amount left in the account to make purchases (in 1000s)
5. credit\_limit: Limit of the amount in credit card (10000s)
6. min\_payment\_amt : minimum paid by the customer while making payments for purchases made monthly (in 100s)
7. max\_spent\_in\_single\_shopping: Maximum amount spent in one purchase (in 1000s)



## Sample of the Salary Data 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. Bank Dataset Sample

Let us check the types of variables in the data frame

spending	float64
probability_of_full_payment	float64
current_balance	float64
credit_limit	float64
min_payment_amt	float64
max_spent_in_single_shopping	float64

Table 2. Bank Data Types

All of the variables are in float format.

Check for missing values in the dataset

```

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

```

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

## Check for the duplicate records in the dataset

Number of duplicate rows = 0

Upon checking, there are no duplicates present in the dataset as well.

## Summary of the 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

Table 3. Summary of the Bank Dataset

Looks, there are no anomalies in the dataset. Also, upon looking at the mean, minimum and maximum values there are very minimal outliers present. Only for the 'min\_payment\_amt' variable may have few outliers. We will be able to see to it in better clarity with boxplot technique for all the continuous variables.

## Observations

- There are 210 rows and 7 columns are present in the dataset.
- All the columns such as *spending*, *advance\_payments*, *probability\_of\_full\_payment*, *current\_balance*, *credit\_limit*, *min\_payment\_amt*, *max\_spent\_in\_single\_shopping* are in float format.
- No duplicates and missing values are present.
- Upon looking at the average and minimum/maximum values, there are very minimal outliers present in the dataset.
- Only for 'min\_payment\_amt' variable may have few outliers.
- There are no anomalies present in the dataset which concludes our pre-processing the data further.

## Boxplot to identify the Outliers

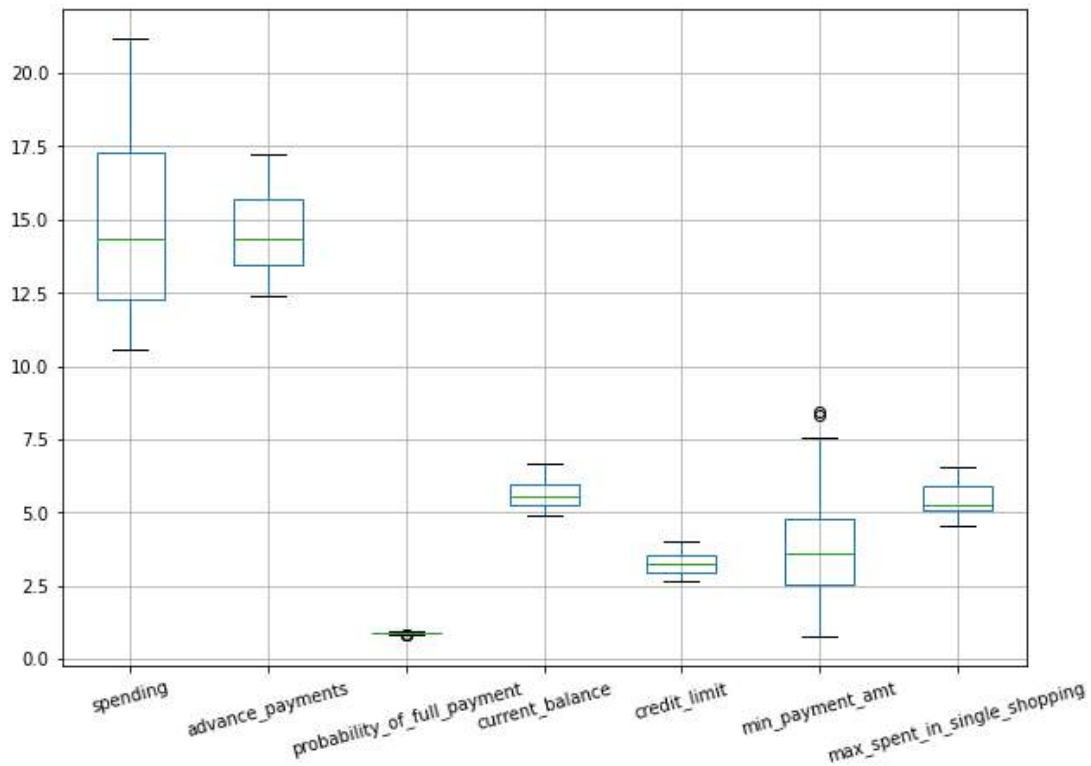


Figure 1. Histogram for all numerical variables to find the distribution

As we assumed, 'min\_payment\_amt' variable has very few outliers.

## Histograms and Boxplots

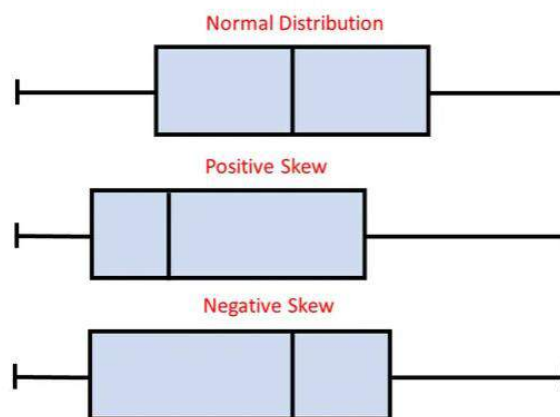
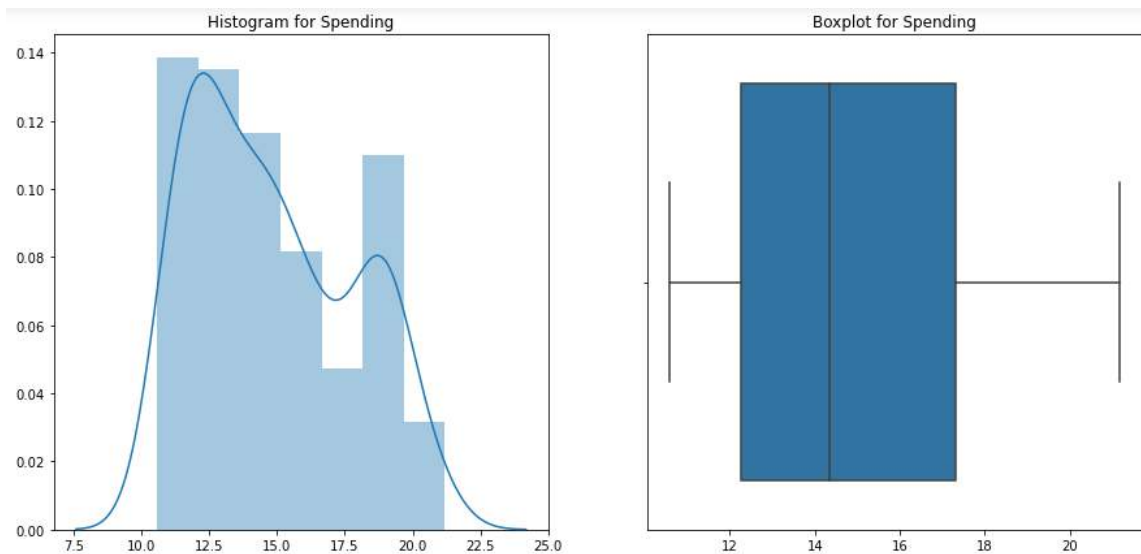


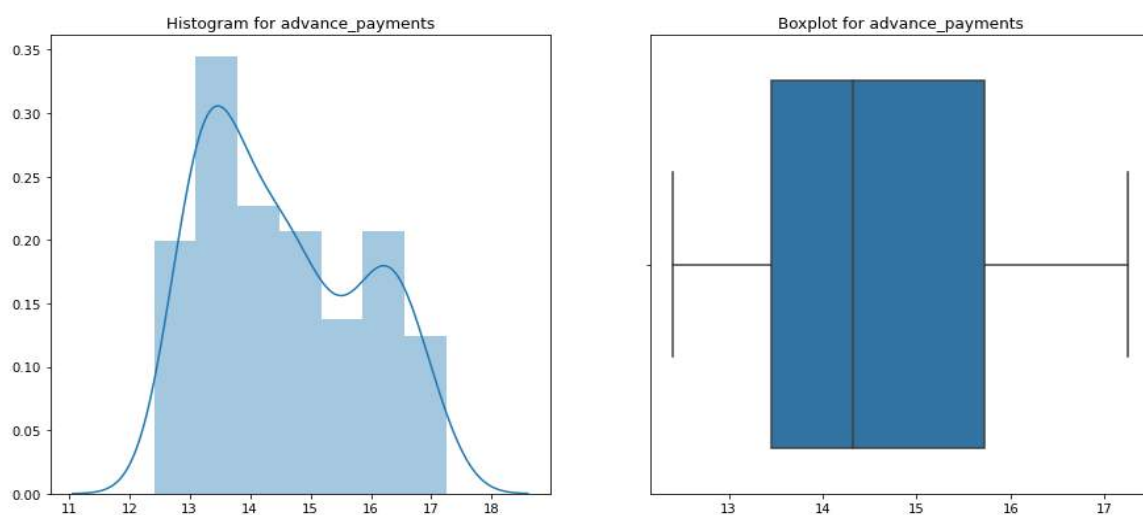
Figure 2. Source: <https://www.simplypsychology.org/boxplots.html>

Below graphs helps us identifying the data distribution for continuous columns.



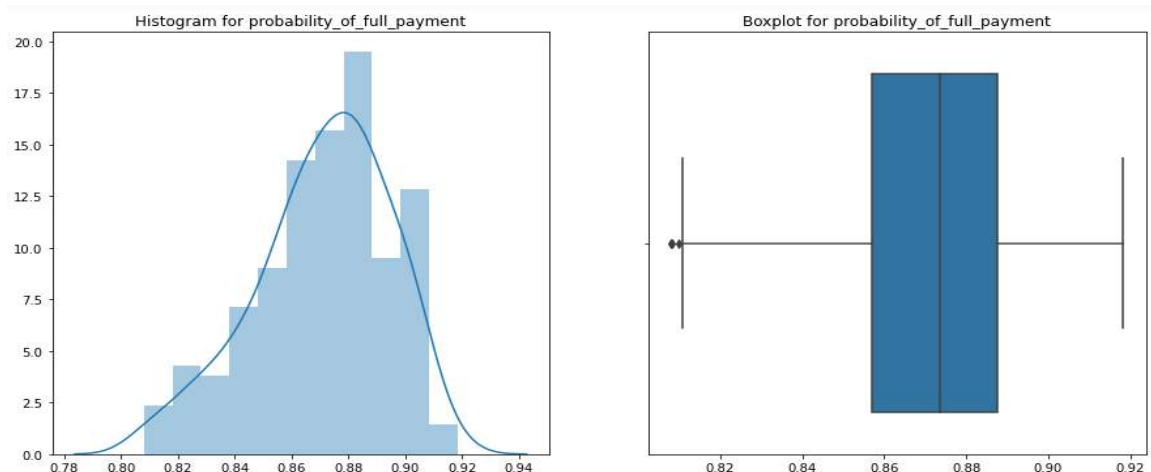
*Figure 3. Histogram & Boxplot for Spending*

- Histogram shows the data's distributed from ~10.5 to 21
- The 'Spending' variable has no outliers.
- 'Spending' is positively skewed with the value of 0.4
- Most of the data's are clustered around the left tail of the distribution while the right tail of the distribution is longer.



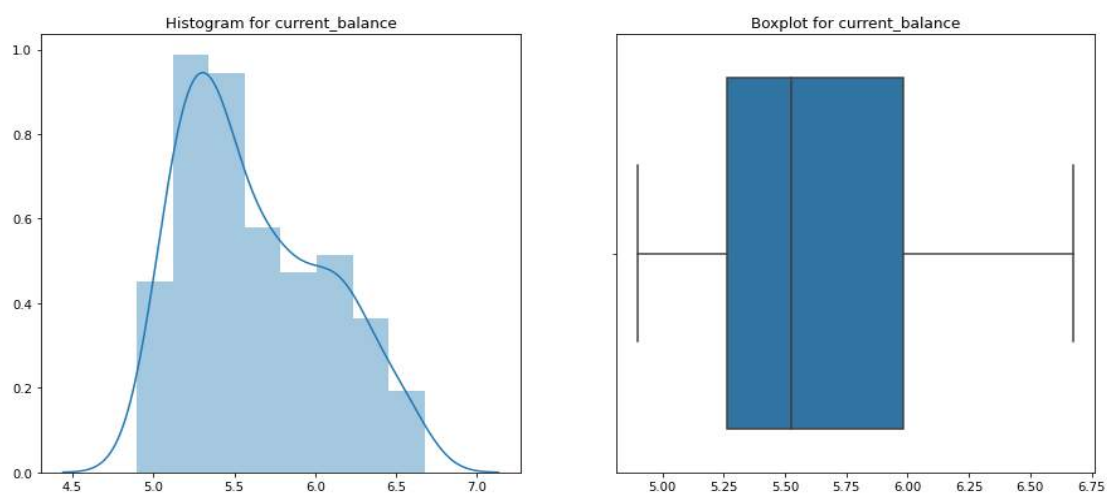
*Figure 4. Histogram & Boxplot for Advance Payments*

- Histogram shows the data's distributed from ~12 to 17.
- The 'advance\_payments' variable has no outliers.
- 'advance\_payments' is positively skewed with the value of 0.39
- Most of the data's are clustered around the left tail of the distribution while the right tail of the distribution is longer.



*Figure 5. Histogram & Boxplot for Probability of full payment*

- Histogram shows the data's distributed from ~0.80 to 0.92
- In 'probability\_of\_full\_payment' variable, only two records are outliers.
- 'probability\_of\_full\_payment' is negatively skewed with the value of -0.54
- Most of the data's are clustered around the right tail of the distribution while the left tail of the distribution is longer.



*Figure 6. Histogram & Boxplot for current balance*

- Histogram shows the data's distributed from ~4.8 to 6.7
- The 'current\_balance' variable has no outliers.
- 'current\_balance' is positively skewed with the value of 0.53
- Most of the data's are clustered around the left tail of the distribution while the right tail of the distribution is longer.

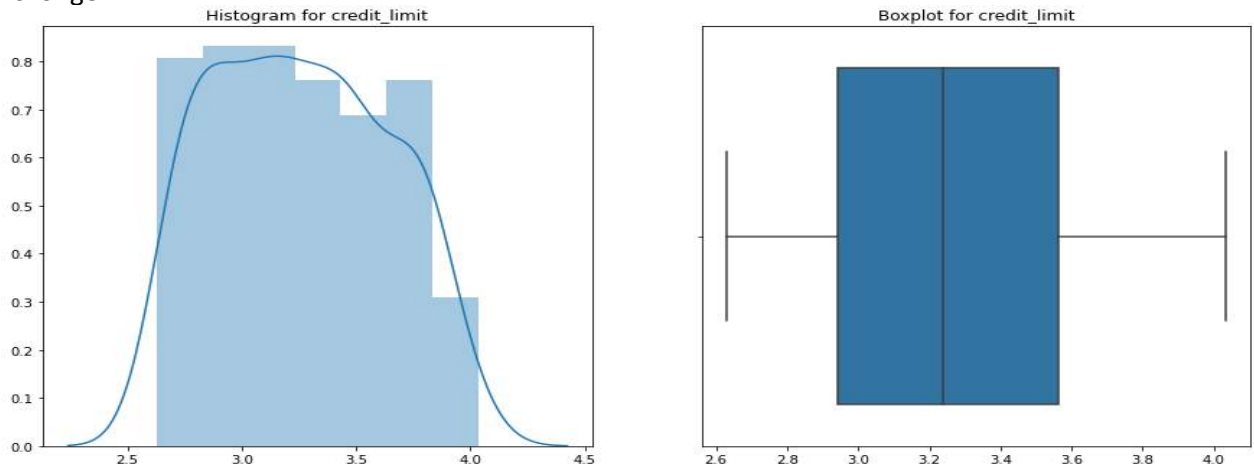


Figure 7. Histogram & Boxplot for credit limit

- Histogram shows the data's distributed from ~2.6 to 4.0
- The 'credit\_limit' variable has no outliers.
- 'credit\_limit' is slightly positively skewed with the value of 0.13
- Distribution looks almost symmetric.

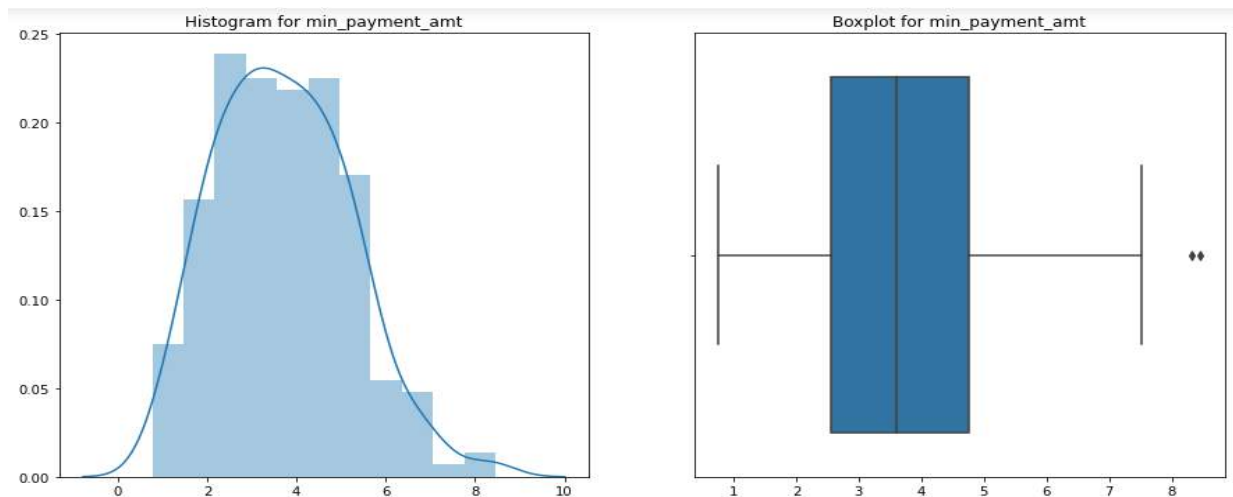
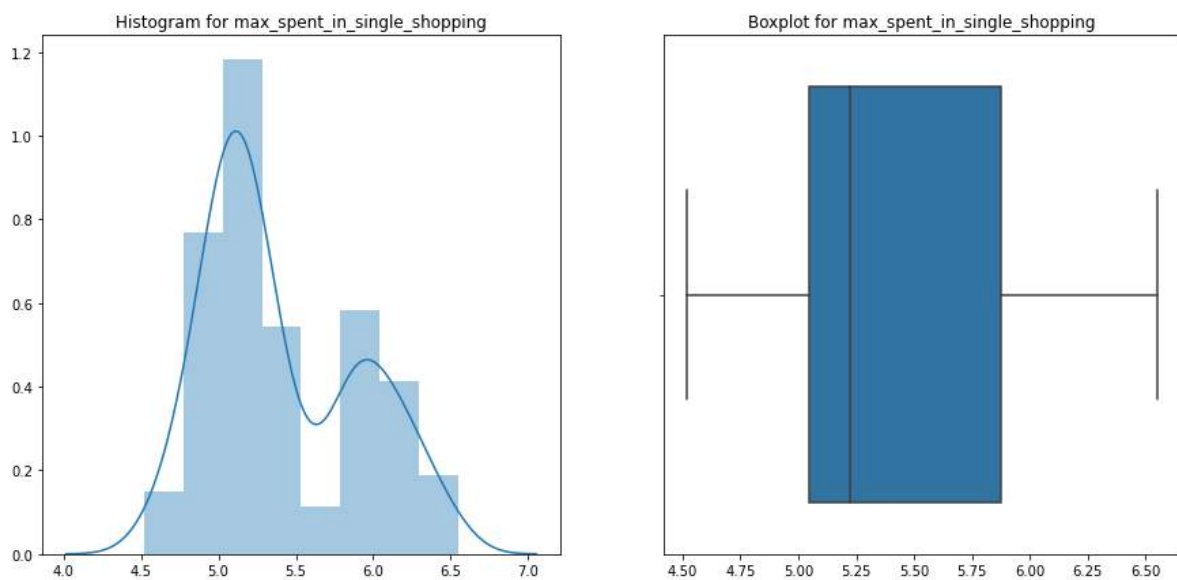


Figure 8. Histogram & Boxplot for payment amount

- Histogram shows the data's distributed from ~0.7 to 8.4
- In 'min\_payment\_amt' variable, two records are outliers.
- 'min\_payment\_amt' is positively skewed with the value of 0.4
- Most of the data's are clustered around the left tail of the distribution while the right tail of the distribution is longer.



*Figure 9. Histogram & Boxplot for max\_spent\_in\_single\_shopping*

- Histogram shows the data's distributed from ~4.5 to 6.5
- The 'max\_spent\_in\_single\_shopping' variable has no outliers.
- 'max\_spent\_in\_single\_shopping' is positively skewed with the value of 0.56
- Most of the data's are clustered around the left tail of the distribution while the right tail of the distribution is longer.



Pair plot and Correlation Plot:

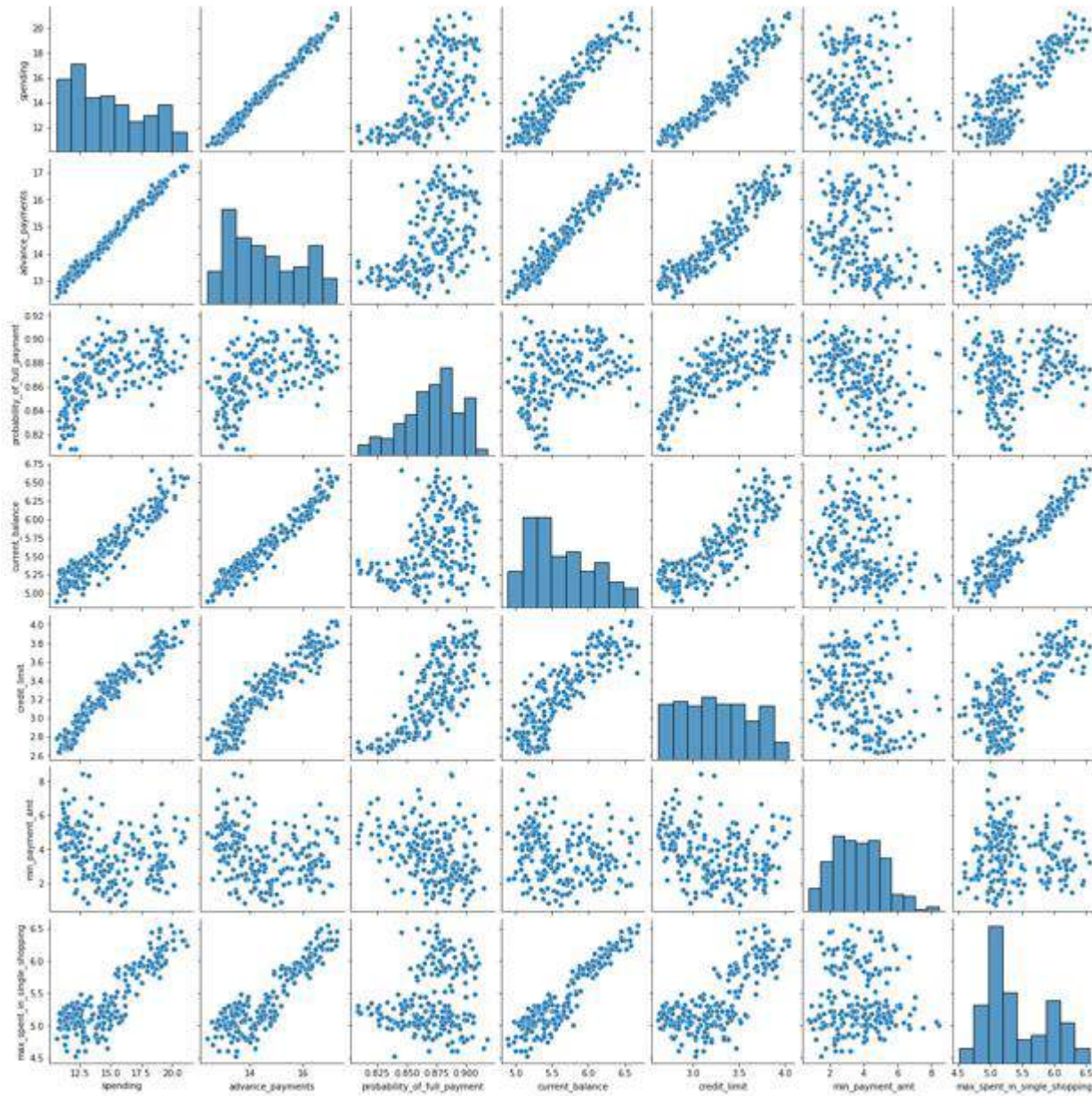


Figure 10. Pair plot for all the numerical variables



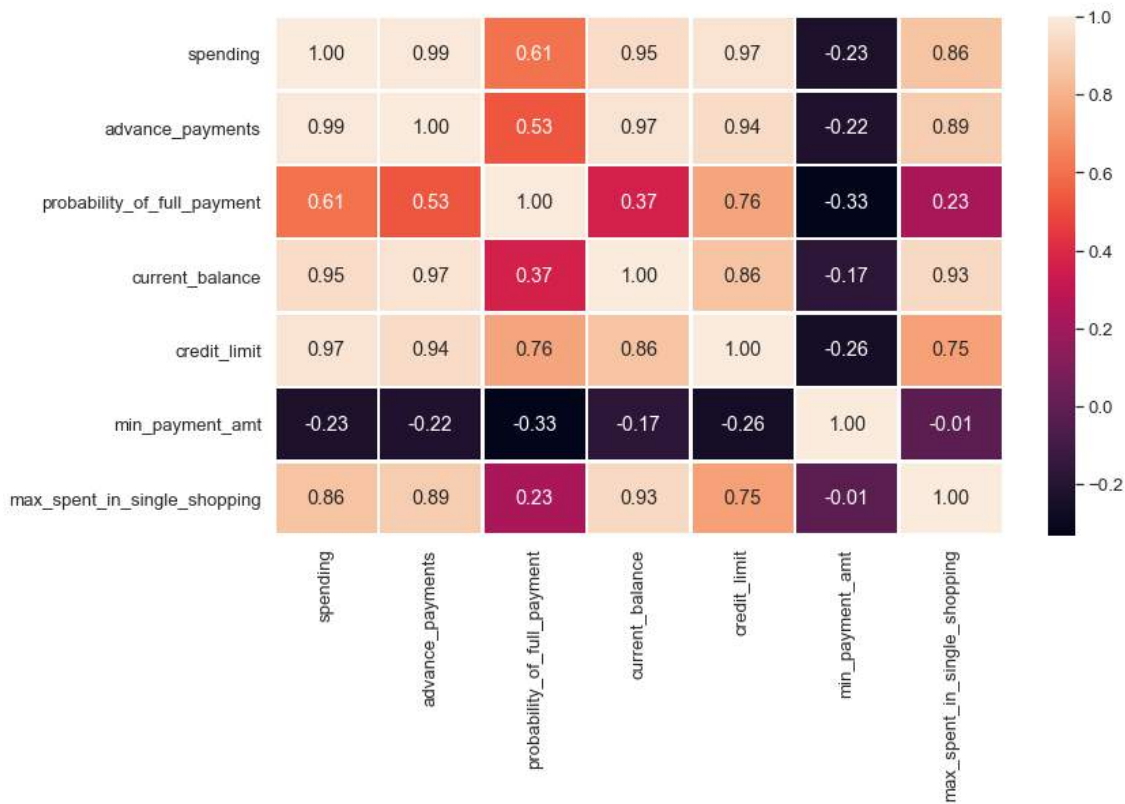


Figure 11. Heat Map for all the numerical variables

## Inferences from correlation plots:

From the correlation heat map, we can see that attributes of the banks are highly correlated in many places. You could see it in orange/brown shaded area.

- Spending is very highly correlated with Advance payments, Current Balance, Credit Limits and Maximum amount spent in single shopping.
- Advance Payments is very highly correlated with Current Balance, Credit Limits and Maximum amount spent in single shopping.
- Current Balance is very highly correlated with Credit Limits and Maximum amount spent in single shopping.
- Credit Limits and Probability of full payment/ Maximum amount spent in single shopping is highly correlated.
- Probability of full payment is moderately correlated with Spending/Advance Payments.
- Minimum payment amount variable is weakly correlated with other attributes.

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

### Summary of the Dataset before Scaling

	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 4. Summary of the Bank Dataset before scaling*

Clustering works with distance based algorithms. Intra-distance clusters (within) are minimized and Inter-distance clusters are maximized. So then, samples are separated (or) associated to the appropriate clusters. In our problem, scaling is necessary as we could see the values are in different scales. For e.g., compare the 'Probability of Full payment' values with other variables. Also, the standard deviation for Spending, Advance payments and Minimum Payment amount is high. By scaling the dataset, data comes to the origin with the unit standard deviation. Here, we can use the common scaling technique called 'z-score' ( $z = \frac{x - \mu}{\sigma}$ ).  $x$  - observed value,  $\mu$  - mean,  $\sigma$  - std. dev.

### Summary of the Dataset after Scaling

	count	mean	std	min	25%	50%	75%	max
spending	210.0	9.148766e-16	1.002389	-1.466714	-0.887955	-0.169674	0.846599	2.181534
advance_payments	210.0	1.097006e-16	1.002389	-1.649686	-0.851433	-0.183664	0.887069	2.065260
probability_of_full_payment	210.0	1.243978e-15	1.002389	-2.668236	-0.598079	0.103993	0.711677	2.006586
current_balance	210.0	-1.089076e-16	1.002389	-1.650501	-0.828682	-0.237628	0.794595	2.367533
credit_limit	210.0	-2.994298e-16	1.002389	-1.668209	-0.834907	-0.057335	0.804496	2.055112
min_payment_amt	210.0	5.302637e-16	1.002389	-1.956769	-0.759148	-0.067469	0.712379	3.170590
max_spent_in_single_shopping	210.0	-1.935489e-15	1.002389	-1.813288	-0.740495	-0.377459	0.956394	2.328998

*Table 5. Summary of the Bank Dataset after scaling*

## 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 6. Scaled data after applying z-score

**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 is a type of clustering approach where records are sequentially grouped to create clusters, based on distance between records and distance between clusters. We will use the 'ward's' method and 'average' linkage method.

Using the scientific python package (scipy.cluster.hierarchy), we import the 'linkage' and 'dendrogram' function.

The linkage is for using the different types of methods in Hierarchical clustering.

In our case, we use 'ward' and 'average' method. Additionally, we leave the affinity (distance) with the default 'euclidean'. i.e., the default distance metric used to measure the distance between clusters and is simply the straight line distance between two points. This is mathematically expressed as:

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

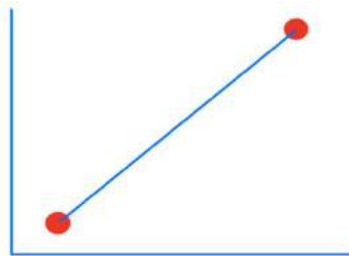


Figure 12. Source: [Towards Data Science](#)

## Ward's Method

Ward's method (a.k.a. *Minimum variance method* or *Ward's Minimum Variance Clustering Method*). Like other clustering methods, Ward's method starts with  $n$  clusters, each containing a single object. These  $n$  clusters are combined to make one cluster containing all objects. In order to select a new cluster at each step, every possible combination of clusters must be considered. The increase in within cluster variance is lesser (*with all possible combination among the clusters*) are getting merged compared to the other combination of distances. Finally, all the clusters are merged into one.

The dendrogram is used for visual representation of the compound correlation data.

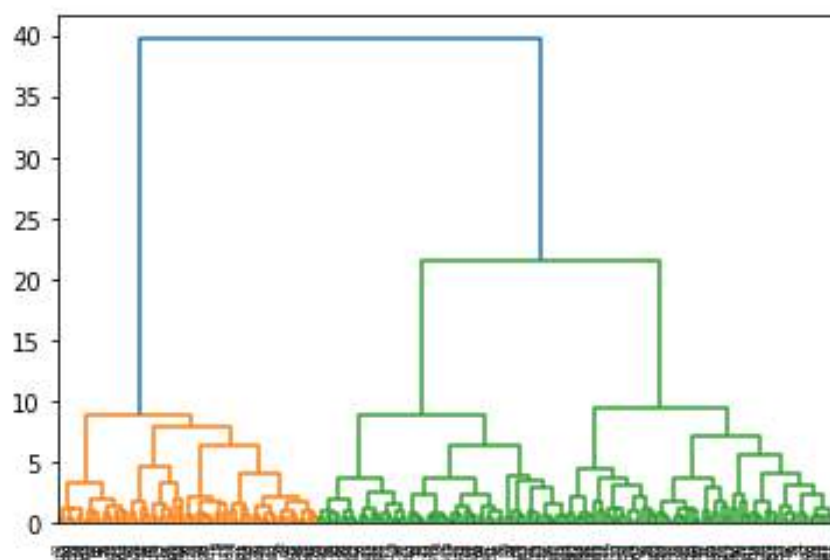


Figure 13. Dendrogram for ward's method

The above dendrogram is the representation for all the every single clusters by using Ward's method.

We will use the 'truncate\_mode' parameter in the dendrogram function to truncate the last 10 merges produces the clean output to the viewer.

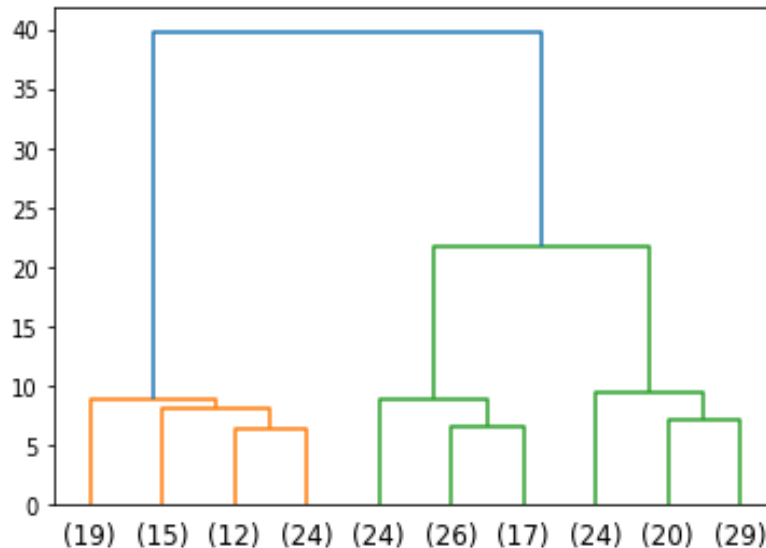


Figure 14. Truncated Dendrogram for ward's method (maxclust criterion)

As you can see, the green cluster shows the maximum number of clusters (records of 24+26+17+24+20+29).

The next step is to obtain the clusters and the credit card users belong to each of the clusters. In order to do that, we will use the same scientific python package (scipy.cluster.hierarchy) and import 'fcluster' function.

### Method 1 – maxclust criterion (ward's method)

Using the fcluster function and the criterion 'maxclust' we are instructing the data's to associate to the clusters passing in the function. In our problem, we use 3 as the maximum number of clusters.

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

Figure 15. Cluster mapping using 'maxclust' criterion

Here, 1<sup>st</sup> records mapped to the cluster 1 and 2 record mapped to cluster 3 and so on.

## Method 2 – distance criterion (ward's method)

Using the fcluster function and the criterion 'distance', will cut the dendrogram at a specific point to see whether the 'maxclust' and 'distance' criterion produces the similar number of cluster outputs.

In our case, we cut the dendrogram at the point 20.

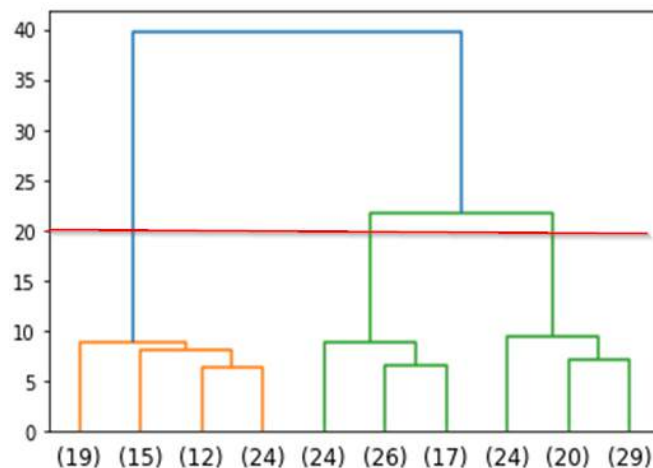


Figure 16. Truncated Dendrogram for ward's method (distance criterion)

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

Figure 17. Cluster mapping using 'distance' criterion

By comparing the outputs of these methods, both of the 'maxclust' criterion output and 'distance' criterion output produces the same output.

Next, we will attach the output of one of the method to the original dataframe and analyze the group of users mapped to 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 7. Clusters mapped to the Dataset

Our records are almost evenly mapped to the clusters.

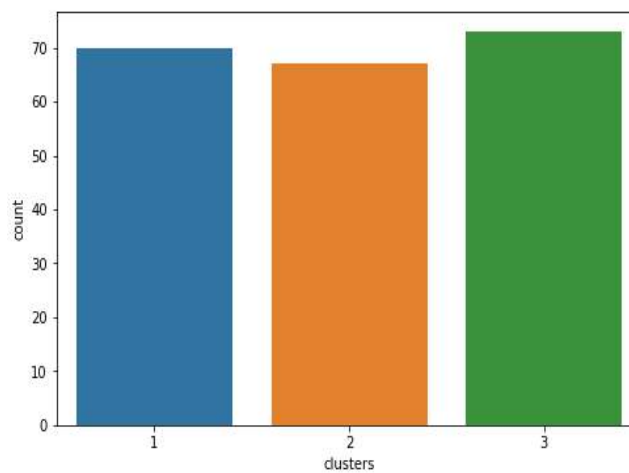


Figure 18. Cluster counts

## Cluster profiling for Ward's Method

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
clusters								
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 8. Cluster profiling for Ward's Method

Cluster 1 has 70 records.

Cluster 2 has 67 records.

Cluster 3 has 73 records.



## Average Method

The distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster.

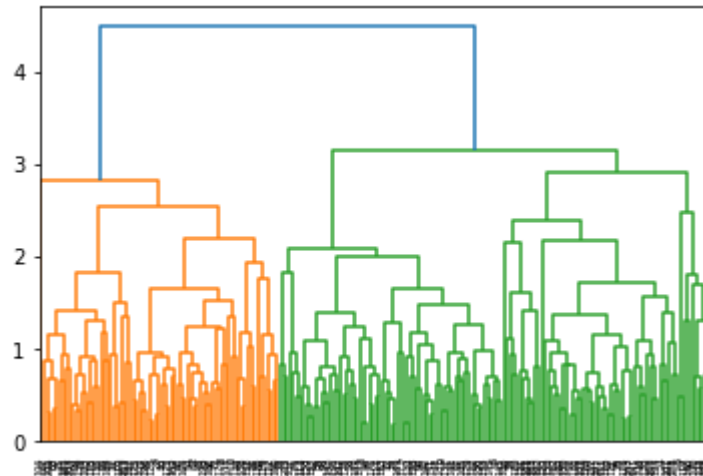


Figure 19. Dendrogram for average method

The above dendrogram is the representation for all the every single clusters by using Average method.

We will use the 'truncate\_mode' parameter in the dendrogram function to truncate the last 10 merges produces the clean output to the viewer.

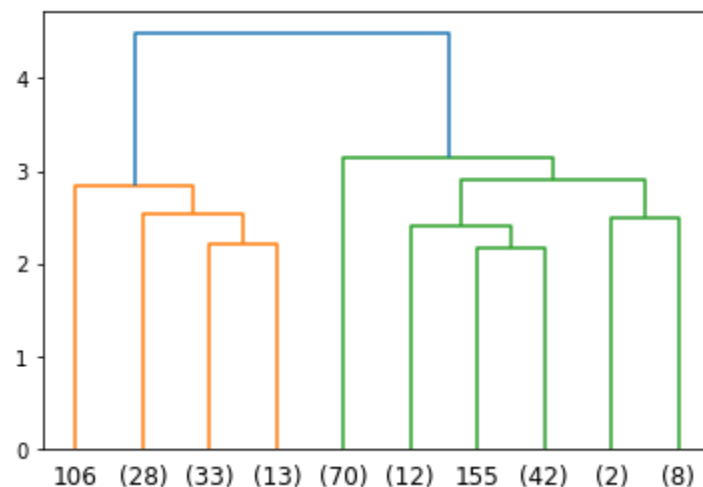


Figure 19. Truncated Dendrogram for average method (maxclust criterion)



The next step is to obtain the clusters and the credit card users belong to each of the clusters. In order to do that, we will use the same scientific python package (scipy.cluster.hierarchy) and import 'fcluster' function.

## Method 1 – maxclust criterion (Average Method)

Using the fcluster function and the criterion 'maxclust' we are instructing the data's to associate to the clusters passing in the function. In our problem, we use 3 as the maximum number of clusters.

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

Figure 20. Cluster mapping using 'maxclust' criterion

Here, 1<sup>st</sup> records mapped to the cluster 1 and 2<sup>nd</sup> record mapped to cluster 3 and so on.

## Method 2 – distance criterion (Average Method)

Using the fcluster function and the criterion 'distance', will cut the dendrogram at a specific point to see whether the 'maxclust' and 'distance' criterion produces the similar number of cluster outputs.

In our case, referring to the dendrogram we cut the dendrogram at the point 3.

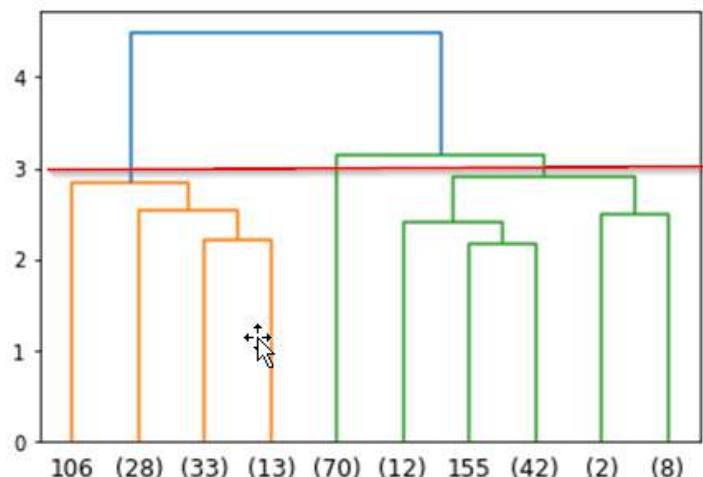


Figure 21. Truncated Dendrogram for average method (distance criterion)

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

Figure 22. Cluster mapping using 'distance' criterion

By comparing the outputs of these methods, both of the 'maxclust' criterion output and 'distance' criterion output produces the same output.

Next, we will attach the output of one of the method to the original dataframe and analyze the group of users mapped to 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 9. Cluster mapping for Average Method to the dataset

### Cluster profiling for Average Method

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
clusters								
1	18.129200	16.058000	0.881595	6.135747	3.648120	3.650200	5.987040	75
2	11.916857	13.291000	0.846766	5.258300	2.846000	4.619000	5.115071	70
3	14.217077	14.195846	0.884869	5.442000	3.253508	2.768418	5.055569	65

Table 10. Cluster profiling for Average Method

Cluster 1 has 75 records.

Cluster 2 has 70 records.

Cluster 3 has 65 records.

## Customer Segmentation for Hierarchical Clustering

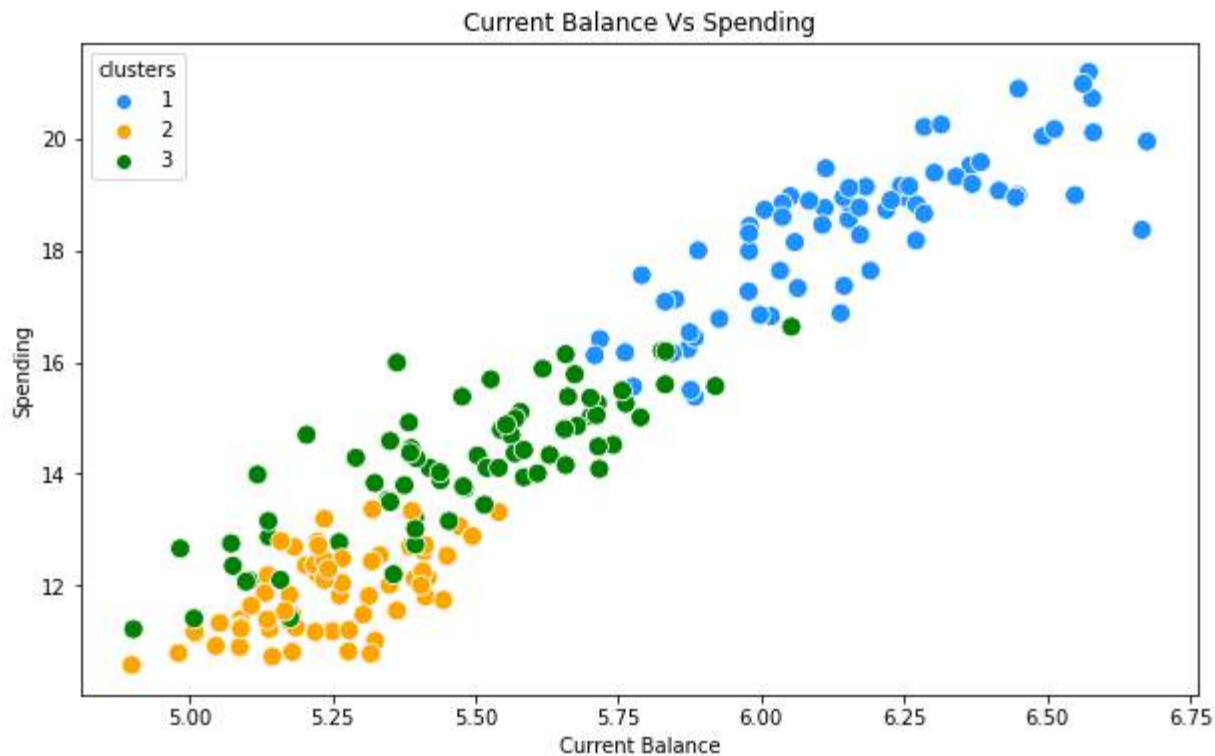


Figure 23. Cluster segmentation for Hierarchical clustering (ward's method)

## Observations

- Both the methods (ward's and average) produce similar means. Based on the dendrogram, we conclude with 3 clusters which tell us High/Medium/Low spending.
- On an average, the users present in cluster 1 have the high spending of 18K, cluster 2 has the low spending of 11.9K, and cluster 3 has the medium spending of 14.2K per month.
- From the above scatter plot for current balance vs spending, the high-spending group has a high balance in their card. There are few moderate-spending group customers with a current balance in their card. Otherwise, 3 clusters are well settled with High, Medium, and Low groups.

**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 Technique

K means clustering is type of unsupervised learning and part of non-hierarchical clustering approach where we specify the number of clusters needed as output lets say, k. K means algorithm groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

From sklearn package, we import the KMeans function.

At a minimal, we fit and run the scaled dataset with 2 clusters initially and with random\_state=1.

```
KMeans(n_clusters=2, random_state=1)
```

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

*Figure 24. Cluster mapping using K Means algorithm*

Here, the value 0 indicates the 'cluster 1' and the value 1 indicates the 'cluster 2' because we've predefined the clusters (*k value*) as 2.

For 3 clusters, 0 indicates the cluster 1, the value 1 indicates the cluster 2 and the value 2 indicates the cluster 3.

To find the optimum number of clusters, we will use elbow curve method.

Lets calculate within sum of squares(inertia) for the range of clusters from 1 to 11, will run through the loop from 1 to 10 cluster K value and plot the elbow curve.

```
[1469.9999999999995,  
659.1717544870411,  
430.65897315130064,  
371.301721277542,  
327.9608240079031,  
290.5900305968219,  
264.83153087478144,  
240.6837259501598,  
220.85285825594738,  
206.3829103601579]
```

Figure 25. Inertia values from K=1 to K-10

## Elbow curve

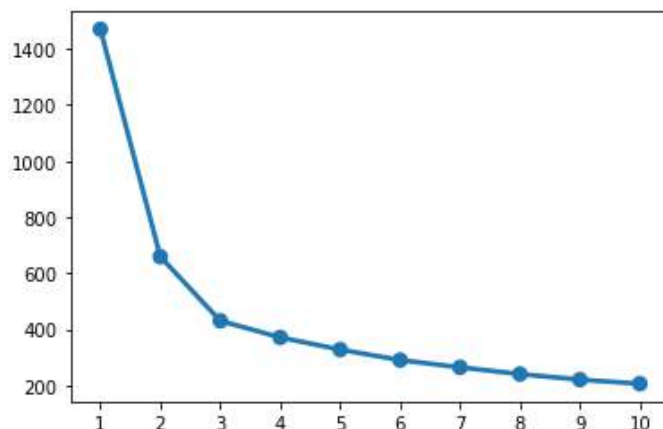


Figure 26. Elbow curve from K=1 to K-10

From the point plot, we could see, it is advisable to run the scaled dataset with 3 clusters. Because, after the cluster 3 we observed that there is no huge drop in the inertia values.

Let's figure out the silhouette\_score for the 3 clusters. Silhouette\_score is a indirect model evaluation techniques which we can verify once clustering procedures are completed namely the K-means model which is distance based.

We import silhouette\_score, silhouette\_samples function from sklearn package.

```
KMeans(n_clusters = 3,random_state=1)
silhouette_score(scaled_df,labels,random_state=1)
```

silhouette\_score for 3 clusters is 0.400727

When we run with 4 clusters, silhouette\_score value is 0.32757. It once again confirms, cluster 3 holds good for our dataset.

We conclude with 3 number of clusters & fit it into our original dataset.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Clus_kmeans3
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	2
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	0
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837	1

Table 11. Cluster mapping for K Means

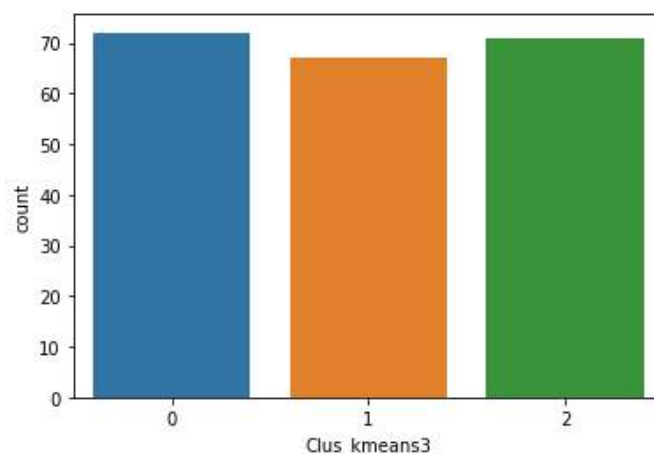


Figure 27.Cluster counts for K Means

Our records are almost evenly mapped to the clusters.

## Customer Segmentation for K Means Clustering

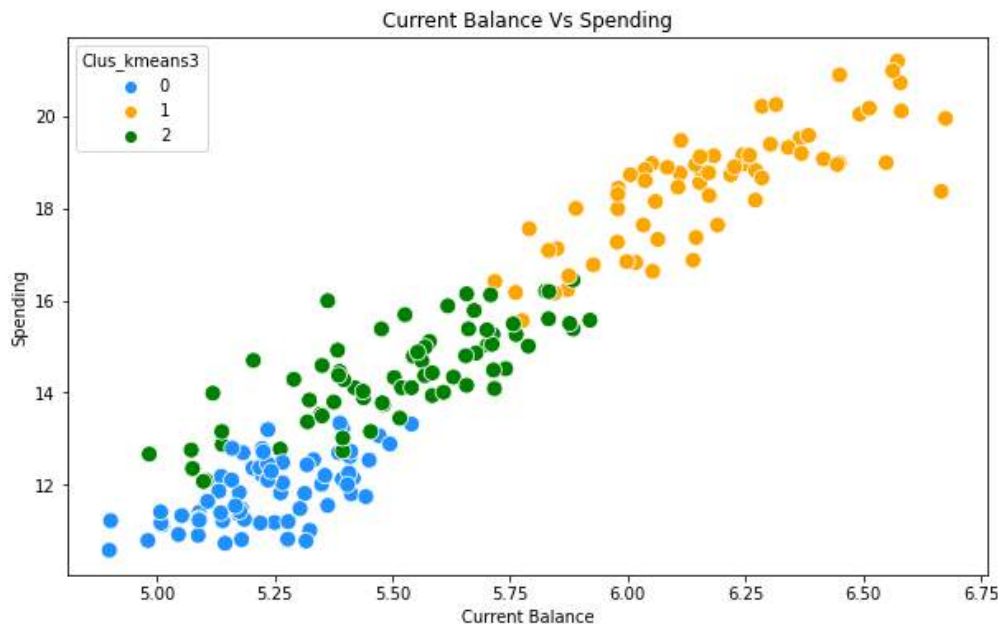


Figure 28. Cluster Segmentation for K Means

From the above scatter plot for current balance vs spending, high spending group has high balance in his card. There are few moderate spending group customers has current balance in his card. Otherwise, 3 clusters are well settled with High, Medium and Low groups.

The minimum silhouette\_samples value for 3 cluster is 0.002713 (*greater than 0*) which tells us 3 is the suitable number of clusters and the clusters are well settled for our business bank dataset.

**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.**



## Cluster profiling for K Means Technique

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq	
Clus_kmeans3									
0	11.856944	3	13.247778	0.848253	5.231750	2.849542	4.742389	5.101722	72
1	18.495373	1	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701	67
2	14.437887	2	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803	71

*Table 12. Cluster profiling for K Means*

Cluster 0 – Low spending group, has 72 records.

Cluster 1 – High spending group, has 67 records.

Cluster 2 – Medium spending group, has 71 records.

### Cluster 1(High Spending Group)

- Promote offers for buying luxury cars, gadgets.
- Provide discounts/cash back offers for branded apparels, refreshments, Dining.
- Offer loans and increase the credit limit as their advance payment figure is good.
- Offer discounts for airline and hotel bookings.

### Cluster 2(Medium Spending Group)

- Offer cash back for fuel & order foods online.
- Provide discounts for budgeted motor vehicles.
- Allow customers to withdraw cash from ATMs.
- Increase the amount on travel and rental car insurance.



#### **Cluster 0(Low Spending Group)**

- Their minimum payment due amount is good compared to the high spending group. Offer reward points buying specific grocery,essential items from ecommerce sites.
- Offer cash back for booking movie tickets, phone, electric, gas bills etc.
- As their maximum spent in single shopping is quite good, tie up with apparels, kitchen items.
- Recommend sending polite payment reminder for payments.

The above suggestions might encourage the users to spend more on their card usage.

## 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.

### Executive Summary

An Insurance firm providing tour insurance is facing higher claim frequency. The management decides to collect data from the past few years. We will create a model which predicts the claim status and provide recommendations to management. Here, we are going to use Decision Tree Classifier, Random Forest and Artificial Neural Network model and compare the models' performances in train and test sets.

### Introduction

The purpose of this whole exercise is to build the CART, RF and ANN model using the Insurance firm dataset collected from 3000 samples. The dataset consists of 3000 rows and 10 columns, among the variables, the column 'Claimed' is the target variable which tells us whether claim has been made or not. Once the models are trained, we will run the model using the test set and compare each model's precision/recall value to finalize the model for our business problem.

**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.**

The insurance firm provided dataset consists of 3000 records and 10 columns. The column 'Claimed' is the target variable which indicates the customer made the tour insurance claim or not. Other variables are independent variables. These models are part of supervised learning where the variables are defined for our analysis.

## Sample of the Salary Data dataset

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

*Table 13. Insurance Dataset Sample*

## Data Description

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)

Let us check the types of variables in the data frame.

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

*Figure 27. Data Types*

There are only 4 continuous columns such as Age, Commission, Duration and Sales. Other variables such as Agency Code, Type, Claimed, Channel, Product Name and Destination are object data types which we will be encoding shortly for our model building.

```

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 29.Data information

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

## Summary of the Dataset

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

Table 14.Insurance Dataset Summary

## Observations

Age: Minimum age is 8 and maximum 84, which looks good.

Commission: Sometimes, the commission received for tour insurance firm goes above 100% which is fine.

Duration: Note, the minimum value is -1. This needs to be corrected because duration of the tour is supposed to be in positive values. We will replace it with mode value.

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

Table 15. Anomalies in Duration variable

The highest frequency of duration of the tour is 10; let's replace -1 with 10.

Sales: The average sales per customer procuring tour insurance policies are 6000/- rupees.

Standard deviation is high for all the integer variables and the values are in different scales. So, we need to standardize the dataset before building the model.

### Check for duplicate values in the dataset

There are 139 duplicate records are present in the dataset. It is just 4% of the records on the whole dataset. We will be dropping them before building the model.

After dropping them, we have 2861 records present in the dataset.

### Outliers Proportion in numerical columns

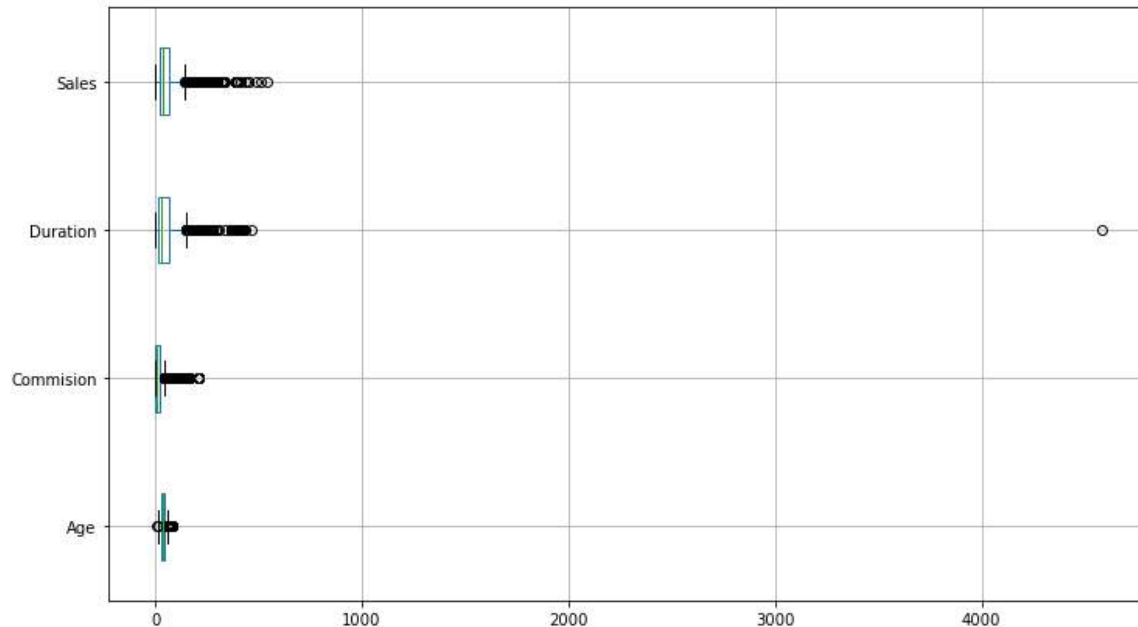


Figure 30. Histogram for all numerical variables to find the distribution

Outliers present in the all the continuous columns and their proportions are mentioned below.

Age: 4 . 54%

Commission: 12 . 37%

Duration: 12 . 65%

Sales: 12 . 09%

Not needed to treat the outliers for Decision Tree Classifier Model. But for Random Classifier which deals with averages of multiple decision trees and ANN which deals with weights of every node, it is advisable to treat the outliers.

Check the null values

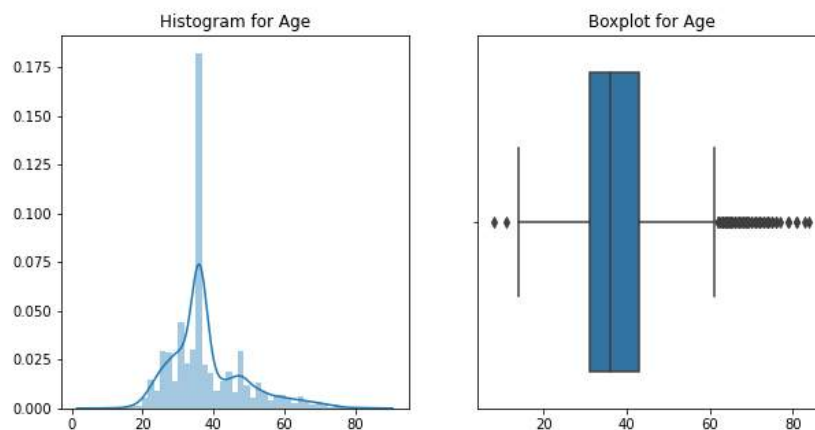
```
Age          0
Agency_Code 0
Type         0
Claimed      0
Commision    0
Channel      0
Duration     0
Sales        0
Product Name 0
Destination  0
dtype: int64
```

*Figure 31.Null Values check*

There are no null values present in the dataset.

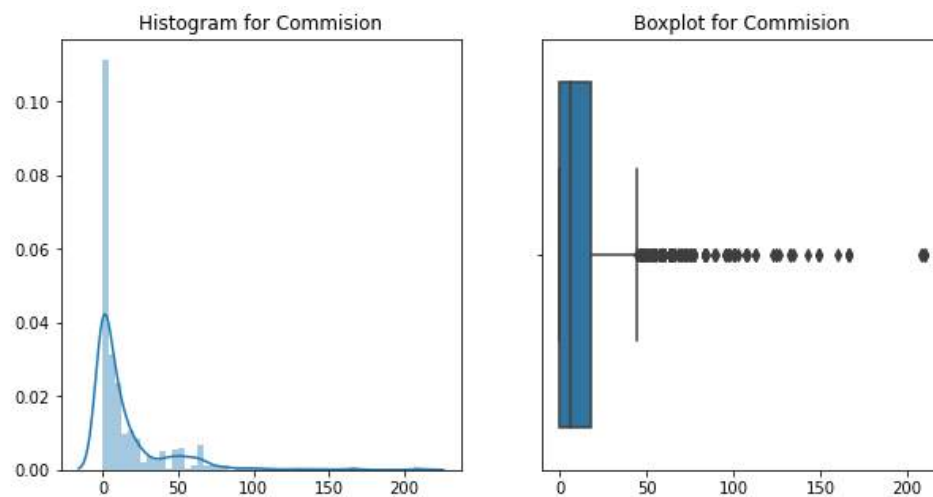
## Histograms and Boxplots

Below graphs helps us identifying the data distribution for continuous columns.



*Figure 32.Histogram and Box plot for Age*

- Histogram shows the data's distributed from ~8 to 84
- The 'Age' variable has outliers.
- 'Age' is positively skewed with the value of 1.1031446044352335
- Most of the data's are clustered around the left tail of the distribution while the right tail of the distribution is longer.



*Figure 33. Histogram and Box plot for Commission*

- Histogram shows the data's distributed from 0 to ~90; there are few records go beyond 90. And we can notice in the line travelling along the x-axis.
- The 'Commission' variable has outliers.
- 'Commission' is positively skewed with the value of 3.1047406576922842
- Most of the data's are clustered around the left tail of the distribution while the right tail of the distribution is longer.

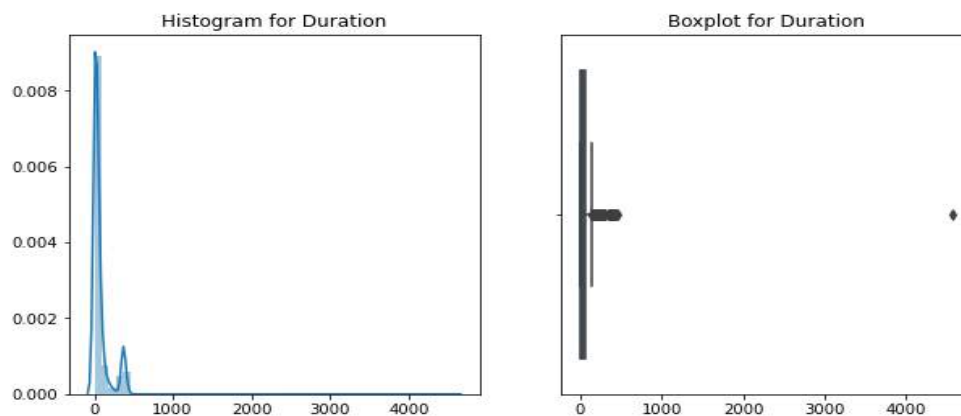


Figure 34. Histogram and Box plot for Duration

- Histogram shows the data's distributed from 0 to ~450.
- The 'Duration' variable has outliers. One of the records is clearly standing out from the crowd and the value is 4580 which is definitely a bad data and it is to be treated.
- 'Duration' is positively skewed with the value of 13.786614016519017
- Most of the data's are clustered around the left tail of the distribution while the right tail of the distribution is longer.

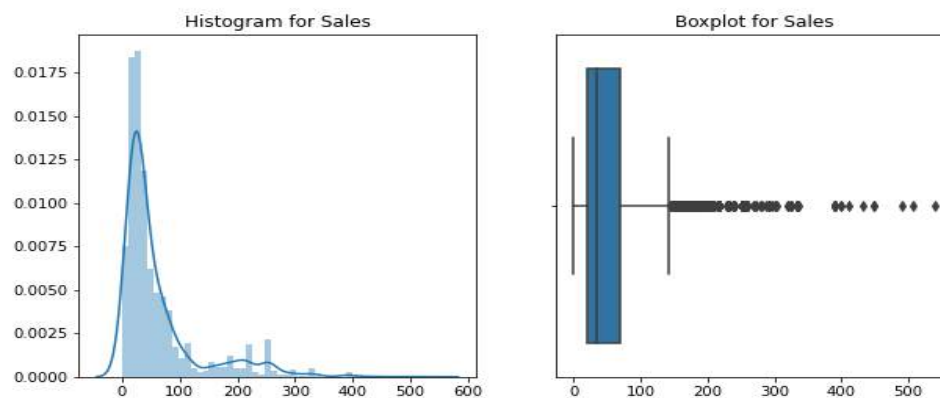


Figure 35. Histogram and Box plot for Sales

- Histogram shows the data's distributed from 0 to ~350.
- The 'Sales' variable has outliers. Few records are out from the crowd.
- 'Sales' is positively skewed with the value of 2.3446426921667585
- Most of the data's are clustered around the left tail of the distribution while the right tail of the distribution is longer.



## Correlation Plots

### Pair plot

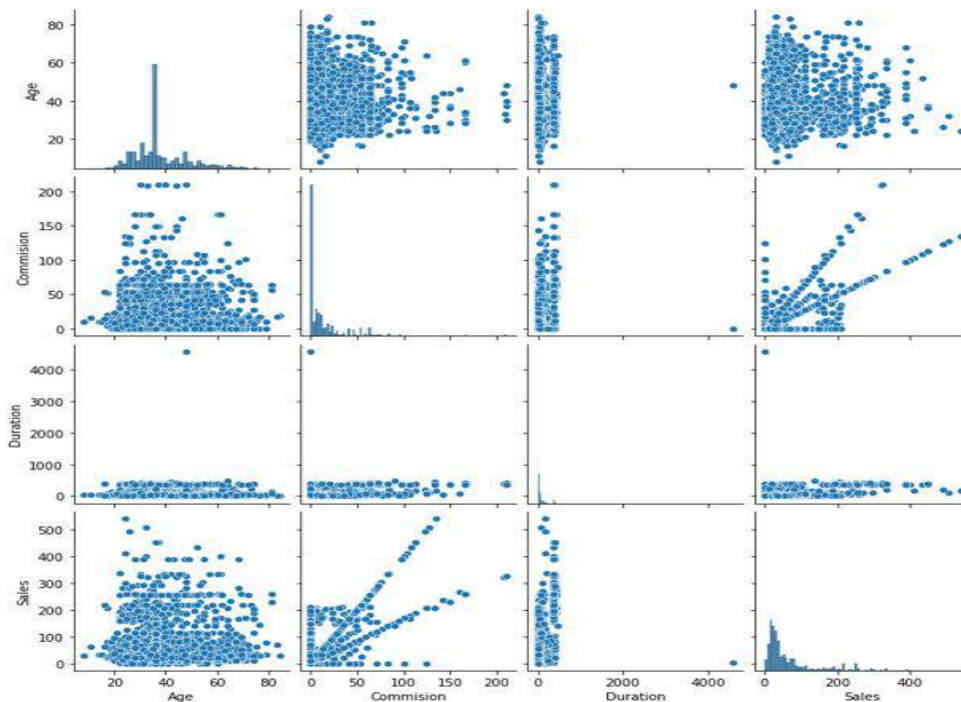


Figure 36. Pair plot to see the correlation between the variables

Sales and Commission are highly correlated.

### Heat Map



Figure 37. Heat Map to see the correlation between the variables

## Observations

- Sales and Commission are highly correlated.
- Duration and Sales/Commission are moderately correlated.
- Age and Commission/Duration/Sales are very weakly correlated.
- In general, observers that the multi-collinearity between the attributes is not much.

## Categorical Variables

### Agency Code

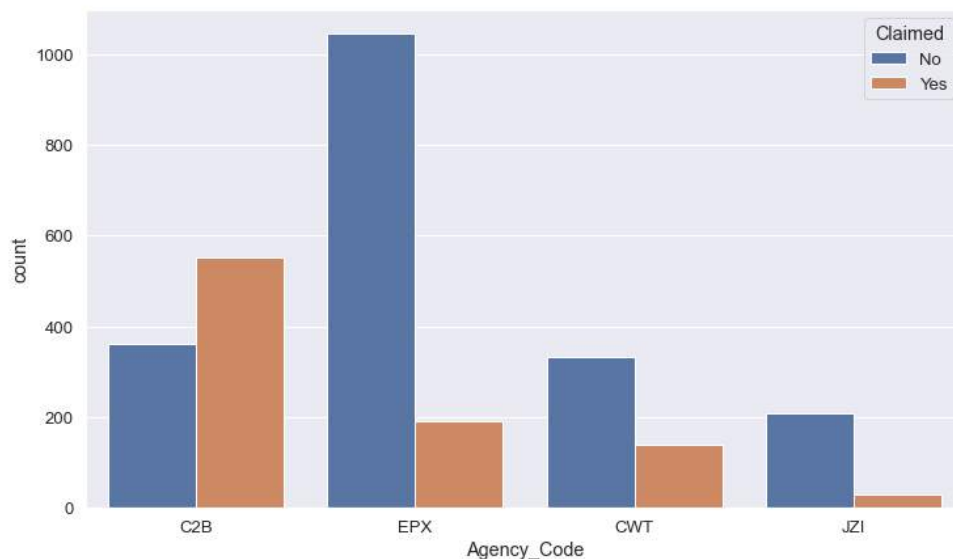


Figure 38.Count plot for Agency Code

- The Agency code C2B and JZI belong to Airlines.
- The Agency code EPX and CWT belong to Travel Agency.
- The Agency C2B shows the maximum number of claims has been made.

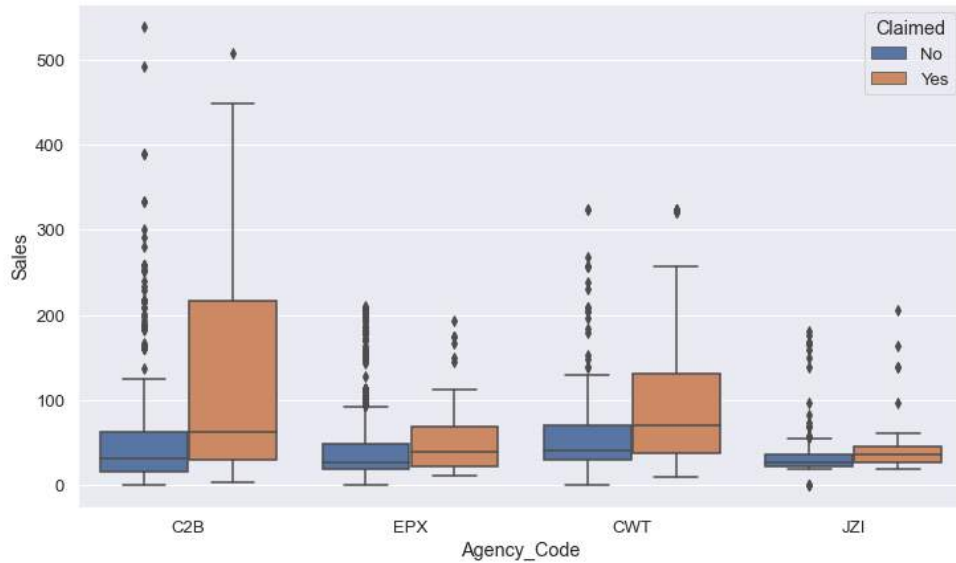


Figure 39.Count plot for Agency Code Vs Sales

C2B from Airlines type plays the vital role procuring the insurance policies.

## Type

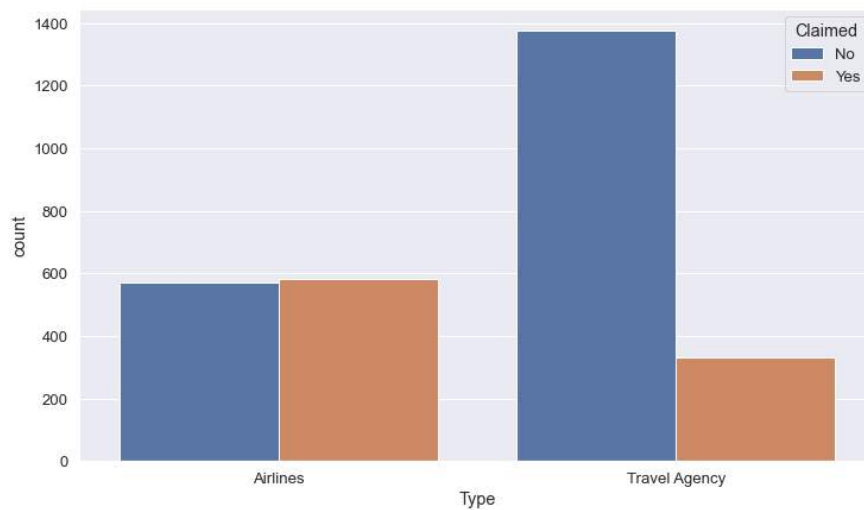


Figure 40.Count plot for Type

Airlines are the type where maximum number of claim has been made.

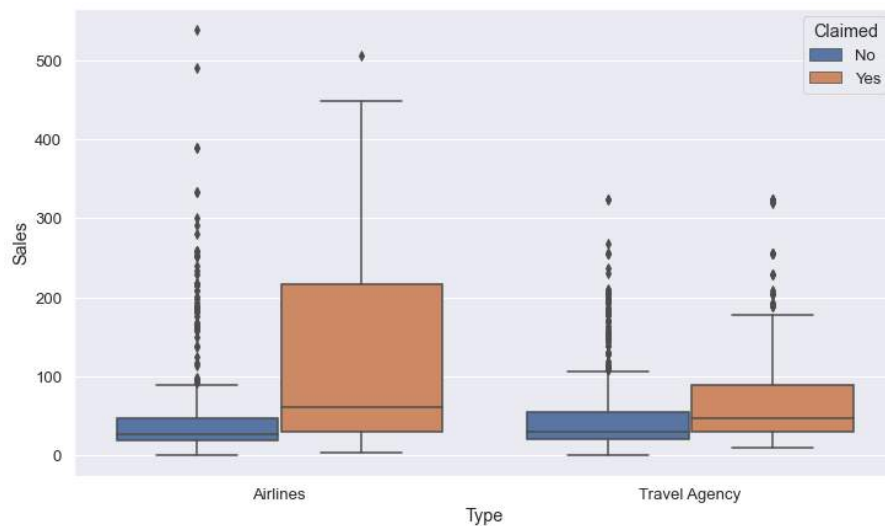


Figure 41.Count plot for Type Vs Sales

Airlines are the type where sales per customer are maximum in procuring the insurance policies.

## Channel

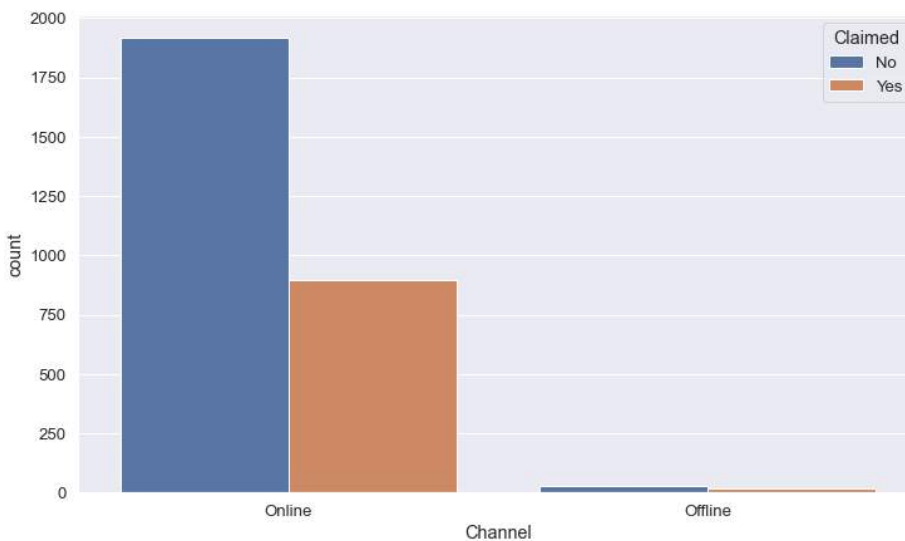


Figure 42.Count plot for Channel

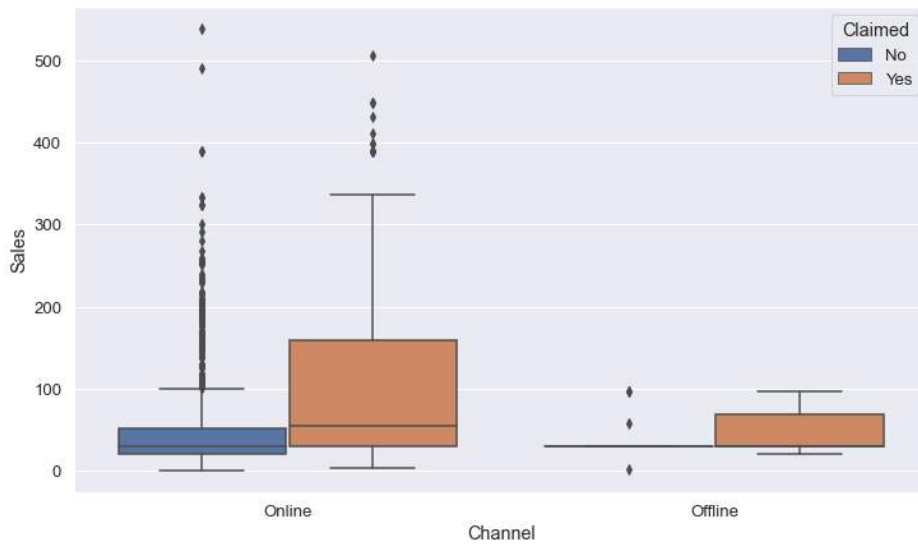


Figure 43.Count plot for Channel Vs Sales

Through Online channel, above 70% of claim has been made.

## Product Name

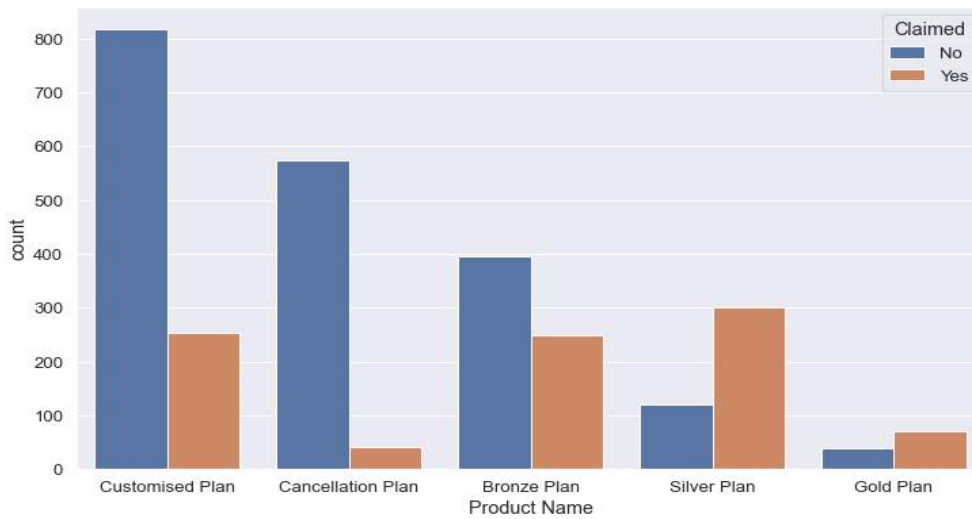


Figure 44.Count plot for Product Name

The customer buys the silver, Bronze and customized plan makes the most number of claims.

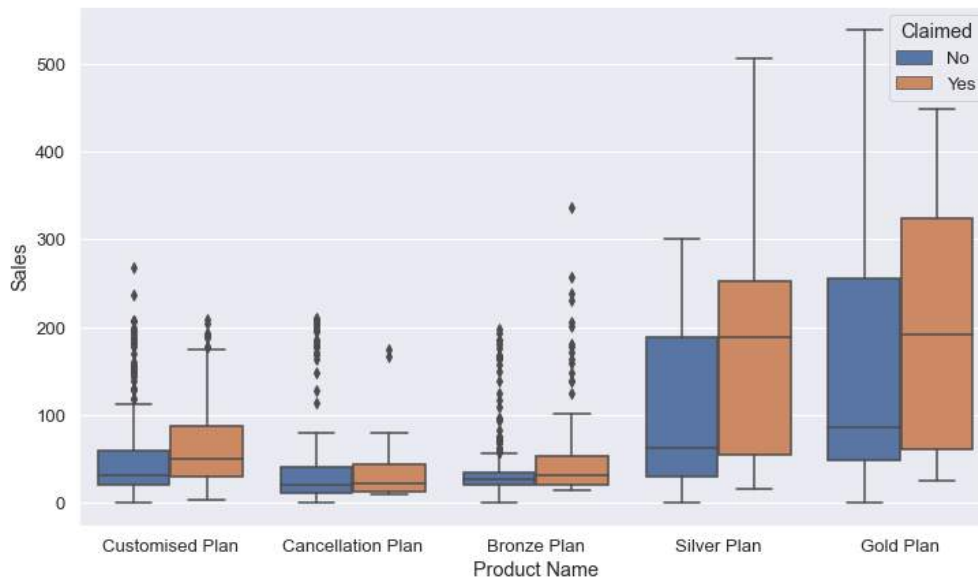


Figure 45. Count plot for Product Name Vs Sales

As expected, the Gold and Silver plan policy amount worth of sales are high & the claim amount is high too.

## Destination

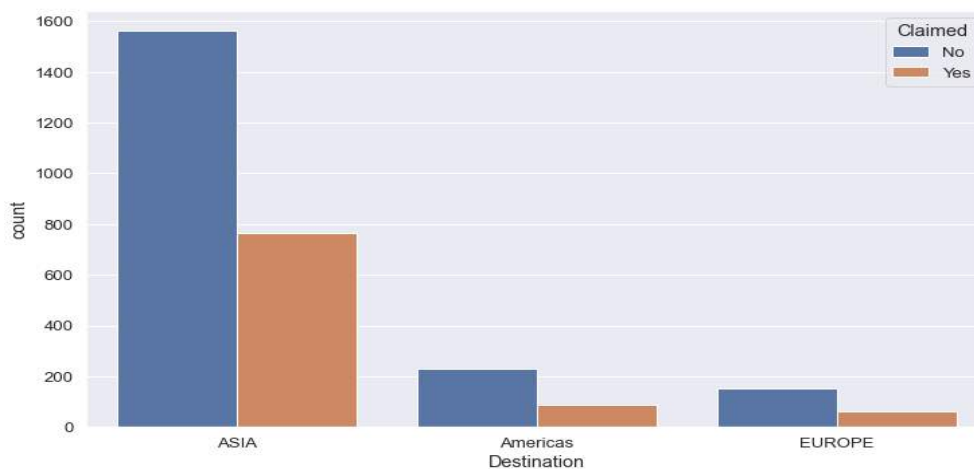


Figure 46. Count plot for Destination

ASIA is the favorite destination where people take part in trip & perhaps the claims are high travelled to that region.

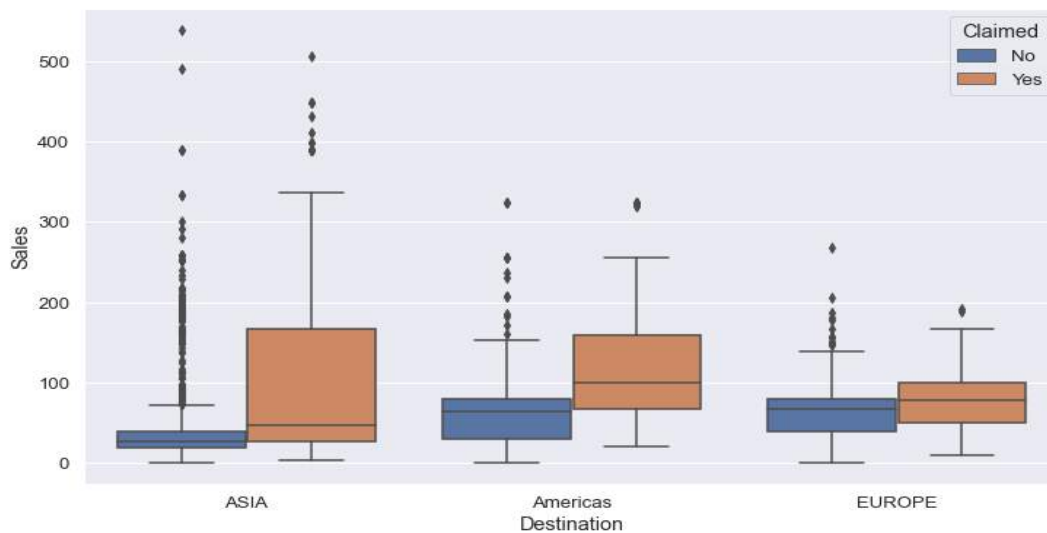
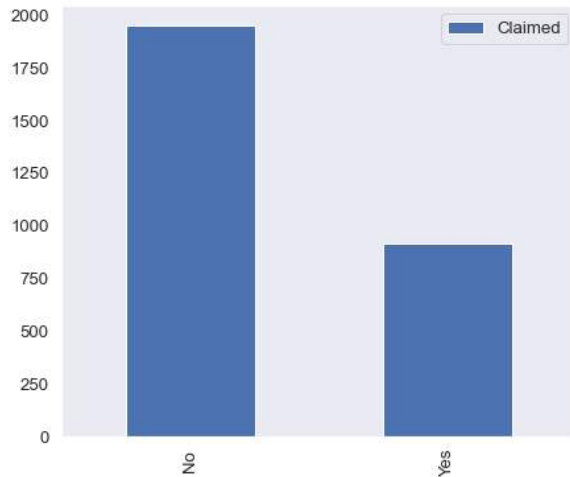


Figure 47.Count plot for Destination Vs Sales

Claims made by the customers travelled to EUROPE are comparatively less to ASIA and AMERICA's destination.

## Claimed



```
No      1947
Yes      914
Name: Claimed, dtype: int64
```

Figure 48.Count plot for Claimed

About 31% of the customers made tour insurance claims.

Value counts of each column to see if anomalies are present in the dataset.

```
Agency Code
EPX      1238
C2B      913
CWT      471
JZI      239
Name: Agency_Code, dtype: int64
-----
Type
Travel Agency    1709
Airlines         1152
Name: Type, dtype: int64
-----
Claimed
No      1947
Yes      914
Name: Claimed, dtype: int64
-----
Channel
Online    2815
Offline    46
Name: Channel, dtype: int64
-----
Product Name
Customised Plan    1071
Bronze Plan        645
Cancellation Plan   615
Silver Plan        421
Gold Plan          109
Name: Product Name, dtype: int64
-----
Destination
ASIA      2327
Americas   319
EUROPE     215
Name: Destination, dtype: int64
-----
```

There are no anomalies or bad data's present in 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.**

### Converting object data types to categorical codes

By pd.Categorical method, let's convert all the object types and create unique codes for the object columns.

```
feature: Agency_Code
['C2B', 'EPX', 'CWT', 'JZI']
Categories (4, object): ['C2B', 'CWT', 'EPX', 'JZI']
[0 2 1 3]
```

```
feature: Type
['Airlines', 'Travel Agency']
Categories (2, object): ['Airlines', 'Travel Agency']
[0 1]
```

```
feature: Claimed
['No', 'Yes']
Categories (2, object): ['No', 'Yes']
[0 1]
```

```
feature: Channel
['Online', 'Offline']
Categories (2, object): ['Offline', 'Online']
[1 0]
```

```
feature: Product Name
['Customised Plan', 'Cancellation Plan', 'Bronze Plan', 'Silver Plan', 'Gold Plan']
Categories (5, object): ['Bronze Plan', 'Cancellation Plan', 'Customised Plan', 'Gold Plan', 'Silver Plan']
[2 1 0 4 3]
```

```
feature: Destination
['ASIA', 'Americas', 'EUROPE']
Categories (3, object): ['ASIA', 'Americas', 'EUROPE']
[0 1 2]
```

```
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   int64
7   Sales            2861 non-null   float64
8   Product Name     2861 non-null   int8
9   Destination      2861 non-null   int8
dtypes: float64(2), int64(2), int8(6)
memory usage: 193.1 KB
```

*Figure 48. Verify data types after conversion*

Now, all the objects are converted to integers which are required for building the model.

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

*Table 16. Sample Dataset after encoding*

Here, we assign independent columns to variable X and target to Y.

Using the sklearn package, we import train\_test\_split function. Split the dataset, one for training the model and another one for test the model (*unseen data by the model*).

Throughout this problem, we run the model with random state=1 to be consistent across the results.

```
from sklearn.model_selection import train_test_split
cX_train, cX_test, ctrain_labels, ctest_labels = train_test_split(X, y,
test_size=.30, random_state=1)
```

Assign 30% of the data's to the test set and 70% for training the model.

- Train set contains 2002 records and 9 columns.
- Test set contains 859 records and 9 columns.

Target variable counts and percentage for Train & Test

	Train Labels		Test Labels	
Counts	0	1359	0	588
	1	643	1	271
Ratio	0	0.678821	0	0.684517
	1	0.321179	1	0.315483

Ratio of split between train dataset and test dataset is nominal.

## Building the Decision Tree Classifier

```
dt_model = DecisionTreeClassifier(criterion = 'gini', random_state=1)
dt_model.fit(cX_train, ctrain_labels)
```

Feature Importance Check

	Imp
Duration	0.280493
Sales	0.229321
Age	0.187876
Agency_Code	0.168711
Commision	0.072804
Product Name	0.028635
Destination	0.028253
Channel	0.003908
Type	0.000000

Figure 49. Feature Importance Check

After going through the Decision Tree, we will get the rough estimation on few parameters.

**Criterion** – gini is the common technique, it will go through the all the columns individually and calculate the gini index value. So, the feature holds the maximum gini index value where the split the starts. In our case, Agency code gini index value is 0.436.

**max\_depth** – Our decision node gets break at the depth 4 or 5 levels.

**min\_samples\_leaf** – At the depth 3, 4 or 5, the minimum number of samples at leaf node under 70.

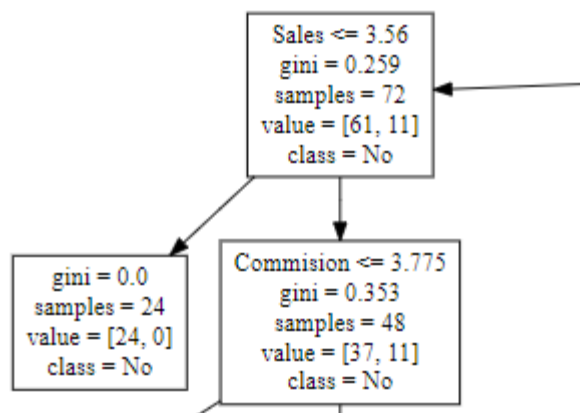


Figure 50.Sales leaf node

**min\_samples\_split** - The minimum number of samples required to split an internal node, the ranges are between 100 and 300 mostly.

Grid Search to find out the optimal hyper parameters for CART training set

```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(random_state=1),
             param_grid={'criterion': ['gini'], 'max_depth': [4, 5],
                          'min_samples_leaf': [15, 20, 30],
                          'min_samples_split': [100, 150, 200]})
```

Best Parameters are

```
{'criterion': 'gini', 'max_depth': 4, 'min_samples_leaf': 30, 'min_samples_split': 200}
```

## Regularizing the tree using the best grid parameters

To get the better precision and recall, fine tuning further randomly with close to the best grid values.

```
DecisionTreeClassifier(max_depth=4, min_samples_leaf=20, min_samples_split=300,  
                      random_state=1)
```

## Feature Importance's

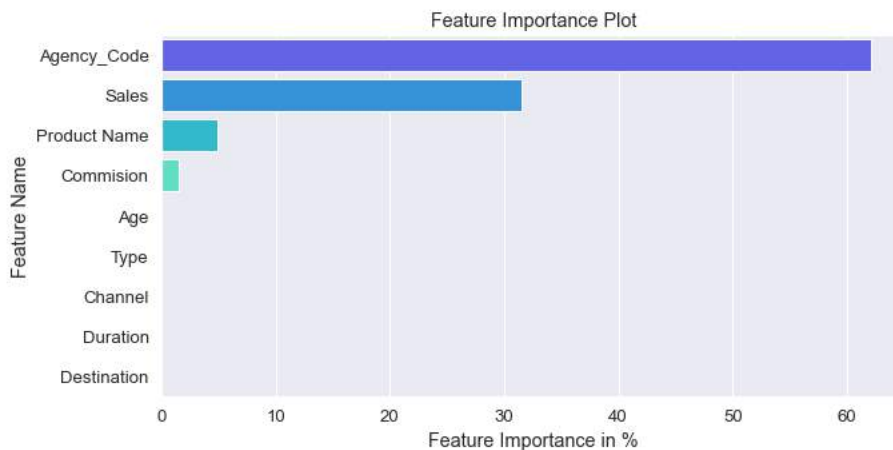


Figure 51. Feature importance for CART

Using the feature, such as Agency code, sales, product Name and commission alone, we are able to make the predictions. Moreover, Agency code and sales plays an important role.

## Predicting on Training and Test dataset

```
ct_ytrain_predict = ct_best_grid.predict(cX_train)  
ct_ytest_predict = ct_best_grid.predict(cX_test)
```

## Building the Random Classifier Model

Random Forest Model build multiple Decision Trees and merges them to get a more accurate and stable prediction. Since the Random Classifier considers the averages of multiple decision trees, it is advisable to treat the outliers.

Boxplot to verify after treating the outliers

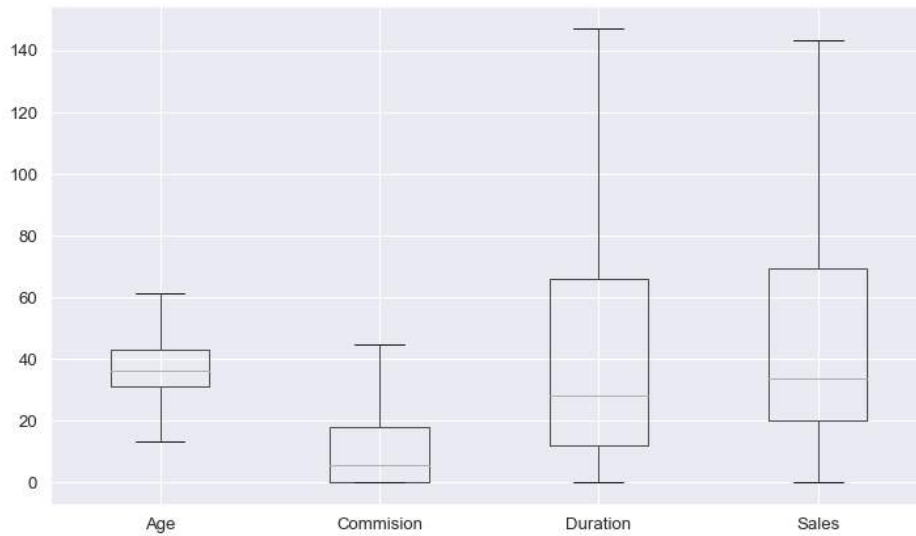


Figure 52.Box plots after treating the outliers

Correlation Map

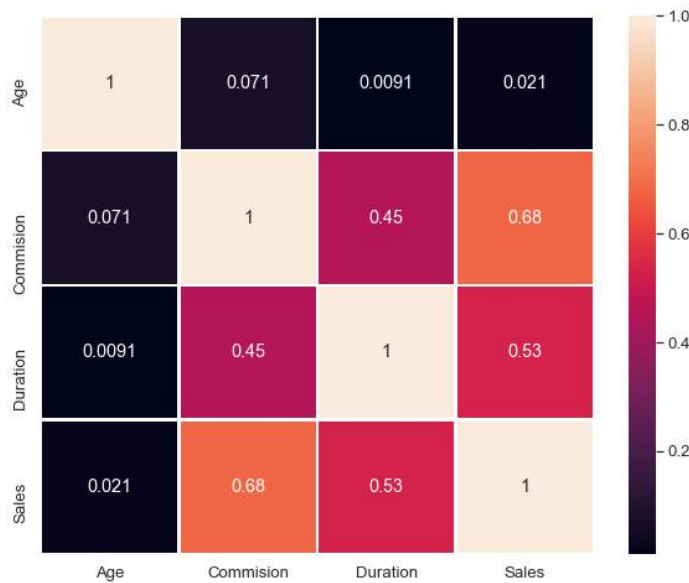


Figure 53.Heat Map after treating the outliers

Our Inferences still holds true even after treating the outliers.

- Sales and Commission are highly correlated.
- Duration and Sales/Commission are moderately correlated.
- Age and Commission/Duration/Sales are very weakly correlated.
- In total, multi collinearity between the attributes is not much.

```
rfcl = RandomForestClassifier(random_state=1)
```

Grid Search to find out the optimal hyper parameters for RF training set

```
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [5, 6], 'max_features': [5],
                          'min_samples_leaf': [8, 10],
                          'min_samples_split': [20, 30],
                          'n_estimators': [150, 250]},
             scoring='recall')
```

Since we have run the CART model before, we would be assuming with the some of the hyper parameters.

**max\_depth** – Decision nodes ends maximum at 4, 5 or 6 in CART, let's choose around the same. Because, even in Random classifier model, the process will be building the multiple DTs and considering the maximum outcomes from the multiple DTs.

**min\_samples\_leaf** – samples at leaf node in CART is approximately around 50, some leaf holds 24. So, let's run it with lesser combinations.

**n\_estimators** - the number of trees you want to build before taking the maximum voting or averages of predictions. Let's run the model between 100 and 300.

Best parameters are

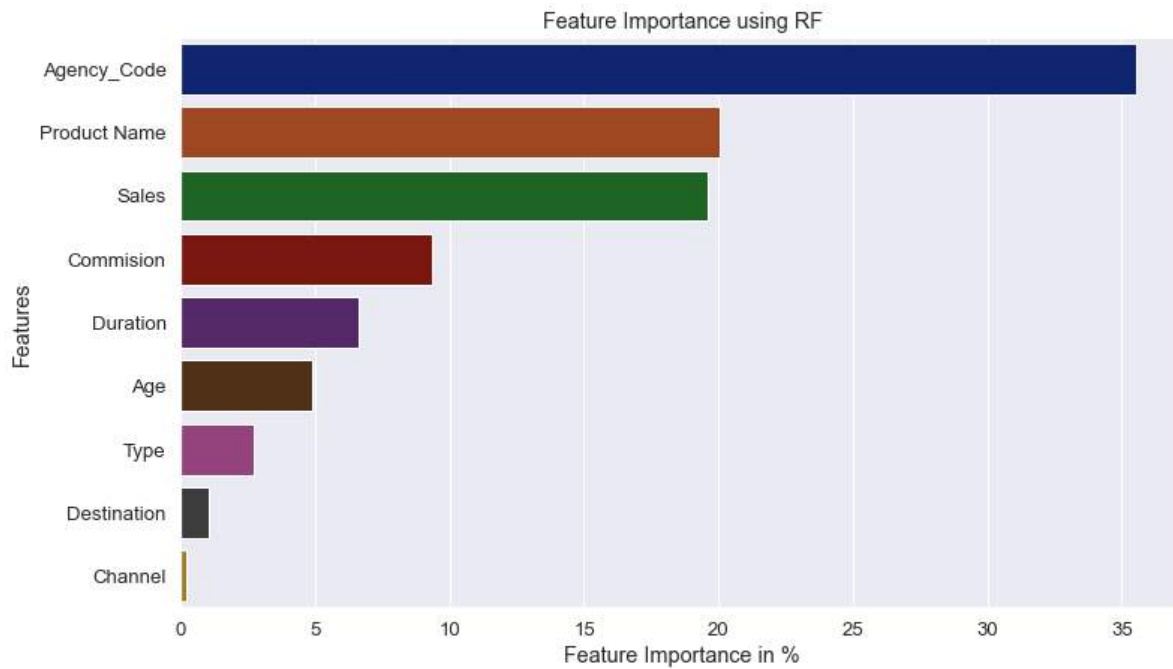
```
{'max_depth': 6, 'max_features': 5, 'min_samples_leaf': 8, 'min_samples_split': 30,
 'n_estimators': 250}
```

After fine tuning further combinations, we attain the maximum precision/recall with the below hyper parameters

```
rfcl = RandomForestClassifier(n_estimators = 250, min_samples_leaf=8,
                             min_samples_split=20, max_features=5, max_depth=5, random_state=1)
```

We will fit these parameters into the training dataset.

## Feature Importance's



*Figure 53. Feature importance's for RF Model*

By looking at the graph, RF considers all the features for predictions. Seems to be a good fit for our business problem. However, we will go through with our results.

## Predicting on Training and Test dataset

```
ytrain_predict = best_grid.predict(X_train)
```

```
ytest_predict = best_grid.predict(X_test)
```



## Building the Artificial Neural Network Model

The ANN calculates the synoptic weights at every node; it is advisable to treat the outliers.

	count	mean	std	min	25%	50%	75%	max
Age	2861.0	37.896190	9.821593	13.0	31.0	36.00	43.00	61.00
Agency_Code	2861.0	1.280671	1.003773	0.0	0.0	2.00	2.00	3.00
Type	2861.0	0.597344	0.490518	0.0	0.0	1.00	1.00	1.00
Claimed	2861.0	0.319469	0.466352	0.0	0.0	0.00	1.00	1.00
Commision	2861.0	11.756865	15.502632	0.0	0.0	5.63	17.82	44.55
Channel	2861.0	0.983922	0.125799	0.0	1.0	1.00	1.00	1.00
Duration	2861.0	47.342887	47.294283	0.0	12.0	28.00	66.00	147.00
Sales	2861.0	51.085089	42.604294	0.0	20.0	33.50	69.30	143.25
Product Name	2861.0	1.666550	1.277822	0.0	1.0	2.00	2.00	4.00
Destination	2861.0	0.261797	0.586239	0.0	0.0	0.00	0.00	2.00

*Table 17. Sample Dataset after encoding*

Standard deviation is high for all the integer variables and the values are in different scales. So, we need to standardize the dataset before building the ANN model because it calculates the weights from Input to Hidden layers to Output.

Here, we use the `StandardScaler()` method to normalize the data. We must be cautious here to fit and transform on the train data set and only transform on the test data set. Because, mean, standard deviation is supposed to be calculated only on the train dataset (*should not consider the hold out set*). In fact, test data set should not been seen at all by the model until we make the predictions.

### Grid Search to find out the optimal hyper parameters for ANN training set

```
param_grid = {
    'hidden_layer_sizes': [100,200], # 50, 200
    'max_iter': [1500,2000,2500],
    'solver': ['adam'],
    'tol': [0.0001],
    'activation':['relu'],
    'verbose':['True'],
}
```

**hidden\_layer\_sizes** – creating the 2 hidden layers with 100, 200 neurons in each. In general, size of the neurons is the square values of the independent variables. Here, we have 9 variables, so we will run with little above 100 neurons.

**max\_iter** - Maximum iterations the model can run for updating the synoptic weights. After starting with 1000, we end up with 1500 range.

**Solver** – tried with stochastic gradient descent(sgd) and adam, gridsearch returns out with adam is the best solver we can use here. Thus, fixed with it.

Best Parameters are

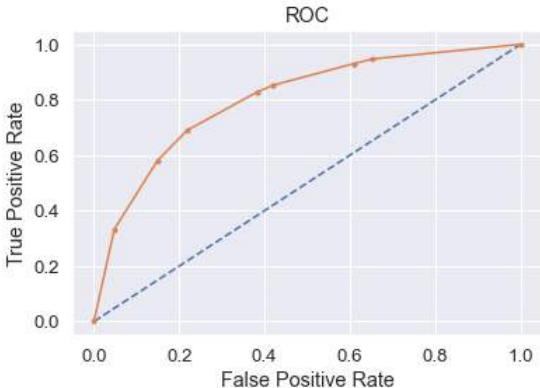
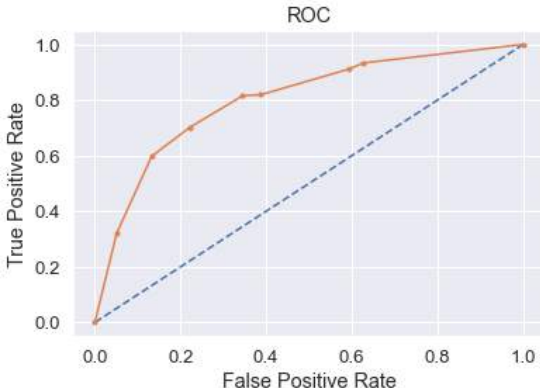
```
{'activation': 'relu', 'hidden_layer_sizes': 100, 'max_iter': 1500, 'solver': 'adam',  
 'tol': 0.0001, 'verbose': 'True'}
```

Predicting on Training and Test dataset

```
n_ytrain_predict = nn_best_grid.predict(nX_train)  
n_ytest_predict = nn_best_grid.predict(nX_test)
```

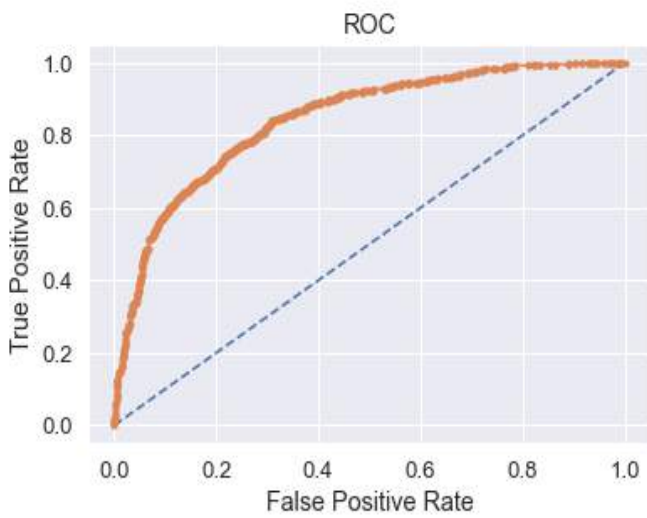
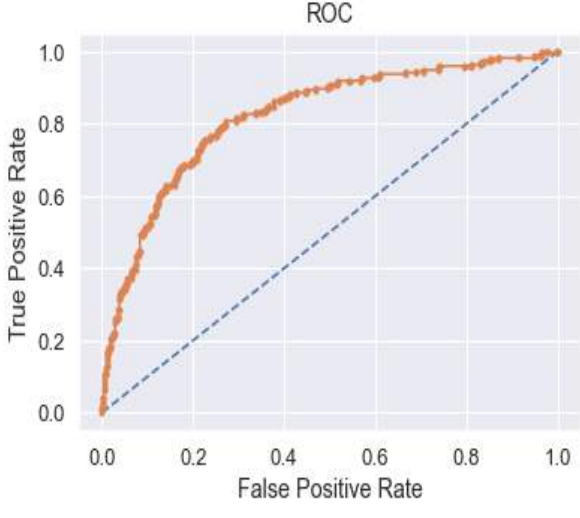
**2.3 Performance Metrics:** Check the performance of Predictions on Train and Test sets using Accuracy (1 pts), Confusion Matrix (2 pts), Plot ROC curve and get ROC\_AUC score for each model (2 pts), Make classification reports for each model. Write inferences on each model (2 pts). Calculate Train 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.

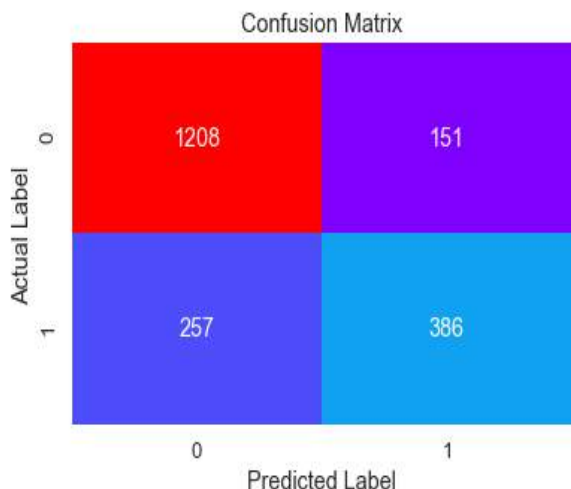
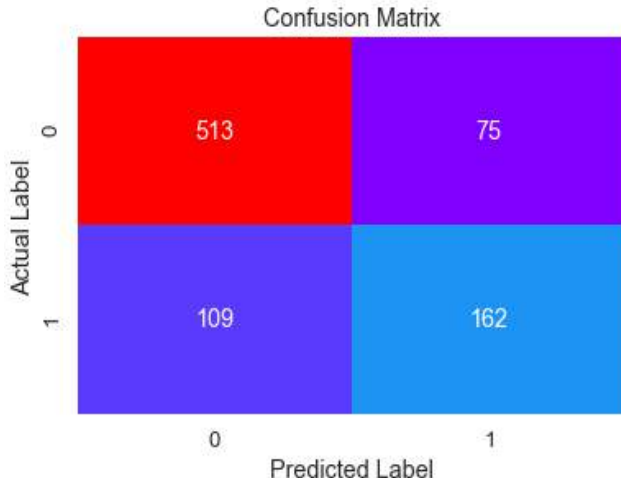
## Decision Tree Predictions

CART Train Set	CART Test Set
<b>Prediction</b> <code>ct_ytrain_predict = t_best_grid.predict(cX_train)</code>	<b>Prediction</b> <code>ct_ytest_predict = ct_best_grid.predict(cX_test)</code>
<b>Data shape</b> There are 2002 records	<b>Data shape(30% of the data)</b> There are 859 records
<b>Area Under the Curve</b> <code>cart_train_auc = roc_auc_score(ctrain_labels, ct_probs)</code> AUC: 0.802	<b>Area Under the Curve</b> <code>cart_test_auc = roc_auc_score(ctest_labels, ct_probs)</code> AUC: 0.802
<b>ROC</b> 	<b>ROC</b> 
<b>Data Accuracy</b> <code>cart_train_acc=ct_best_grid.score(cX_train,ctrain_labels)</code> 0.7642357642357642	<b>Data Accuracy</b> <code>cart_test_acc=ct_best_grid.score(cX_test,ctest_labels)</code> 0.7823050058207218

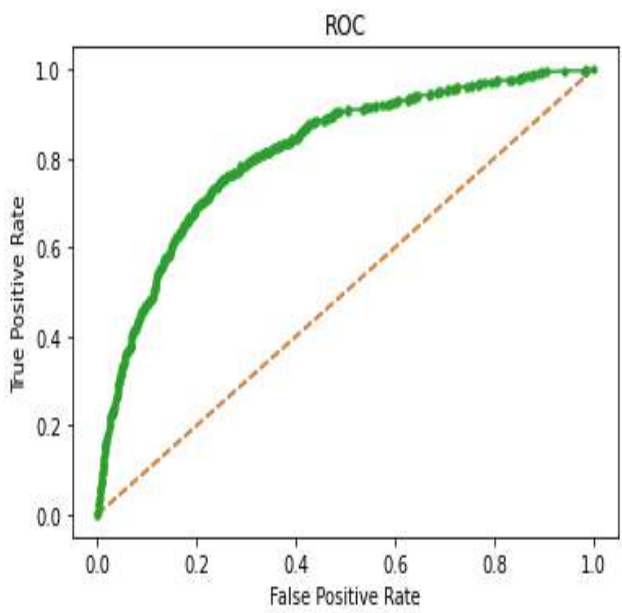
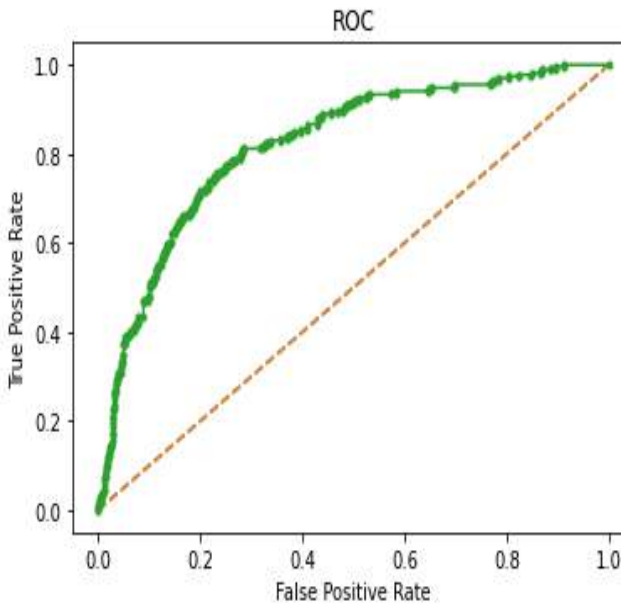
CART Train Set	CART Test Set																																																												
<div>Confusion Matrix</div> <div><div>Confusion Matrix</div><table><tr><td>0</td><td>1157</td><td>202</td></tr><tr><td>1</td><td>270</td><td>373</td></tr><tr><td></td><td>0</td><td>1</td></tr></table><div>Predicted Label</div></div>	0	1157	202	1	270	373		0	1	<div>Confusion Matrix</div> <div><div>Confusion Matrix</div><table><tr><td>0</td><td>510</td><td>78</td></tr><tr><td>1</td><td>109</td><td>162</td></tr><tr><td></td><td>0</td><td>1</td></tr></table><div>Predicted Label</div></div>	0	510	78	1	109	162		0	1																																										
0	1157	202																																																											
1	270	373																																																											
	0	1																																																											
0	510	78																																																											
1	109	162																																																											
	0	1																																																											
<div>Classification Report</div> <table><tr><td></td><td>precision</td><td>recall</td><td>f1-score</td><td>support</td></tr><tr><td>0</td><td>0.81</td><td>0.85</td><td>0.83</td><td>1359</td></tr><tr><td>1</td><td>0.65</td><td>0.58</td><td>0.61</td><td>643</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.76</td><td>2002</td></tr><tr><td>macro avg</td><td>0.73</td><td>0.72</td><td>0.72</td><td>2002</td></tr><tr><td>weighted avg</td><td>0.76</td><td>0.76</td><td>0.76</td><td>2002</td></tr></table>		precision	recall	f1-score	support	0	0.81	0.85	0.83	1359	1	0.65	0.58	0.61	643	accuracy			0.76	2002	macro avg	0.73	0.72	0.72	2002	weighted avg	0.76	0.76	0.76	2002	<div>Classification Report</div> <table><tr><td></td><td>precision</td><td>recall</td><td>f1-score</td><td>support</td></tr><tr><td>0</td><td>0.82</td><td>0.87</td><td>0.85</td><td>588</td></tr><tr><td>1</td><td>0.68</td><td>0.60</td><td>0.63</td><td>271</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.78</td><td>859</td></tr><tr><td>macro avg</td><td>0.75</td><td>0.73</td><td>0.74</td><td>859</td></tr><tr><td>weighted avg</td><td>0.78</td><td>0.78</td><td>0.78</td><td>859</td></tr></table>		precision	recall	f1-score	support	0	0.82	0.87	0.85	588	1	0.68	0.60	0.63	271	accuracy			0.78	859	macro avg	0.75	0.73	0.74	859	weighted avg	0.78	0.78	0.78	859
	precision	recall	f1-score	support																																																									
0	0.81	0.85	0.83	1359																																																									
1	0.65	0.58	0.61	643																																																									
accuracy			0.76	2002																																																									
macro avg	0.73	0.72	0.72	2002																																																									
weighted avg	0.76	0.76	0.76	2002																																																									
	precision	recall	f1-score	support																																																									
0	0.82	0.87	0.85	588																																																									
1	0.68	0.60	0.63	271																																																									
accuracy			0.78	859																																																									
macro avg	0.75	0.73	0.74	859																																																									
weighted avg	0.78	0.78	0.78	859																																																									
<div>Summary</div> <div>Train Data:</div> <div>AUC: 80%</div> <div>Accuracy: 76%</div> <div>Precision: 65%</div> <div>Recall: 58%</div> <div>f1-Score: 61%</div>	<div>Summary</div> <div>Test Data:</div> <div>AUC: 80%</div> <div>Accuracy: 78%</div> <div>Precision: 68%</div> <div>Recall: 60%</div> <div>f1-Score: 63%</div>																																																												

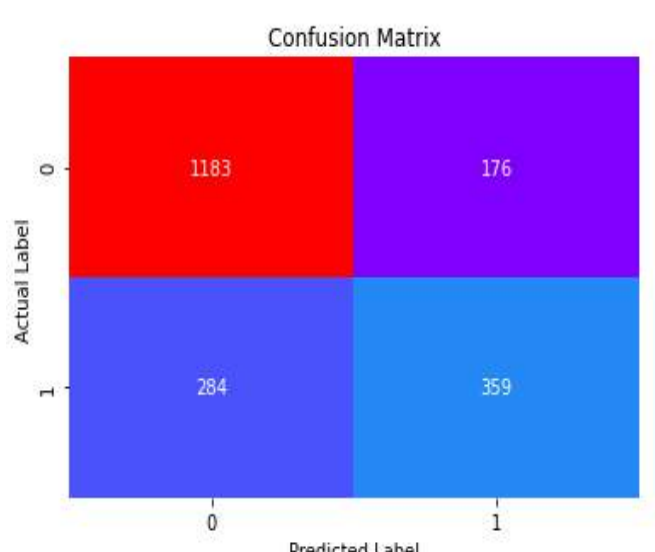
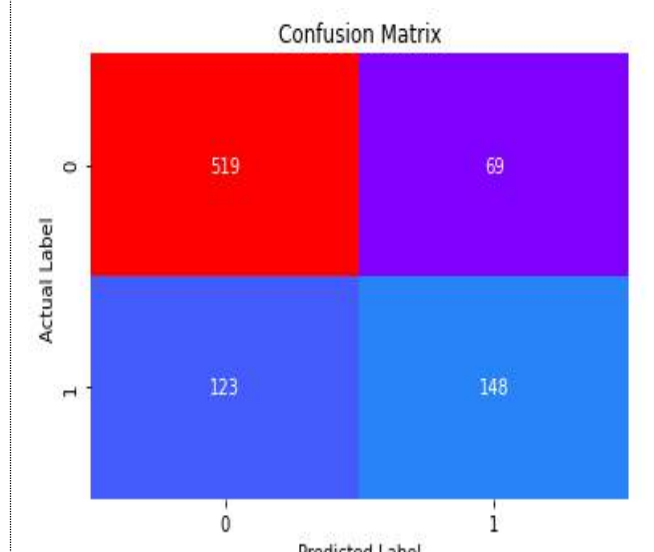
## Random Classifier Model Predictions

RF Train Set	RF Test Set
<b>Predictions</b> <code>ytrain_predict = best_grid.predict(X_train)</code>	<b>Predictions</b> <code>ytest_predict = best_grid.predict(X_test)</code>
<b>Data shape</b> There are 2002 records	<b>Data shape(30% of the data)</b> There are 859 records
<b>Area Under the Curve</b> <code>rf_train_auc = roc_auc_score(train_labels, rfprobs)</code> AUC: 0.844	<b>Area Under the Curve</b> <code>rf_test_auc = roc_auc_score(test_labels, rfprobs)</code> AUC: 0.822
<b>ROC</b> 	<b>ROC</b> 
<b>Data Accuracy</b> <code>rf_train_acc=best_grid.score(X_train,train_labels)</code> 0.7962037962037962	<b>Data Accuracy</b> <code>rf_test_acc=best_grid.score(X_test,test_labels)</code> 0.7857974388824214

RF Train Set	RF Test Set																																																												
<div>Confusion Matrix</div> <div><div>Confusion Matrix</div></div>	<div>Confusion Matrix</div> <div><div>Confusion Matrix</div></div>																																																												
<div>Classification Report</div> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.82</td><td>0.89</td><td>0.86</td><td>1359</td></tr><tr><td>1</td><td>0.72</td><td>0.60</td><td>0.65</td><td>643</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.80</td><td>2002</td></tr><tr><td>macro avg</td><td>0.77</td><td>0.74</td><td>0.75</td><td>2002</td></tr><tr><td>weighted avg</td><td>0.79</td><td>0.80</td><td>0.79</td><td>2002</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.82	0.89	0.86	1359	1	0.72	0.60	0.65	643	accuracy			0.80	2002	macro avg	0.77	0.74	0.75	2002	weighted avg	0.79	0.80	0.79	2002	<div>Classification Report</div> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.82</td><td>0.87</td><td>0.85</td><td>588</td></tr><tr><td>1</td><td>0.68</td><td>0.60</td><td>0.64</td><td>271</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.79</td><td>859</td></tr><tr><td>macro avg</td><td>0.75</td><td>0.74</td><td>0.74</td><td>859</td></tr><tr><td>weighted avg</td><td>0.78</td><td>0.79</td><td>0.78</td><td>859</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.82	0.87	0.85	588	1	0.68	0.60	0.64	271	accuracy			0.79	859	macro avg	0.75	0.74	0.74	859	weighted avg	0.78	0.79	0.78	859
	precision	recall	f1-score	support																																																									
0	0.82	0.89	0.86	1359																																																									
1	0.72	0.60	0.65	643																																																									
accuracy			0.80	2002																																																									
macro avg	0.77	0.74	0.75	2002																																																									
weighted avg	0.79	0.80	0.79	2002																																																									
	precision	recall	f1-score	support																																																									
0	0.82	0.87	0.85	588																																																									
1	0.68	0.60	0.64	271																																																									
accuracy			0.79	859																																																									
macro avg	0.75	0.74	0.74	859																																																									
weighted avg	0.78	0.79	0.78	859																																																									
<div>Summary</div> <div>Train Data: AUC: 84% Accuracy: 80% Precision: 72% Recall: 60% f1-Score: 65%</div>	<div>Summary</div> <div>Test Data: AUC: 82% Accuracy: 79% Precision: 68% Recall: 60% f1-Score: 64%</div>																																																												

## ANN Predictions

ANN Train Set	ANN Test Set
<b>Predictions</b> <code>n_ytrain_predict = nn_best_grid.predict(nX_train)</code>	<b>Predictions</b> <code>n_ytest_predict = nn_best_grid.predict(nX_test)</code>
<b>Data shape</b> There are 2002 records	<b>Data shape(30% of the data)</b> There are 859 records
<b>Area Under the Curve</b> <code>nn_train_auc = roc_auc_score(ntrain_labels, nnprobs)</code> AUC: 0.814	<b>Area Under the Curve</b> <code>nn_test_auc = roc_auc_score(ntest_labels, nnprobs)</code> AUC: 0.822
<b>ROC</b> 	<b>ROC</b> 
<b>Data Accuracy</b> <code>nn_train_acc=nn_best_grid.score(nX_train,ntrain_labels)</code> 0.7702297702297702	<b>Data Accuracy</b> <code>nn_test_acc=nn_best_grid.score(nX_test,ntest_labels)</code> 0.7764842840512224

ANN Train Set	ANN Test Set																																																												
<div><div>Confusion Matrix</div><div><div>Confusion Matrix</div></div></div>	<div><div>Confusion Matrix</div><div><div>Confusion Matrix</div></div></div>																																																												
<div><div>Classification Report</div><table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.81</td><td>0.87</td><td>0.84</td><td>1359</td></tr><tr><td>1</td><td>0.67</td><td>0.56</td><td>0.61</td><td>643</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.77</td><td>2002</td></tr><tr><td>macro avg</td><td>0.74</td><td>0.71</td><td>0.72</td><td>2002</td></tr><tr><td>weighted avg</td><td>0.76</td><td>0.77</td><td>0.76</td><td>2002</td></tr></tbody></table></div>		precision	recall	f1-score	support	0	0.81	0.87	0.84	1359	1	0.67	0.56	0.61	643	accuracy			0.77	2002	macro avg	0.74	0.71	0.72	2002	weighted avg	0.76	0.77	0.76	2002	<div><div>Classification Report</div><table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.81</td><td>0.88</td><td>0.84</td><td>588</td></tr><tr><td>1</td><td>0.68</td><td>0.55</td><td>0.61</td><td>271</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.78</td><td>859</td></tr><tr><td>macro avg</td><td>0.75</td><td>0.71</td><td>0.73</td><td>859</td></tr><tr><td>weighted avg</td><td>0.77</td><td>0.78</td><td>0.77</td><td>859</td></tr></tbody></table></div>		precision	recall	f1-score	support	0	0.81	0.88	0.84	588	1	0.68	0.55	0.61	271	accuracy			0.78	859	macro avg	0.75	0.71	0.73	859	weighted avg	0.77	0.78	0.77	859
	precision	recall	f1-score	support																																																									
0	0.81	0.87	0.84	1359																																																									
1	0.67	0.56	0.61	643																																																									
accuracy			0.77	2002																																																									
macro avg	0.74	0.71	0.72	2002																																																									
weighted avg	0.76	0.77	0.76	2002																																																									
	precision	recall	f1-score	support																																																									
0	0.81	0.88	0.84	588																																																									
1	0.68	0.55	0.61	271																																																									
accuracy			0.78	859																																																									
macro avg	0.75	0.71	0.73	859																																																									
weighted avg	0.77	0.78	0.77	859																																																									
<div><div>Summary</div><div><div>Train Data:</div><div>AUC: 81% Accuracy: 77% Precision: 67% Recall: 56% f1-Score: 61%</div></div></div>	<div><div>Summary</div><div><div>Test Data:</div><div>AUC: 82% Accuracy: 78% Precision: 68% Recall: 55% f1-Score: 61%</div></div></div>																																																												



**Overfitting and Underfitting:** In all the three models, we could see our models are neither overfitted nor underfitted. Because, our training set and testing set results are almost equal.

**Precision:** Ratio between TP and all positives. The number of customers that the model correctly identifies the claim has been made out of all the customers claiming it.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

**Recall:** Model correctly identifying the TPs. For all the customers, who actually have claimed, recall tells us how many the model identified the customers claiming correctly.

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

**F1 score:** Understanding Accuracy made us realize, we need a tradeoff between precision and recall. F1 Score is the weighted average of Precision and Recall. In the context of our business problem, the model must predict the customers claiming correctly.

We would then aim for high '**recall**' than precision.

$$\text{F1 Score} = 2 * (\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

**Random Forest Model has high F1 score compared to the Decision Tree and Neural Network models.**

**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.**

ROC Curve for DT, RF and ANN

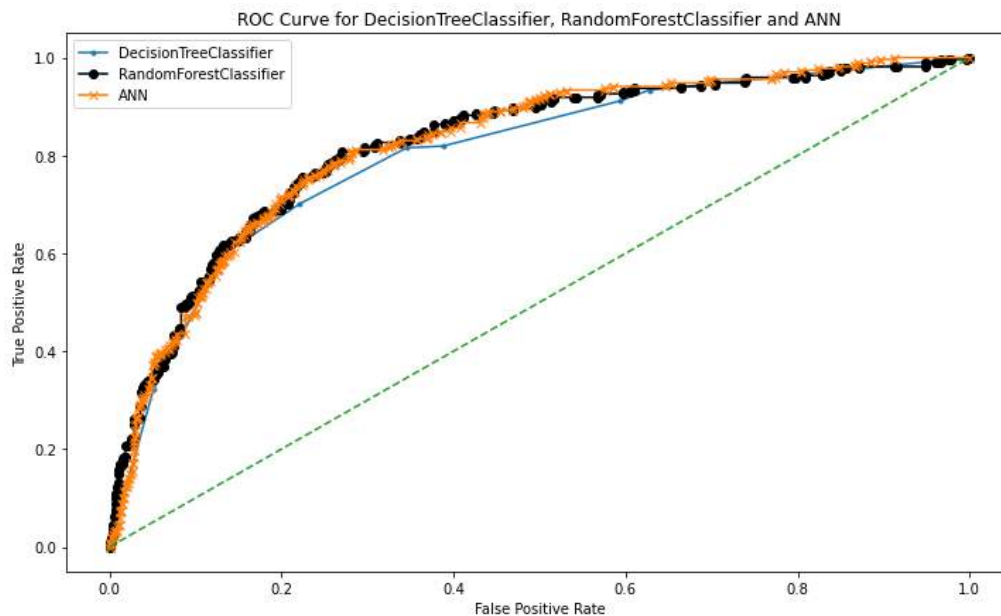


Figure 54.ROC Curve for all the Models

	CART Train	CART Test	CART Test with Threshold >0.3	Random Forest Train	Random Forest Test	Random Forest Test with Threshold >0.3	Neural Network Train	Neural Network Test
Accuracy	0.76	0.78	0.91	0.80	0.79	0.86	0.77	0.78
AUC	0.80	0.80	0.74	0.84	0.82	0.76	0.81	0.82
Recall	0.58	0.60	0.70	0.60	0.60	0.76	0.56	0.55
Precision	0.65	0.68	0.59	0.72	0.68	0.59	0.67	0.68
F1 Score	0.61	0.63	0.64	0.65	0.64	0.66	0.61	0.61

Table 18.Comparison of CART, RF and ANN Models

For our business problem, we aim for high recall. i.e. How many of the actual customers claiming correctly we were able to predict correctly with our model.

- Precision/Recall classifier values for the test data's are almost same for the CART and RF.
- F1 score of RF is slightly higher compared to the CART.
- Accuracy for ANN is 79%. But our precision/recall is low here. Ideally, for our problem, we would completely avoid any situations where the customers are claimed, but our model predicts them as not claimed.
- With ANN, we cannot discover which variable is most important for our business.
- With CART, only Agency Code/Sales are the most important and Products/Commission is little important for our business. Besides, all the other features are insignificant. On the other hand, RF builds the model by considering almost all the features.
- Run the CART & RF model with the threshold value above  $>0.3$ , noticed the good hike in the 'recall' score about 6% in Random Forest Model.
- Random Forest builds multiple Decision Trees and merges them together to get more accurate and stable prediction. Thus, we conclude that **'Random Forest' is the best model for our business problem.**

**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.**

### Insights & Recommendations

- Setup marketing campaigns to JZI agency as their sales is too low or Customers are not interested to use JZI agency for Airlines, may need to tie up with other agency.
- Give message to C2B agency to setup some manual process to verify the claim due to their maximum number of claims made using this agency code.
- Get feedbacks from the customers who used the 'travel agency' which is comparatively low with Airlines and up sell the products & highlight the benefits of the products & services.

- In fact, all the customers who uses Offline channel have used Travel Agency and none of their destination is to Europe.
- Create a multi tour package, enhance some other offers and highlight to the people pick Europe as the destination since the count is low here.
- Customers uses Offline channel are conserved in choosing the plans, either it is Gold or Customized plan. Moreover, their sales in procuring the policies are within 10,000/- rupees.
- Age above ~60 not interested to choose Europe as the destination place. Understand the elderly people experiences and their preferences; inform the agency to promote the products accordingly.
- In general, claiming process is time consuming and cost expensive, improve the process by combining the teams, number of procedures as much as you can.
- Based on the model, 60% of the claims are correctly predicted. Also, the accuracy for the test set is ~80% and the precision/recall classifiers are ~60% which is quite balanced with the records.

**THE END!**