2022

Data Mining Project

Akashatra Sharma

Contents

Table OF Figures

Executive Summary

There are basically two types of Dataset provided which gives us a lot of information. The first data set was named as “Bank Marketing Part 1” and it consists of the Activities of the customers for purchases during the past few months . Next data we were provided was named as “Insurance Part 2 Data - 1 ” which contains information on various tour insurance firm and their policies claim status. In both of the datasets, we performed different analytical techniques and build different models in order to get better understanding and give business implications regarding each case study of data set.

Introduction

The purpose of this assignment is to explore the data sets. For that, we’ll do different analytical & statistical operations in order to get the most of out the data.

Starting with the data sets, we had gone through the both the data sets and the briefing of the data sets are as follows :

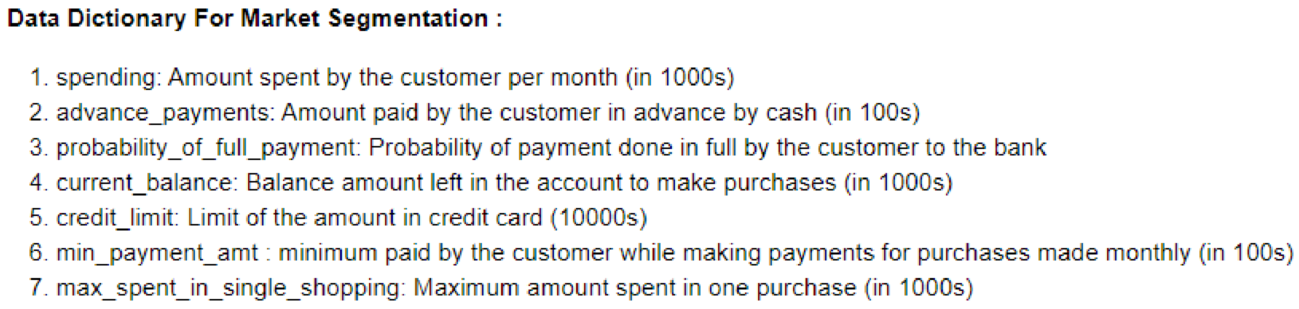
- First Data set that was ‘Bank Marketing Part 1’ consists of spending amount and others features of 210 Customers/Users having different probabilities of making full payment by the customers to the bank on the basis of other features.
- Second data set that was ‘Insurance Part 2 Data -1’ consists variety of information of 3000 customers who had made tour insurance policies along with the information of status of policy claimed.

Data Dictionary

The data dictionary is mainly for the understanding of meaning of columns provided in the data set .

1. **Bank Marketing Part 1**

   It is as follows :



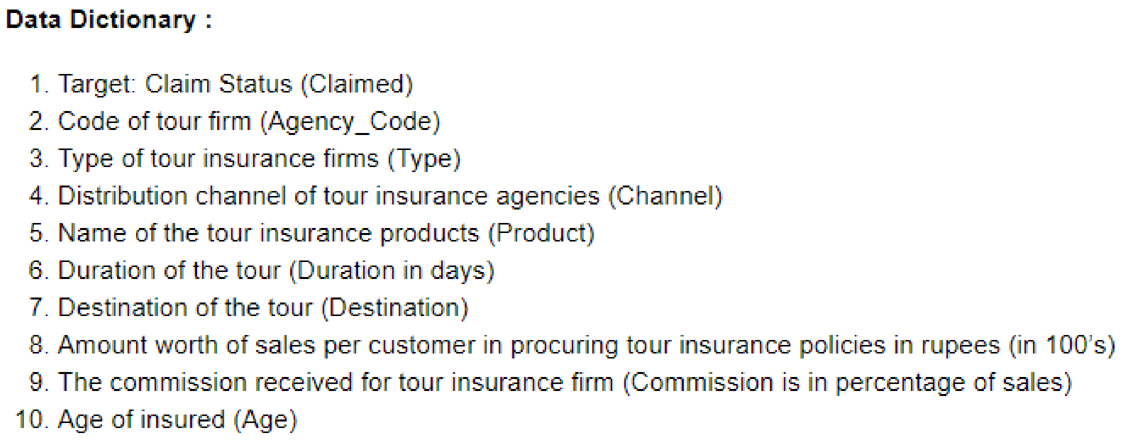
Data Dictionary For Market Segmentation :

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

Figure 1 : Data Dictionary For Market Segmentation

2. **Insurance Part 2 Data -1**

It is as follows :



Data Dictionary :

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)

Figure 2 : Data Dictionary For Insurance Data

Data Description

Description of both data sets are as follows :

1. **Bank Marketing Part 1**

-	spending	:	Continuous Data from	10.590	to	21.1800
-	advance_payments	:	Continuous Data from	12.410	to	17.2500
-	probability_of_full_payment	:	Continuous Data from	0.8081	to	0.9183
-	current_balance	:	Continuous Data from	4.8990	to	6.6750
-	credit_limit	:	Continuous Data from	2.6300	to	4.0330
-	min_payment_amt	:	Continuous Data from	0.7651	to	8.4560

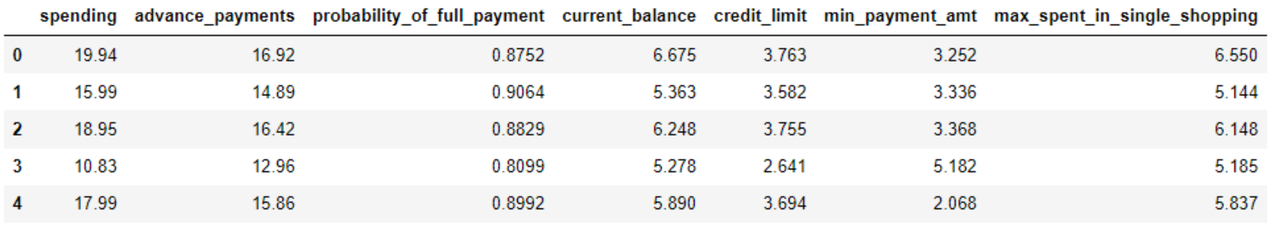
2. **Insurance Part 2 Data -1**

-	Age	:	Continuous Data from	8.0	to	84
-	Agency_Code	:	Categorical Data from	C2B	to	JZI
-	Type	:	Categorical Data from	Airlines	to	Airlines
-	Claimed	:	Categorical Data from	No	to	No
-	Commision	:	Continuous Data from	0.70	to	11.55
-	Channel	:	Categorical Data from	Online	to	Online
-	Duration	:	Continuous Data from	7	to	15
-	Sales	:	Continuous Data from	2.51	to	33
-	Product_Name	:	Categorical Data from	Customised Plan	to	Bronze Plan
-	Destination	:	Categorical Data from	ASIA	to	ASIA

Datasets

1. Bank Marketing Part 1

Here were the first five observation of this data set :

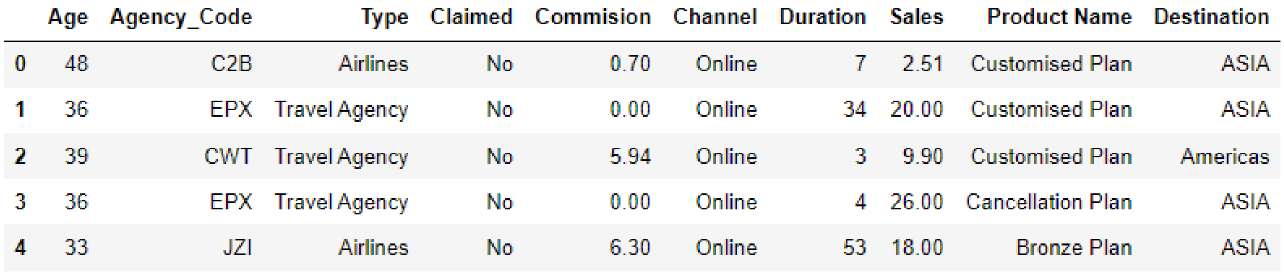


	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148
3	10.83	12.96	0.8099	5.278	2.641	5.182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

Figure 3 : First 5 Observations (Bank Marketing Part - 1)

2. Insurance Part 2 Data -1

Here were the first five observations of this data set :



	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

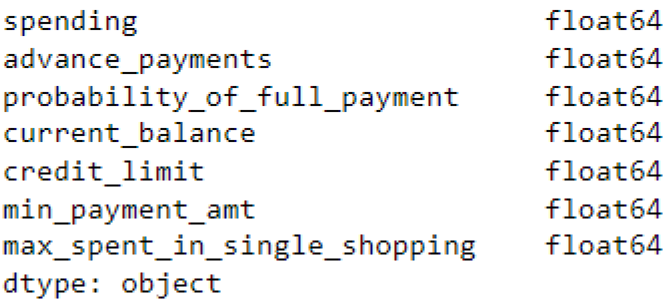
Figure 4 : First 5 Observations (Insurance Part 2 Data - 1)

Data Analysis

- Data Types

○ Bank Marketing Part 1

The data types of variables in the data set were :



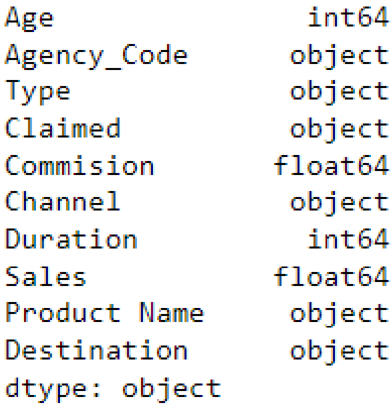
```
spending                        float64
advance_payments                float64
probability_of_full_payment     float64
current_balance                 float64
credit_limit                    float64
min_payment_amt                 float64
max_spent_in_single_shopping    float64
dtype: object
```

Figure 5 : Data Types (Bank Marketing Part 1)

We noted that there were all float values present in the dataset.

- **Insurance Part 2 Data -1**

  The data types of variables in the data set were :

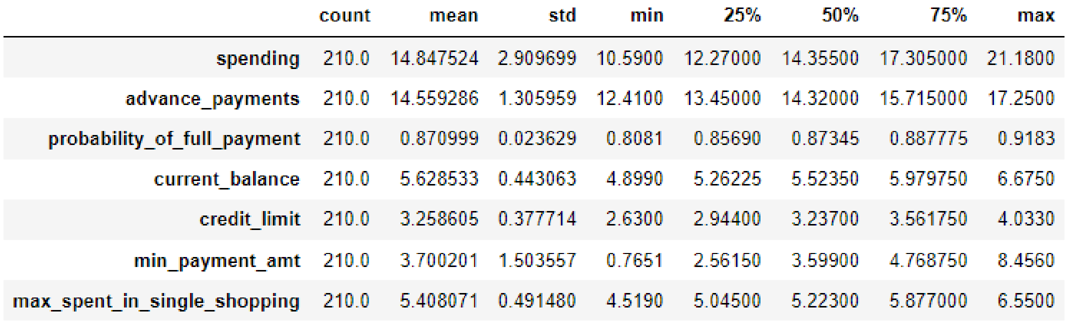


```
Age               int64
Agency_Code      object
Type             object
Claimed          object
Commision       float64
Channel          object
Duration          int64
Sales           float64
Product Name     object
Destination      object
dtype: object
```

Figure 6 : Data Types (Insurance Part 2 Data -1)

- Descriptive Statistics

1. Bank Marketing Part 1



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

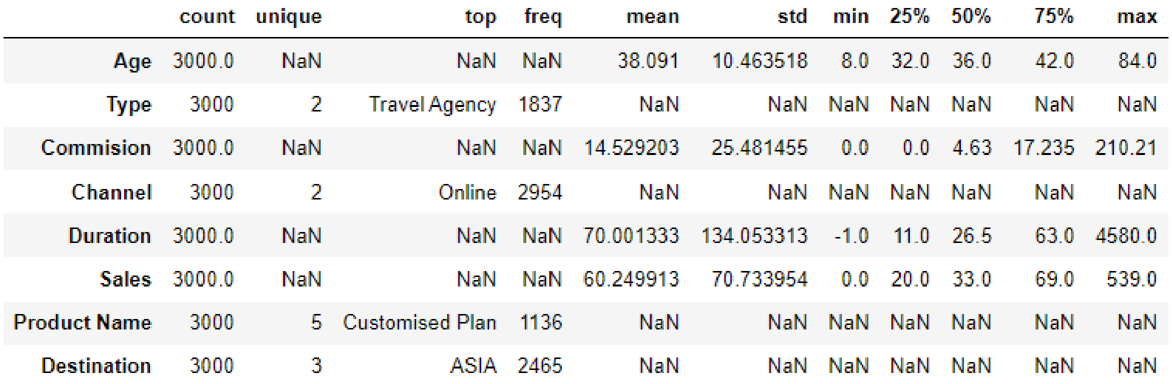
Figure 7 : Descriptive Stats (Bank Marketing)

Interpretations :

- We inferred that mean and median values for all the columns were very close to each other indicating that there were very less skewness between them.

- But in order for proper execution of clustering technique, scaling will be done in order to standardize the values which will eventually help in clustering.

2. **Insurance Part 2 Data -1**



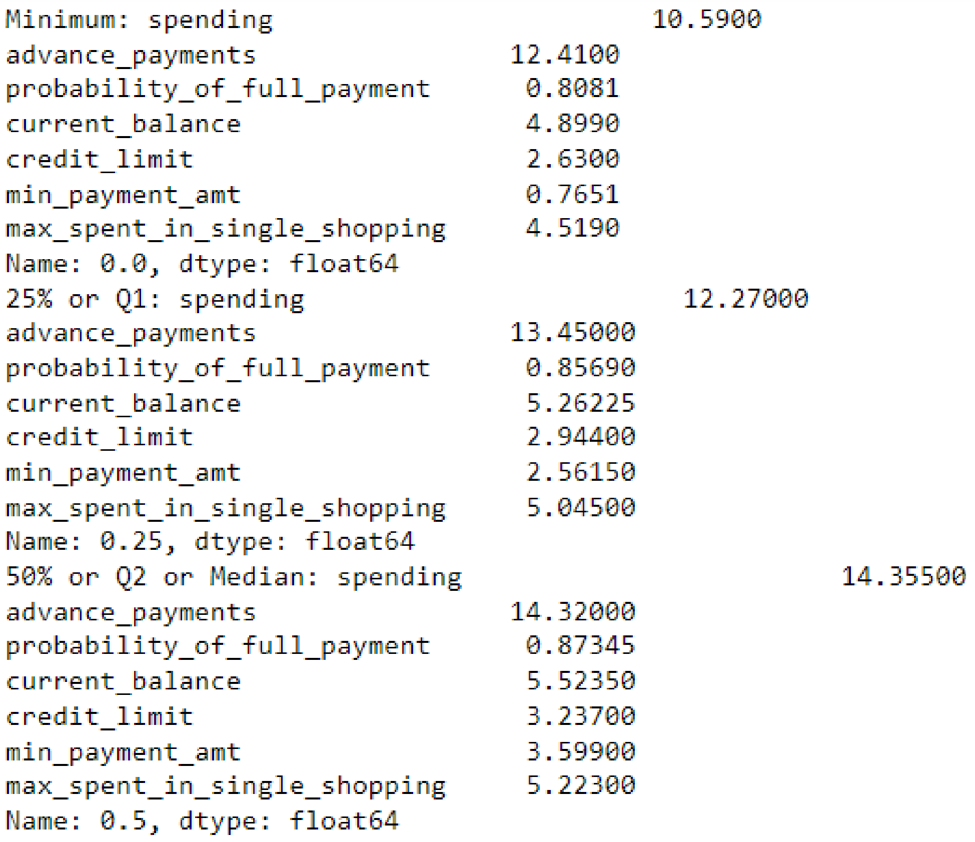
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	3000.0	NaN	NaN	NaN	38.091	10.463518	8.0	32.0	36.0	42.0	84.0
Type	3000	2	Travel Agency	1837	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Commision	3000.0	NaN	NaN	NaN	14.529203	25.481455	0.0	0.0	4.63	17.235	210.21
Channel	3000	2	Online	2954	NaN	NaN	NaN	NaN	NaN	NaN	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

Figure 8 : Descriptive Stats (Insurance Data)

- ## Five Point Summary

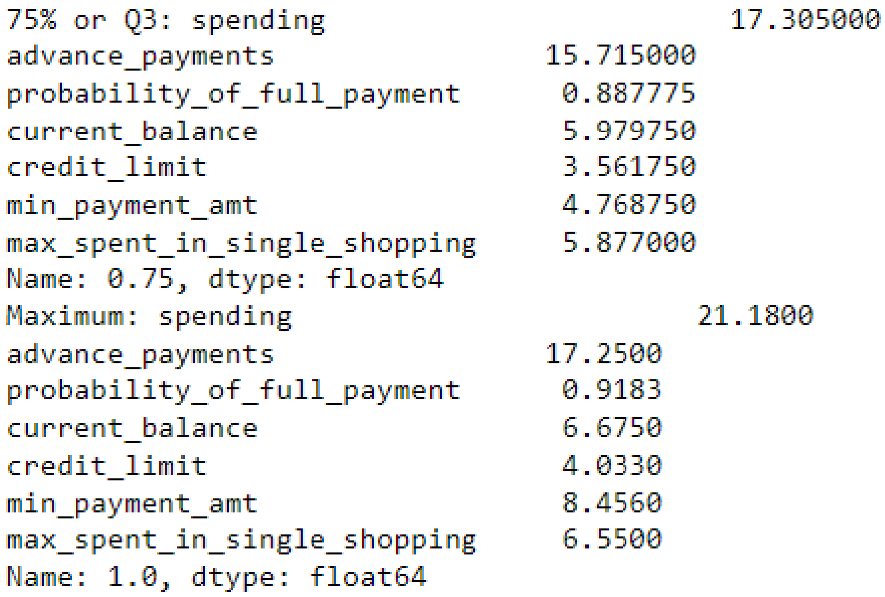
1. Bank Marketing Part 1

The five point summary is shown below :



```
Minimum: spending                            10.5900
advance_payments                    12.4100
probability_of_full_payment          0.8081
current_balance                      4.8990
credit_limit                         2.6300
min_payment_amt                      0.7651
max_spent_in_single_shopping         4.5190
Name: 0.0, dtype: float64
25% or Q1: spending                            12.27000
advance_payments                    13.45000
probability_of_full_payment          0.85690
current_balance                      5.26225
credit_limit                         2.94400
min_payment_amt                      2.56150
max_spent_in_single_shopping         5.04500
Name: 0.25, dtype: float64
50% or Q2 or Median: spending                            14.35500
advance_payments                    14.32000
probability_of_full_payment          0.87345
current_balance                      5.52350
credit_limit                         3.23700
min_payment_amt                      3.59900
max_spent_in_single_shopping         5.22300
Name: 0.5, dtype: float64
```

Figure 9 : Five Point Summary - 1

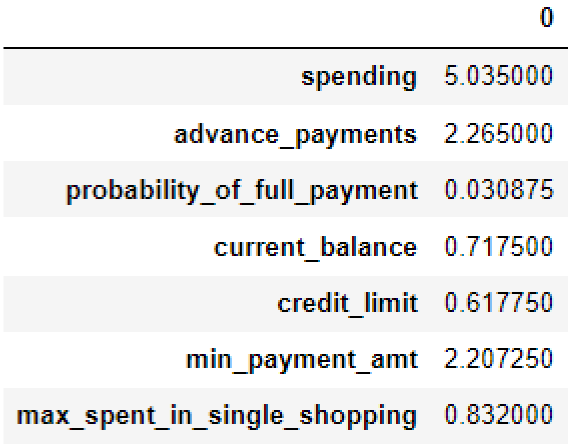


```
75% or Q3: spending                        17.305000
advance_payments                 15.715000
probability_of_full_payment       0.887775
current_balance                   5.979750
credit_limit                      3.561750
min_payment_amt                   4.768750
max_spent_in_single_shopping      5.877000
Name: 0.75, dtype: float64
Maximum: spending                      21.1800
advance_payments                 17.2500
probability_of_full_payment       0.9183
current_balance                   6.6750
credit_limit                      4.0330
min_payment_amt                   8.4560
max_spent_in_single_shopping      6.5500
Name: 1.0, dtype: float64
```

Figure 10 : Five Point Summary – 2

- ## IQR

1. **Bank Marketing Part 1**



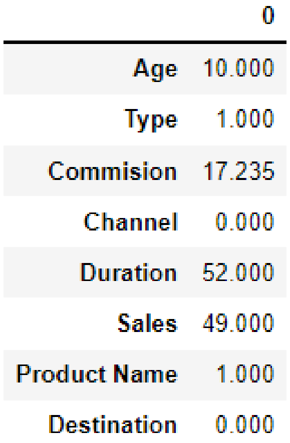
	0
spending	5.035000
advance_payments	2.265000
probability_of_full_payment	0.030875
current_balance	0.717500
credit_limit	0.617750
min_payment_amt	2.207250
max_spent_in_single_shopping	0.832000

Figure 11 : IQR (Bank Marketing Part 1)

Interpretations :

- We inferred that spending column had the highest IQR value meaning the range between quantile 1 & quantile 3 was very high as compared to others in the dataset.

2. **Insurance Part 2 Data - 1**

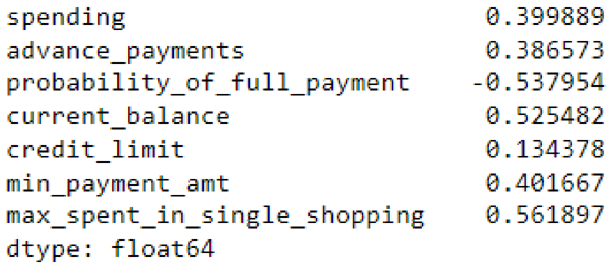


	0
Age	10.000
Type	1.000
Commision	17.235
Channel	0.000
Duration	52.000
Sales	49.000
Product Name	1.000
Destination	0.000

Figure 12 : IQR (Insurance Part 2 Data -1)

- Skewness

1. Bank Marketing Part 1



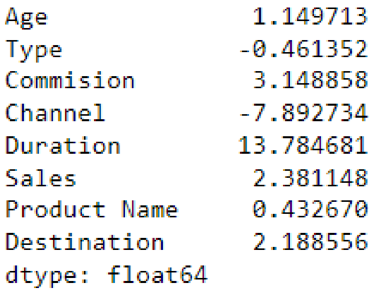
```
spending                            0.399889
advance_payments                    0.386573
probability_of_full_payment        -0.537954
current_balance                     0.525482
credit_limit                        0.134378
min_payment_amt                     0.401667
max_spent_in_single_shopping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

Figure 13 : Skewness (Bank Marketing Part 1)

Interpretations :

- We inferred that current balance and maximum spent in single shopping had maximum skewness among all.
- Among everyone, only probability of full payment had a value of -0.537 which indicated that it was negatively(left) skewed.
- Except Probability of full payment, rest all were positively (right) skewed.

2. Insurance Part 2 Data - 1



```
Age              1.149713
Type            -0.461352
Commision        3.148858
Channel         -7.892734
Duration        13.784681
Sales            2.381148
Product Name     0.432670
Destination      2.188556
dtype: float64
```

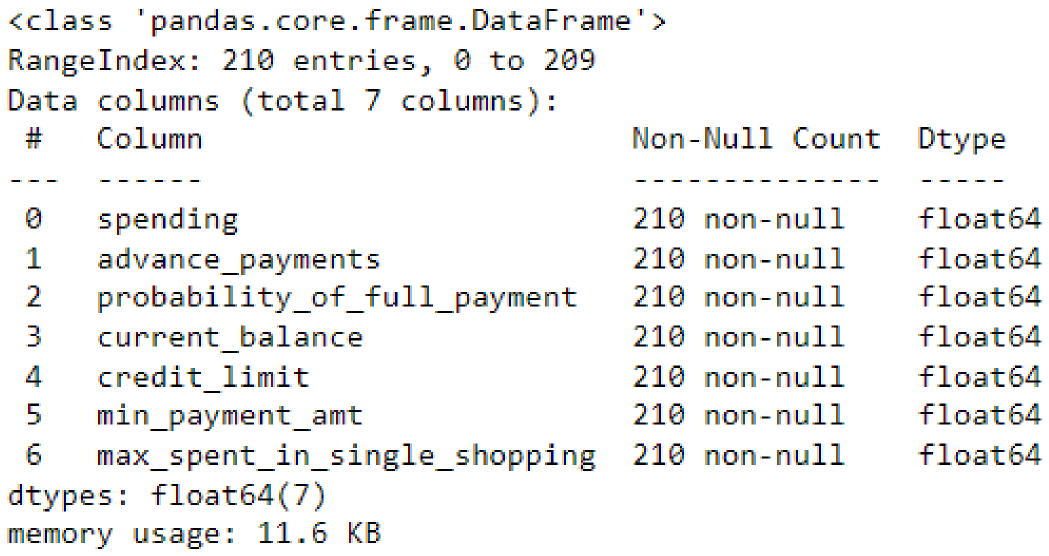
Figure 14 : Skewness (Insurance Part 2 Data - 1)

Interpretations :

- We inferred that Duration had maximum skewness among all.
- Among everyone, Type & Channel had skewness value of -0.4613 & -7.8927 which indicated that they were negatively(left) skewed.
- Except Channel and Type, rest all were positively (right) skewed.

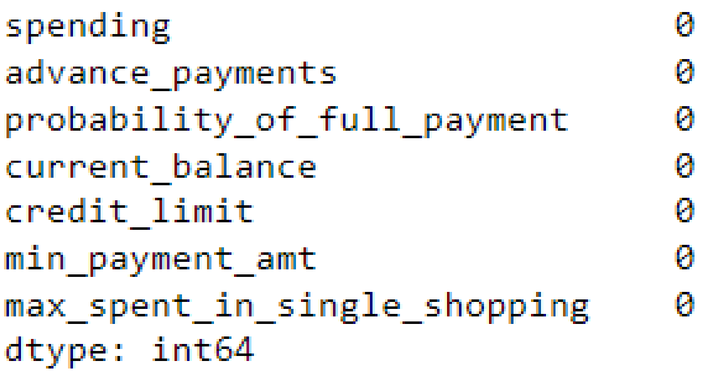
Checking For Null Values

- **Bank Marketing Part 1**



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   spending                      210 non-null    float64
 1   advance_payments              210 non-null    float64
 2   probability_of_full_payment   210 non-null    float64
 3   current_balance               210 non-null    float64
 4   credit_limit                  210 non-null    float64
 5   min_payment_amt               210 non-null    float64
 6   max_spent_in_single_shopping  210 non-null    float64
dtypes: float64(7)
memory usage: 11.6 KB
```

Figure 15 : Null Values Check (Bank Marketing) - 1



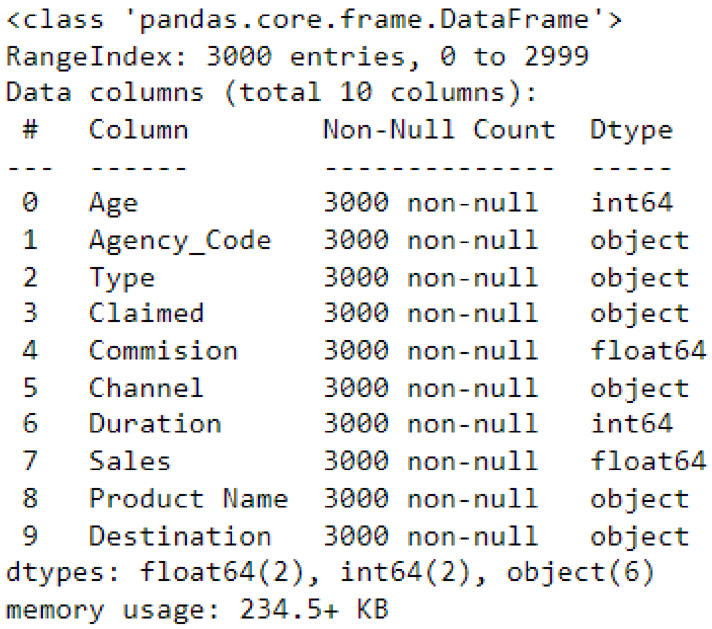
```
spending                        0
advance_payments                0
probability_of_full_payment     0
current_balance                 0
credit_limit                    0
min_payment_amt                 0
max_spent_in_single_shopping    0
dtype: int64
```

Figure 16 : Null Values Check (Bank Marketing) – 2

Interpretations :

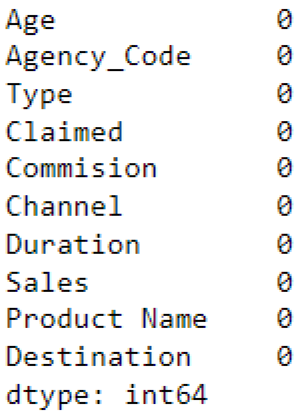
- Hence, it confirms that no null values were present in the dataset.
- Also, we noted that the shape/ dimensions of data set is (210,7) which means that there were 210 entries and 7 columns in the data set.

- **Insurance Part 2 Data -1**



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

Figure 17 : Null Values Check (Insurance Data) - 1



```
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 18 : Null Values Check (Insurance Data) – 2

Interpretations :

- Hence, it also confirms that no null values were present in the dataset.
- Also, we noted that the shape/ dimensions of data set is (3000, 10) which means that there were 3000 entries and 10 columns in the data set.

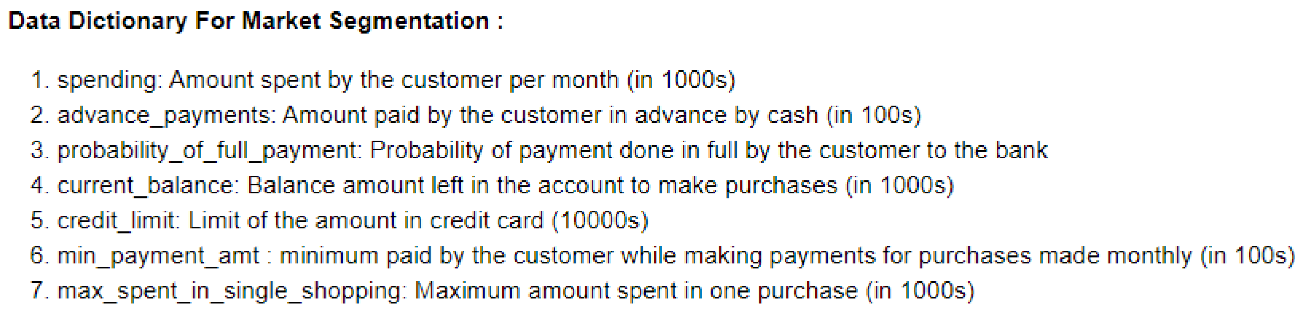
Problem 1 – Clustering

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

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

At first we loaded the Data Dictionary for understanding of column name for data set “Bank Marketing Part 1”.

The Data Dictionary is as follows :



Data Dictionary For Market Segmentation :

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

Figure 19 : Data Dictionary (Problem 1)

EDA

Exploratory Data Analysis (EDA) is an approach to analyze the data using visual techniques. It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations. It is of various types such as univariate, bi-variate and multi-variate.
After getting a brief understanding of what is EDA, we did analyze the data and here it is what we had found :

Univariate Analysis

For Univariate analysis, we plotted a Distribution plot and a Boxplot for each column provided in the data set .

The Distribution plot was used for univariant set of observations and visualizes it through a histogram i.e. only one observation and hence we choose one particular column of the dataset.

And, Boxplot was used as a measure of how well the data is distributed in a data set. It divides the data set into three quartiles. This graph represents the minimum, maximum, median, first quartile and third quartile in the data and also shows us whether there are outliers or not.

Here the Distribution Plot and Boxplot for **Bank Marketing Part 1** data set :

Spending :

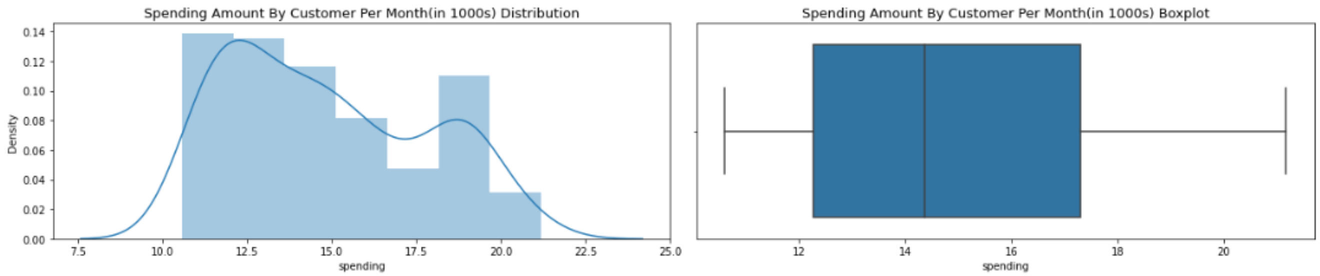


Figure 20 : Distplot & Boxplot (Spending)

The Distplot tells us that the graph is positively (right) skewed as its tail was at the right side of the distribution. Skewness value can be cross checked from the descriptive analysis section.

The 'Spending' variable showed no outliers as per the Boxplot which means all values were laying inside the range from minimum to maximum.

Advance Payment :

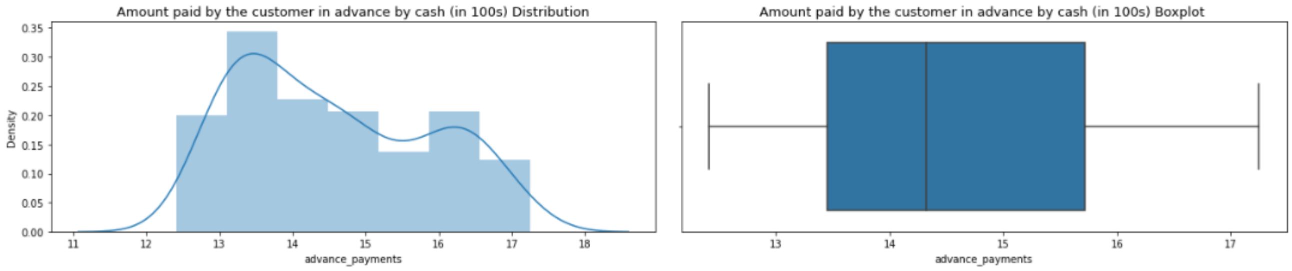


Figure 21 : Distplot & Boxplot (Advance Payment)

The Distplot tells us that the graph is positively (right) skewed as its tail was at the right side of the distribution. Skewness value can be cross checked from the descriptive analysis section.

The ‘Advance Payment’ variable showed no outliers as per the Boxplot which means all values were laying inside the range from minimum to maximum.

Probability Of Full Payment :

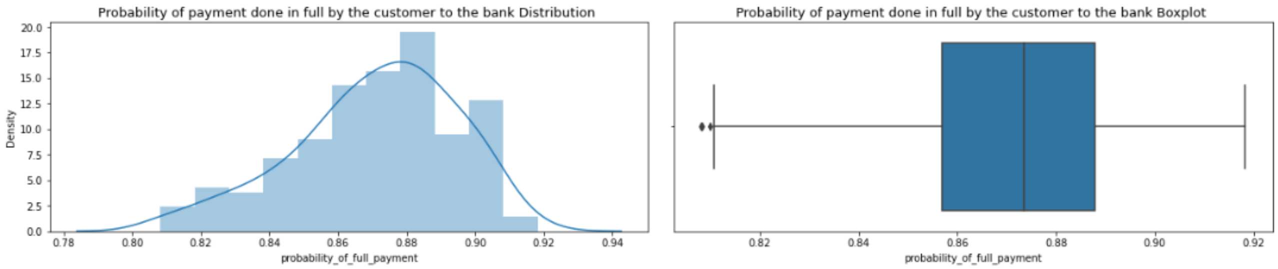


Figure 22 : Distplot & Boxplot (Probability Of Full Payment)

The Distplot tells us that the graph is negatively (left) skewed as its tail was at the left side of the distribution. Skewness value can be cross checked from the descriptive analysis section.

The ‘Probability Of Full Payment’ variable showed some outliers on the left side as per the Boxplot which means some values were laying on the extreme left side of the boxplot.

Current Balance :

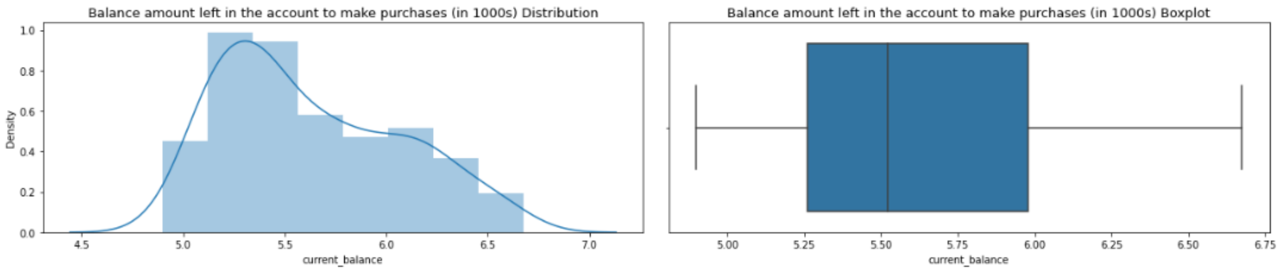


Figure 23 : Distplot & Boxplot (Current Balance)

The Distplot tells us that the graph is positively (right) skewed as its tail was at the left side of the distribution. Skewness value can be cross checked from the descriptive analysis section.

The ‘Current Balance’ variable showed no outliers as per the Boxplot which means all values were laying inside the range from minimum to maximum.

Credit Limit :

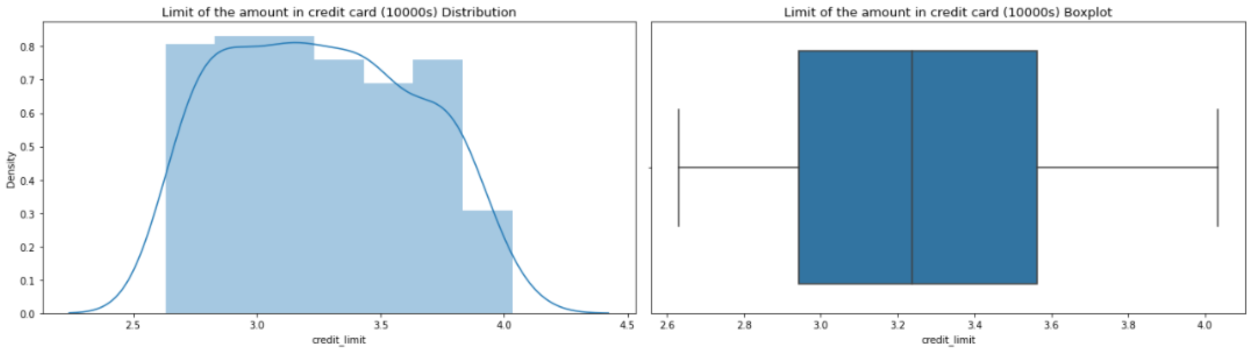


Figure 24 : Distplot & Boxplot (Credit Limit)

The Distplot tells us that the graph is positively (right) skewed as its tail was at the right side of the distribution. Skewness value can be cross checked from the descriptive analysis section.

The ‘Credit Limit’ variable showed no outliers as per the Boxplot which means all values were laying inside the range from minimum to maximum.

Minimum Payment Amount :

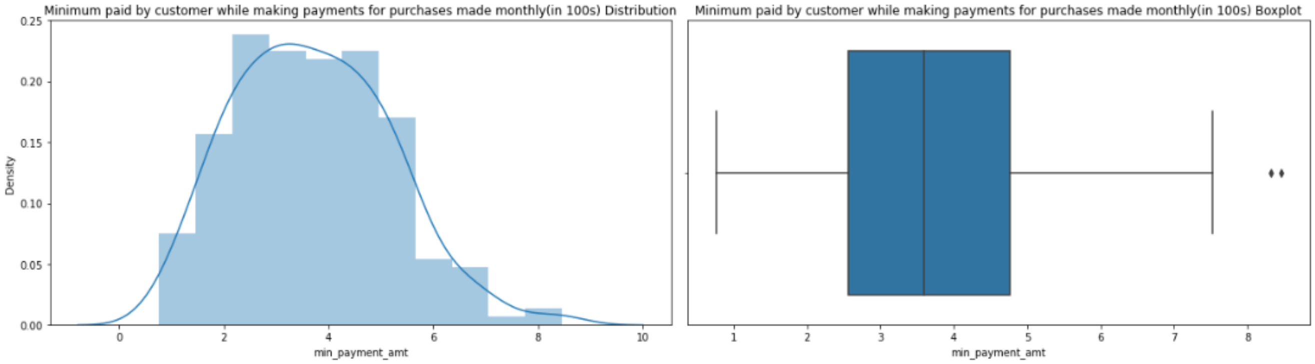


Figure 25 : Distplot & Boxplot (Minimum Payment Amount)

The ‘Minimum Payment Amount’ variable showed only 2 outlier as per the Boxplot which means only 2 values were laying outside the maximum range (as per the 5 Point summary).

As per the Distplot, we got to know that ‘Minimum Payment Amount’ was almost a normally distribution (neither left nor right skewed) but as we saw to the right side of the Distplot, we saw there were values present in high amount which breaks the normal distribution curve but it was close to normal distribution.

Hence, as per the skewness value, it was considered as positively (right) skewed as its tail was at the right side of the distribution.

Maximum Spent In a Single Shopping :

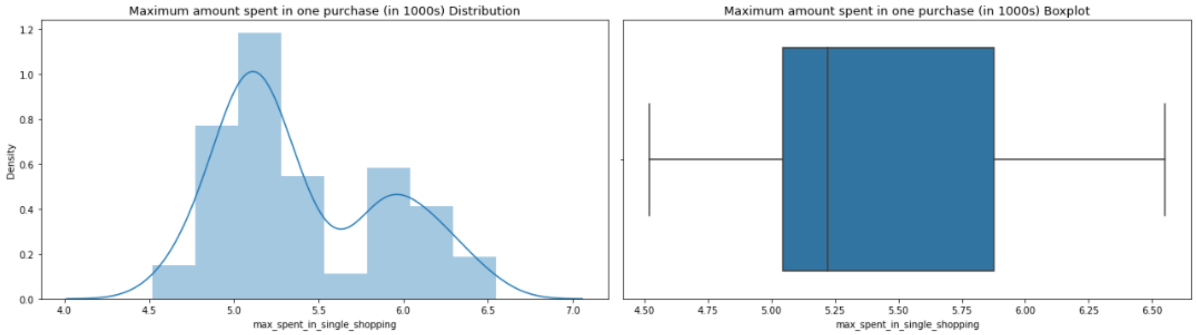


Figure 26 : Distplot & Boxplot (Maximum Spent in a Single Shopping)

The Distplot tells us that the graph is positively (right) skewed as its tail was at the right side of the distribution. Skewness value can be cross checked from the descriptive analysis section.

The 'Maximum Spent In a Spent Shopping' variable showed no outliers as per the Boxplot which means all values were laying inside the range from minimum to maximum.

Conclusion :

- After analysis all the distribution plots we inferred that, current balance and maximum spent in single shopping had maximum skewness among all.

- Except Probability of full payment, rest all were positively (right) skewed, because it's end tail was extending towards the left side which indicated the negative (left) skewness.

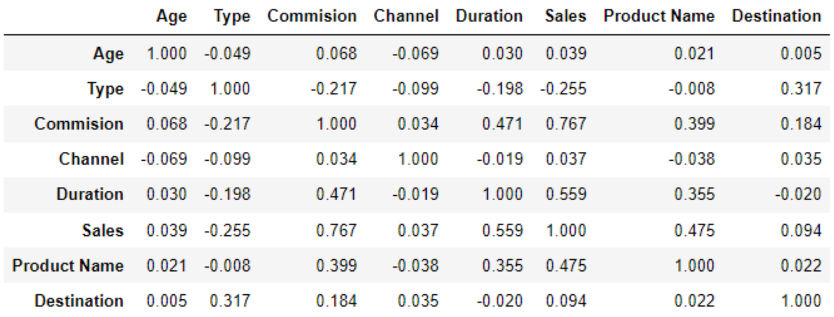
Bi-Variate Analysis :

Bivariate analysis means the analysis of bivariate data. It is one of the simplest forms of statistical analysis, used to find out if there is a relationship between two sets of values. It usually involves the variables X and Y.

There are many types of bivariate analysis such as scatter plot**,** regression analysis, correlation matrix analysis and much more.

- **Correlation Matrix :**

For this data, we did the correlation matrix and find out many insights from it. It is as follows :



	Age	Type	Commision	Channel	Duration	Sales	Product Name	Destination
Age	1.000	-0.049	0.068	-0.069	0.030	0.039	0.021	0.005
Type	-0.049	1.000	-0.217	-0.099	-0.198	-0.255	-0.008	0.317
Commision	0.068	-0.217	1.000	0.034	0.471	0.767	0.399	0.184
Channel	-0.069	-0.099	0.034	1.000	-0.019	0.037	-0.038	0.035
Duration	0.030	-0.198	0.471	-0.019	1.000	0.559	0.355	-0.020
Sales	0.039	-0.255	0.767	0.037	0.559	1.000	0.475	0.094
Product Name	0.021	-0.008	0.399	-0.038	0.355	0.475	1.000	0.022
Destination	0.005	0.317	0.184	0.035	-0.020	0.094	0.022	1.000

Figure 27 : Correlation Matrix (Bank Marketing Part 1)

Interpretations :

- A Correlation Matrix was created above using the 'Pearson' method.
- The matrix depicts the correlation between all the possible pairs of values in a table. It is a powerful tool to summarize a large dataset and to identify and visualize patterns in the given data.
- For better understanding, a graphical representation in the form of heatmap was also created with respect to Correlation Matrix.

- Heatmap :

A **Correlation Heatmap** is a rectangular representation of data and it repeats the same data description twice because the categories are repeated on both axis for computing analysis. Hence, the same result is obtained twice. A correlation heatmap that presents data only once without repetition that is categories are correlated only once is known as a **Triangle Correlation Heatmap**.

Since data is symmetric across the diagonal from left-top to right bottom the idea of obtaining a triangle correlation heatmap is to remove data above it so that it is depicted only once. The elements on the diagonal are the parts where categories of the same type correlate.

Triangle Correlation Heatmap for **Bank Marketing Part 1** data set is as follows :

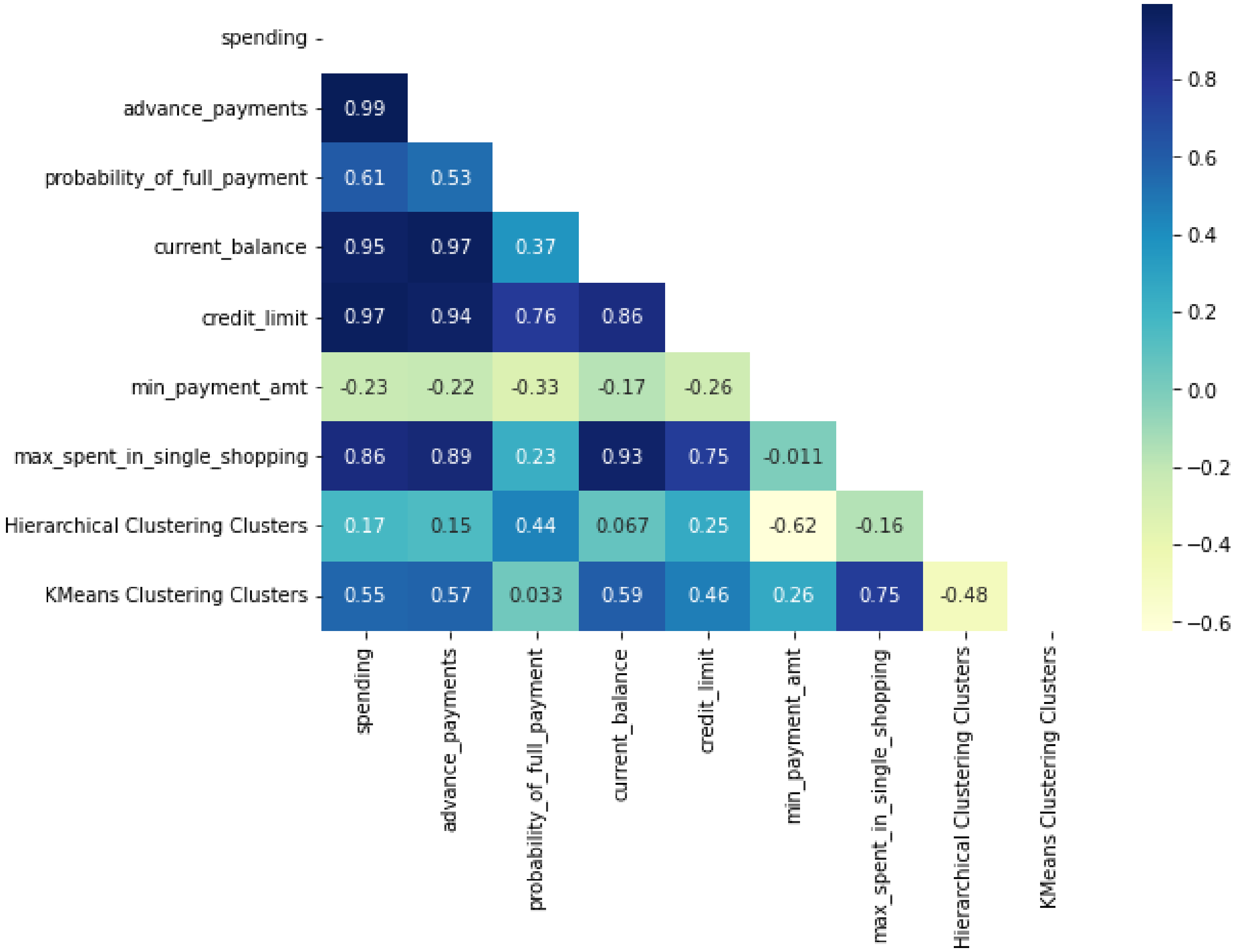


Figure 28 : Triangle Correlation Heatmap (Bank Marketing Part 1)

Interpretations :

- It showed that almost 99% of the customers did advance payments while spending money. While on the other hand, there is minimum correlation that is of -33% between minimum payment amount and probability of full payment which clearly indicated that, the customers who paid minimal amount for making purchases would unlikely had very less amount of probability of making payment in full as these customers preferred to pay minimum amount while conducting purchases.

- Adding on, it shows that there was 53% between probability of full payment and advance payments which means that mostly 50% of the customers are likely to pay the payments full in advance.

- Furthermore, we also get the insight that the correlation between spending and current balance is 95% which clearly indicated that the balance amount left in the customer's bank account per month was being used by him/her while spending at different places per month.

Multivariate Analysis :

Multivariate analysis (**MVA**) is a Statistical procedure for analysis of data involving more than one type of measurement or observation. It may also mean solving problems where more than one dependent variable is analyzed simultaneously with other variables . Multivariate analysis is one of the most useful methods to determine relationships and analyze patterns among large sets of data. It is particularly effective in minimizing bias if a structured study design is employed. However, the complexity of the technique makes it a less sought-out model for novice research enthusiasts. Therefore, although the process of designing the study and interpretation of results is a tedious one, the techniques stand out in finding the relationships in complex.

Pairplot :

In this we found out the **Pairplot** of the original dataset .

Pairplot function allows the users to create an axis grid via which each numerical variable stored in data is shared across the X- and Y-axis in the structure of columns and rows. We can create the Scatter plots in order to display the pairwise relationships in addition to the distribution plot displaying the data distribution in the column diagonally.

The **pairplot()** function can also be used to showcase the subset of variables, or we can plot different types of variables on rows and columns.

The Pairplot of the following data set was as follows :

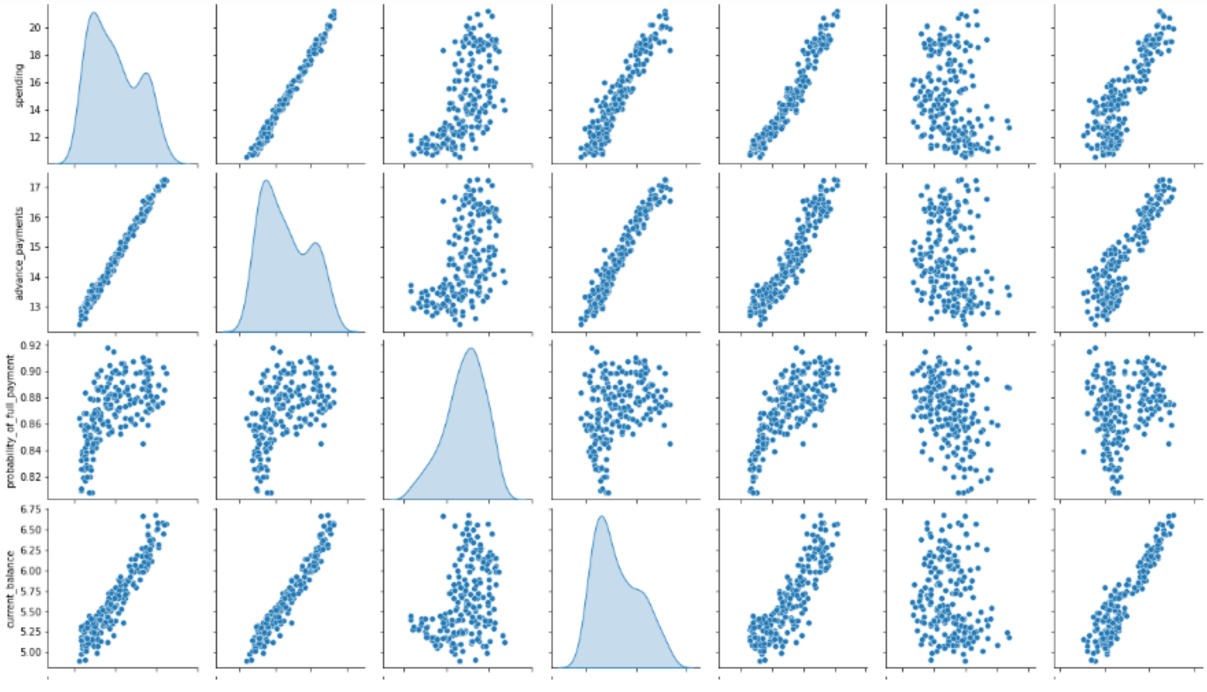


Figure 29 : Pairplot (Bank Marketing Part 1) -1

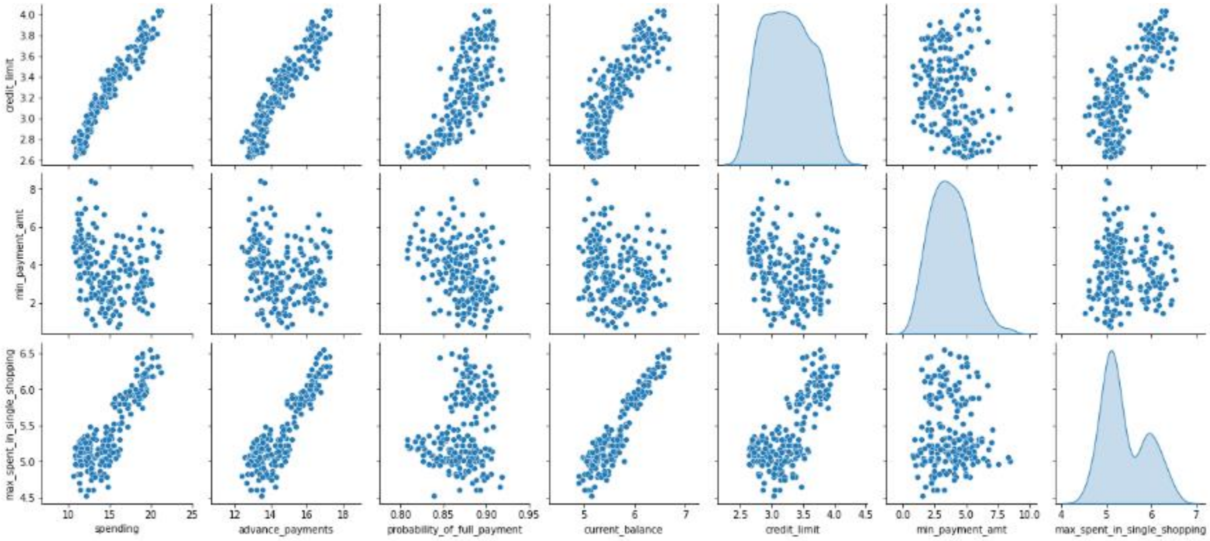


Figure 30 : Pairplot (Bank Marketing Part 1) – 2

Interpretation :

- Firstly, we inferred from scatter plots provided in the pairplot that ‘Spending’ had a very strong correlation with ‘Advance Payments’, ‘Current Balance’, ‘Credit Limit’ & ‘Maximum Spent On Single Shopping’ while on the other hand ‘Spending’ only had moderate positive linear correlation with ‘Probability Of Full Payment’ & ‘Minimum Payment Amount’
- Secondly, we inferred that ‘Advance Payments’ had very good relationship between ‘Spending’, ‘Current Balance’, ‘Credit Limit’ and ‘Maximum Amount Spent In Single Shopping’.
- Except Probability of full payment, rest all were positively (right) skewed, because it's (Probability of Full Payment’ variable end tail was extending towards the left side which indicated the negative (left) skewness.
- Next came ‘Current Balance’ in which we observed that it had a very strong positive correlation with ‘Spending’, ‘ Advance Payments’, ‘Maximum Spent On A Single Shopping’. Meanwhile, there was a moderate positive correlation of ‘Current Balance’ with ‘Probability Of Full Payment’ and ‘Minimum Payment Amount’.
- Adding on to it, we noted that ‘Probability Of Full Payment had a moderate positive linear relationship between every other feature in the data set except for ‘Minimum Payment Amount’ & ‘Maximum Amount Spent on Single Shopping’.
- Moving towards ‘Credit Limit’ , a strong correlation was shown between ‘Credit Limit’ with ‘Spending’ , ‘Advance Payments’ , ‘Probability Of Full Payment’ & ‘Current Balance’. While ‘Credit Limit’ had null relationship with ‘ Minimum Payment Amount’. Furthermore, there were gaps in values between correlation of ‘Credit Limit’ and ‘Maximum Spent On Single Shopping’ which indicated that there were some values present which were higher due to this was recorded in the pairplot.

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

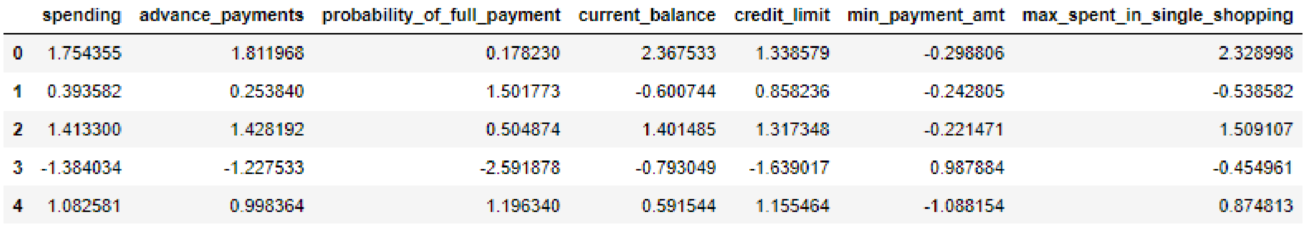
Standardization or scaling is an important aspect of data pre-processing. All machine learning algorithms are dependent on the scaling of data.

Yes, scaling is required in this data set as all features have different weights and to ensure that none of the feature is identified as important only because of the weight, scaling is mandatory for this data set.

Clustering is essentially "grouping close things together and distant things separate". If you don't normalize your features, you will end up giving more weight to some features than others.

Scaling is very important as it normalizes all the observations and thus necessary analytical operations can be performed on the scaled data.

Scaling was done using zscore method and new data frame was created known as df_scaled. Here it is :



	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

Figure 31 : Scaled Data (Problem 1)

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

In hierarchical clustering, Visualization plays an important role in identifying the number of clusters in a set of observations.

Hierarchical clustering (also called **hierarchical cluster analysis** or **HCA**) is a method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:

□ **Agglomerative**: This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

□ **Divisive**: This is a "top-down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

A **Dendrogram** is a tree-like diagram used to visualize the relationship among clusters. More the distance of the vertical lines in the dendrogram, the more the distance between those clusters.

We plotted a **Dendrogram** with **Method = 'Ward'** . It was as follows :

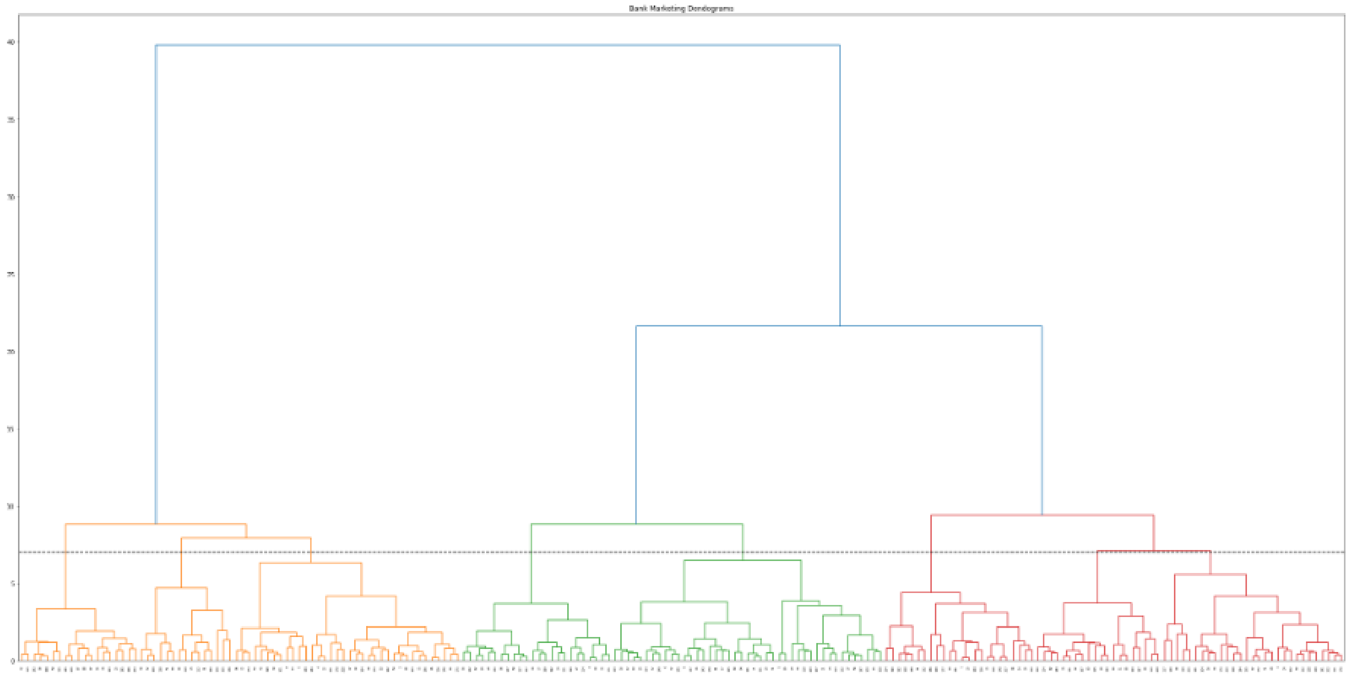


Figure 32 : Dendrogram(Method = Ward)

Besides **'Ward's Method'**, we used another method for plotting Dendrogram and that method was called **'Complete Method'.**

The **Dendrogram** using **Method = 'Complete'** was as follows:

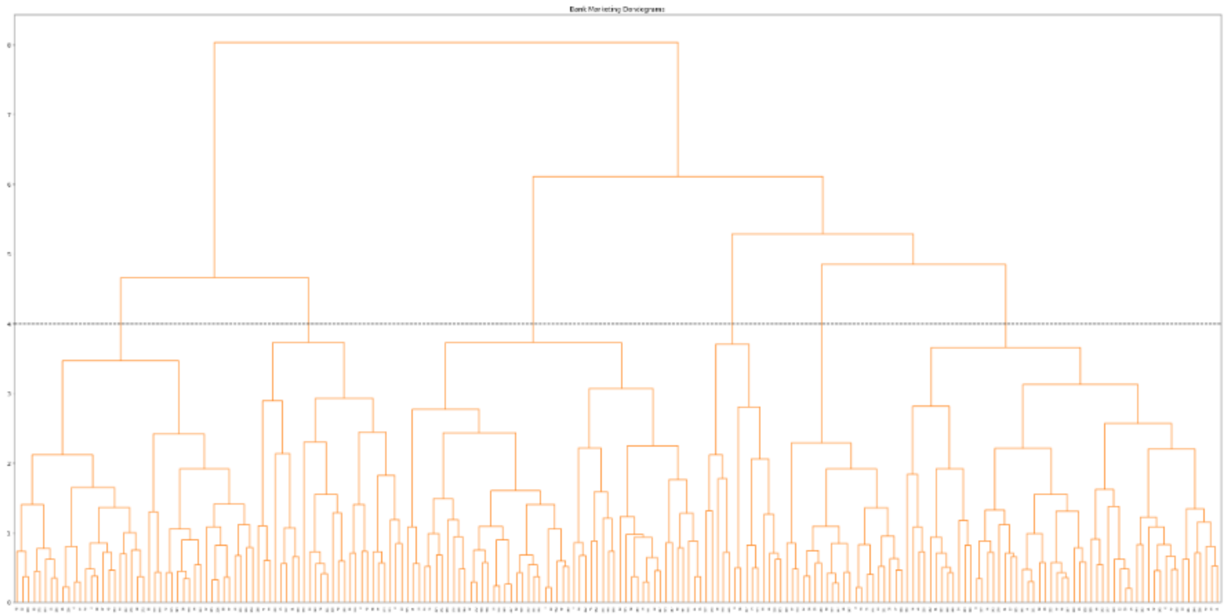


Figure 33 : Dendrogram (Method = 'Complete')

The dendrogram can be hard to read when the original observation matrix from which the linkage is derived is large. So, different parameters are used in order to get visualize the plot more clearly. **Truncation** is a parameter which is used to condense the dendrogram. We also used metric as an important parameter in order to group clusters accordingly.

Truncated Dendrogram by using **Method = ‘Ward’** and **Metric = ‘Euclidean’** was as follows :

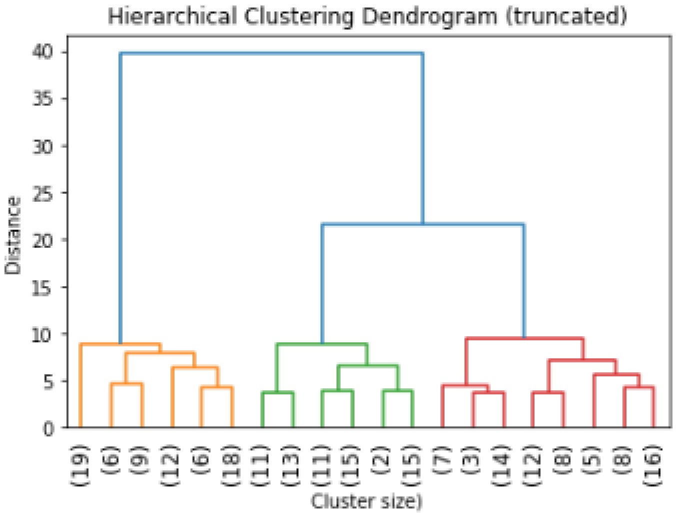


Figure 34 : Hierarchical Clustering Dendrogram Truncated(Ward's Method)

- We inferred from Dendrogram that clusters were differentiated based on their metrics and the ‘Ward’s Method used.
- It showed last 20 merged clusters w.r.t to the Euclidean distance using Ward’s Method. Distance ranges from 0 to 40 in this Dendrogram.

Truncated Dendrogram by using **Method = 'Complete'** and **Metric = 'Euclidean'** was as follows :

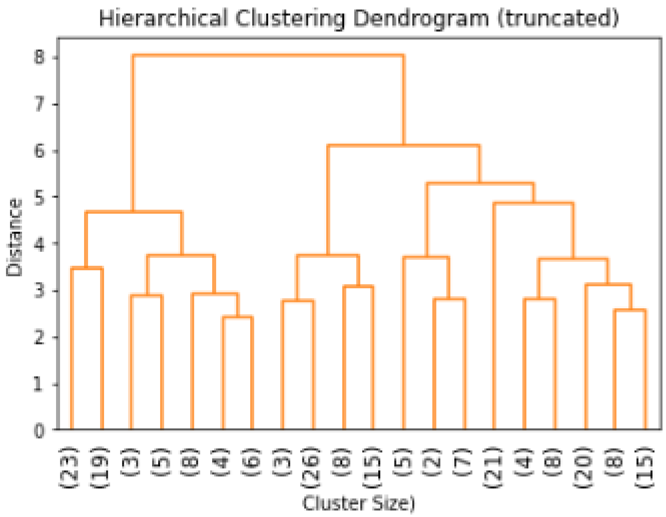


Figure 35 : Hierarchical Clustering Dendrogram Truncated(Complete Method)

- We inferred from Dendrogram that clusters were differentiated based on their metrics and the 'Complete' method used.
- It showed last 20 merged clusters w.r.t to the Euclidean distance using Complete Method. Distance ranges from 0 to 8 in this Dendrogram.

Agglomerative Clustering :

After dendrogram, we performed agglomerative clustering.

Agglomerative Clustering is a type of hierarchical clustering algorithm. It is an unsupervised machine learning technique that divides the population into several clusters such that data points in the same cluster are more similar and data points in different clusters are dissimilar.

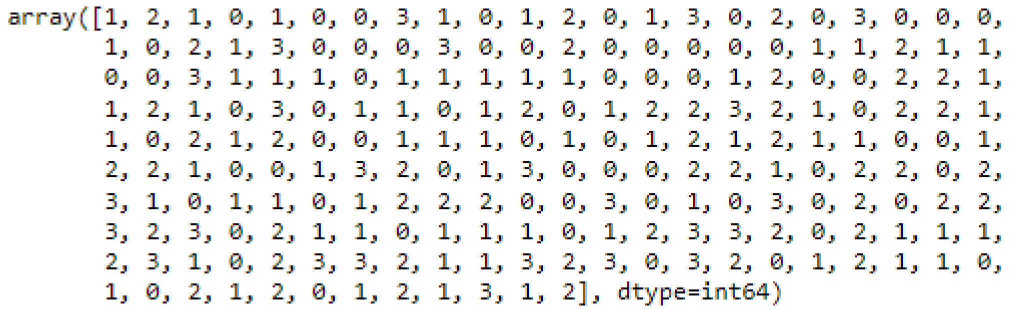
- Points in the same cluster are closer to each other.
- Points in the different clusters are far apart.

Agglomerative Clustering is a bottom-up approach, initially, each data point is a cluster of its own, further pairs of clusters are merged as one moves up the hierarchy.

We used **'Method = Ward'** for further clustering process.

We installed necessary libraries and packages and perform Agglomerative Clustering as well as FCluster for performing Clustering.

Agglomerative Cluster was as follows :

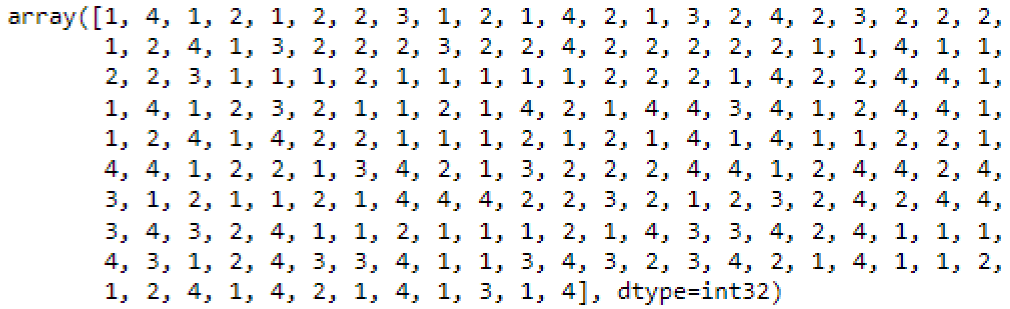


```
array([1, 2, 1, 0, 1, 0, 0, 3, 1, 0, 1, 2, 0, 1, 3, 0, 2, 0, 3, 0, 0, 0,
       1, 0, 2, 1, 3, 0, 0, 0, 3, 0, 0, 2, 0, 0, 0, 0, 0, 1, 1, 2, 1, 1,
       0, 0, 3, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 2, 0, 0, 2, 2, 1,
       1, 2, 1, 0, 3, 0, 1, 1, 0, 1, 2, 0, 1, 2, 2, 3, 2, 1, 0, 2, 2, 1,
       1, 0, 2, 1, 2, 0, 0, 1, 1, 1, 0, 1, 0, 1, 2, 1, 2, 1, 1, 0, 0, 1,
       2, 2, 1, 0, 0, 1, 3, 2, 0, 1, 3, 0, 0, 0, 2, 2, 1, 0, 2, 2, 0, 2,
       3, 1, 0, 1, 1, 0, 1, 2, 2, 2, 0, 0, 3, 0, 1, 0, 3, 0, 2, 0, 2, 2,
       3, 2, 3, 0, 2, 1, 1, 0, 1, 1, 1, 0, 1, 2, 3, 3, 2, 0, 2, 1, 1, 1,
       2, 3, 1, 0, 2, 3, 3, 2, 1, 1, 3, 2, 3, 0, 3, 2, 0, 1, 2, 1, 1, 0,
       1, 0, 2, 1, 2, 0, 1, 2, 1, 3, 1, 2], dtype=int64)
```

Figure 36 : Agglomerative Cluster

We observed in this cluster that the values/observations were now grouped into clusters(0,1,2,3) according to the parameters and labeled as per their respective cluster.

FCluster was as follows :



```
array([1, 4, 1, 2, 1, 2, 2, 3, 1, 2, 1, 4, 2, 1, 3, 2, 4, 2, 3, 2, 2, 2,
       1, 2, 4, 1, 3, 2, 2, 2, 3, 2, 2, 4, 2, 2, 2, 2, 2, 1, 1, 4, 1, 1,
       2, 2, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2, 2, 1, 4, 2, 2, 4, 4, 1,
       1, 4, 1, 2, 3, 2, 1, 1, 2, 1, 4, 2, 1, 4, 4, 3, 4, 1, 2, 4, 4, 1,
       1, 2, 4, 1, 4, 2, 2, 1, 1, 1, 2, 1, 2, 1, 4, 1, 4, 1, 1, 2, 2, 1,
       4, 4, 1, 2, 2, 1, 3, 4, 2, 1, 3, 2, 2, 2, 4, 4, 1, 2, 4, 4, 2, 4,
       3, 1, 2, 1, 1, 2, 1, 4, 4, 4, 2, 2, 3, 2, 1, 2, 3, 2, 4, 2, 4, 4,
       3, 4, 3, 2, 4, 1, 1, 2, 1, 1, 1, 2, 1, 4, 3, 3, 4, 2, 4, 1, 1, 1,
       4, 3, 1, 2, 4, 3, 3, 4, 1, 1, 3, 4, 3, 2, 3, 4, 2, 1, 4, 1, 1, 2,
       1, 2, 4, 1, 4, 2, 1, 4, 1, 3, 1, 4], dtype=int32)
```

Figure 37 : FCluster

We observed in this cluster that the values/observations were now grouped into clusters(1,2,3,4) according to the parameters and labeled as per their respective cluster.

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

K Means Clustering :

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

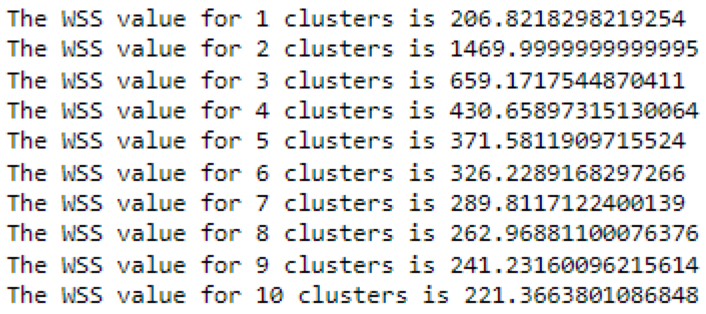
The **K-Means clustering** aims to partition n observations into *k* clusters in which each observation belongs to the cluster whose mean (centroid) is nearest to it, serving as a prototype of the cluster. It minimizes within-cluster variances (squared Euclidean distances).

The most common algorithm uses an iterative refinement technique. Due to its ubiquity, it is often called "the k-means algorithm".

We performed **K-Means Clustering** using necessary libraries and packages.

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

We first found out the WSS Value for the clusters. They were as follows :



```
The WSS value for 1 clusters is 206.8218298219254
The WSS value for 2 clusters is 1469.9999999999995
The WSS value for 3 clusters is 659.1717544870411
The WSS value for 4 clusters is 430.65897315130064
The WSS value for 5 clusters is 371.5811909715524
The WSS value for 6 clusters is 326.2289168297266
The WSS value for 7 clusters is 289.8117122400139
The WSS value for 8 clusters is 262.96881100076376
The WSS value for 9 clusters is 241.23160096215614
The WSS value for 10 clusters is 221.3663801086848
```

Figure 38 : WSS Values (K-Means)

The Elbow method looks at the total WSS as a function of the number of clusters.

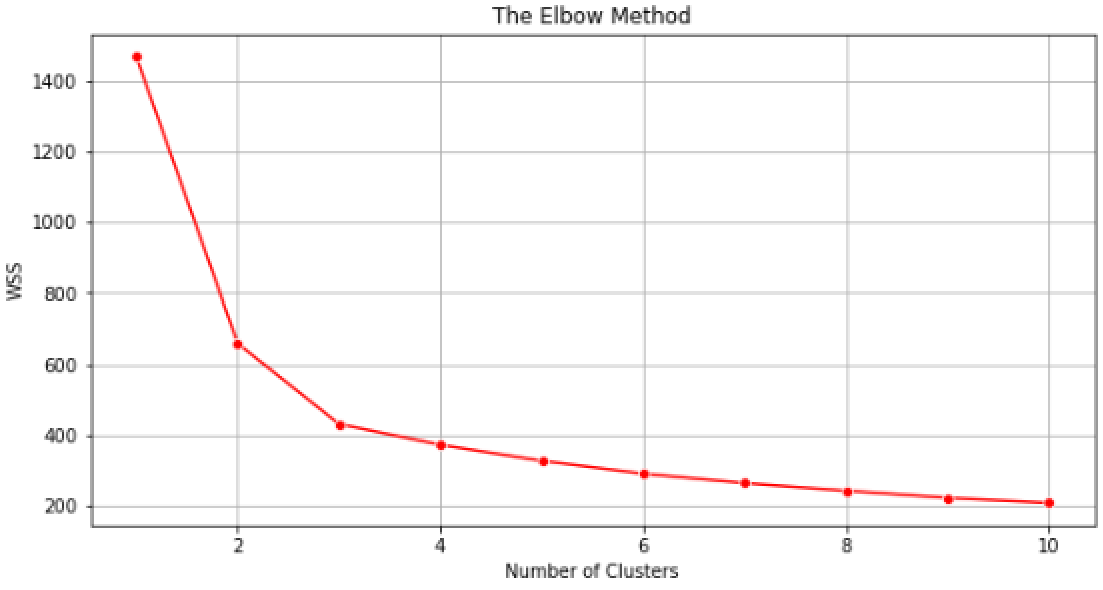


Figure 39 : Elbow Curve(Method)

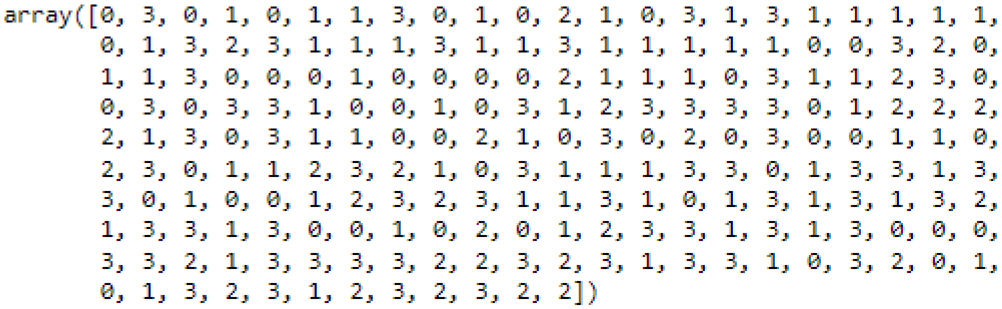
Interpretation :

- - K-means clustering technique was used along with elbow curve to define the optimum clusters for this data set.
- In the Elbow Curve, we saw that there was a steep slope at 2 but we also noticed that after 3 there was a significant decline in the value of WSS which indicated that 3 was the elbow point/turning point in the curve.
- Thus, in other words, the optimal number of clusters are 3 as per the Elbow Curve.

After this, we apply **K means** method where we tell no. of clusters and random state was basically fixing the starting point from where we want to start.

Then we picked up the scaled data and then use the k means we used the above code and put **no. of clusters =4** and got the output by the name **'labels'**.

The output was as follows :



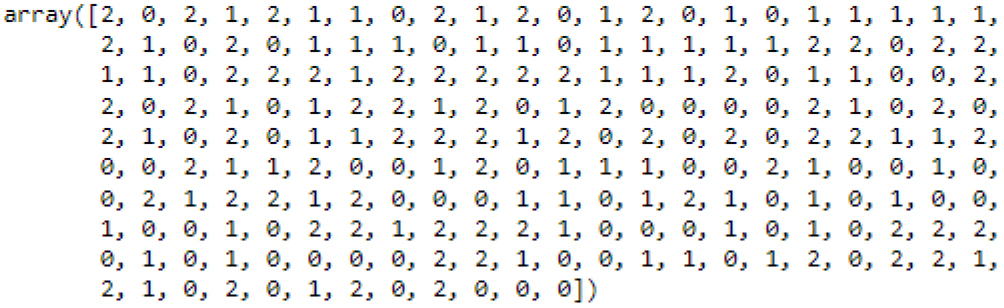
```
array([0, 3, 0, 1, 0, 1, 1, 3, 0, 1, 0, 2, 1, 0, 3, 1, 3, 1, 1, 1, 1, 1,
       0, 1, 3, 2, 3, 1, 1, 1, 3, 1, 1, 3, 1, 1, 1, 1, 1, 0, 0, 3, 2, 0,
       1, 1, 3, 0, 0, 0, 1, 0, 0, 0, 0, 2, 1, 1, 1, 0, 3, 1, 1, 2, 3, 0,
       0, 3, 0, 3, 3, 1, 0, 0, 1, 0, 3, 1, 2, 3, 3, 3, 3, 0, 1, 2, 2, 2,
    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

Figure 40 : Labels (K-Means where no. of clusters =4)

We observed in this cluster that the values/observations were now grouped into clusters(0,1,2,3) according to the parameters and labeled as per their respective cluster.

After trying out for no. of clusters = 4 , we then checked for the **"no. of clusters"** that Elbow Curve Method showed us that is **"3"** and it was named as **labels_3**.

The output for **labels_3** was as follows :



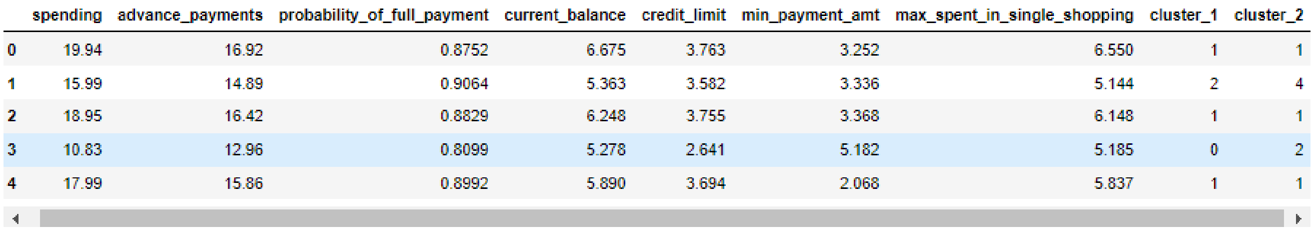
```
array([2, 0, 2, 1, 2, 1, 1, 0, 2, 1, 2, 0, 1, 2, 0, 1, 0, 1, 1, 1, 1, 1,
       2, 1, 0, 2, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 2, 2, 0, 2, 2,
       1, 1, 0, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 1, 2, 0, 1, 1, 0, 0, 2,
       2, 0, 2, 1, 0, 1, 2, 2, 1, 2, 0, 1, 2, 0, 0, 0, 0, 2, 1, 0, 2, 0,
    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

Figure 41 : Labels_3 (K-Means where no. of clusters = 3)

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

We did cluster profiling and add them into the original dataset .

The **Cluster Profiling** for **Agglomerative Clustering** & **FCluster** was as follows :



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	cluster_1	cluster_2
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	1	1
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	2	4
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1	1
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	0	2
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837	1	1

Figure 42 : Cluster Profiling (Agglomerative Clustering & FCluster)

After profiling, we evaluated both profiles for K-Means, interpreted the results and got the insights from it as follows :

Interpretation :

- The total number of clusters derived from Agglomerative/FCluster methods were same and were equal to 4.

- We inferred that, the cluster 1 & cluster 2(as per index) in **Cluster 1** had given highest ranking to all features except minimum payment amount as compared to others clusters. It means that the customers in these particular clusters tends to spend the most amount of money per month and giving advance payments in cash along with a highest probability value for doing full payment to bank and rest of the features also except for minimum payment amount made while making purchases. It was quite interesting to note that as highest values for making advance payments were already done in cash by these customers, the minimum amount paid is quite good as well. With a remarkable highest probability values of making full payments clearly shows that the always made their payments on time and can be considered as great customers.

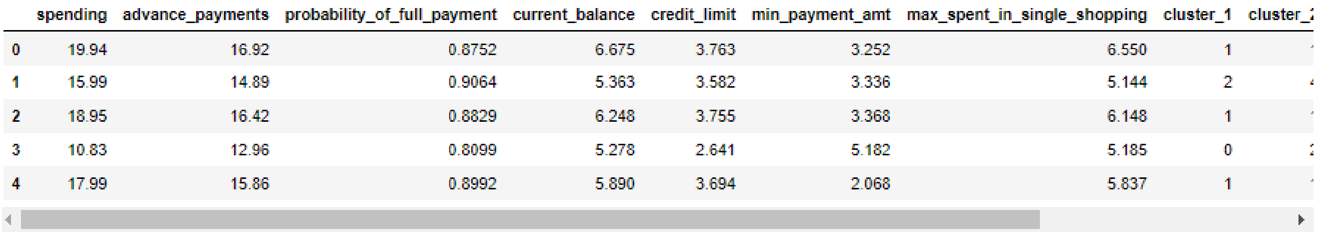
- While on the other hand, cluster 0(as per index) in **Cluster 1** had given best ranking to minimum payment amount as compared to the other clusters. Adding on , cluster 0 had lowest value in spending and at the same time , the customers in these cluster group gives highest ranking in minimum payment while making purchases. This shows the correlation between these two features in cluster 0(as per index).

- Moving on, when we looked at cluster 3(as per index) in **Cluster 1** , the cluster count(customer frequency) is lowest among the others which indicates how precisely the clustering is done in this cluster. From this cluster, we determined that maximum amount spend on one single purchase is lowest in this cluster as the customers in this cluster don't tend to spend excess amount of money as compared to other clusters.

- We noted that, the advance payments were made in cash and it was directly proportional to spending. Spending amount generally include payments made in cash and payment done via credit card and Upi(indirectly credit card/net banking).

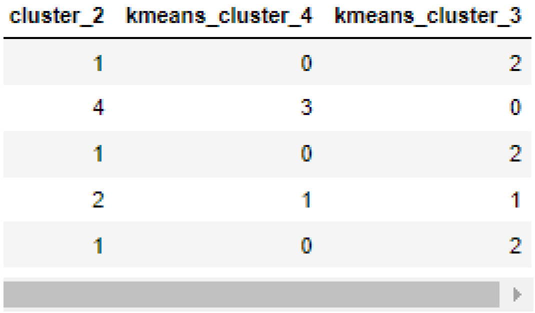
The **Cluster Profiling** for **K-Means Clustering** were as follows :

K-Means Profiling :



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	cluster_1
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

Figure 43 : K-Means Profiling - 1



cluster_2	kmeans_cluster_4	kmeans_cluster_3
1	0	2
4	3	0
1	0	2
2	1	1
1	0	2

Figure 44 : K-Means Profiling - 2

After profiling, we evaluated both profiles for K-Means, interpreted the results and got the insights from it as follows :

From profiling both the clusters for no. of clusters = 3 & no. of clusters =4, we determined that , the **Final Number of Clusters** for **K-Means** Method is 3.

Interpretation :

- We inferred that cluster 2(as per index) in **kmeans_cluster_3** had given the highest ranking among all features except for minimum payment amount as compared to other clusters.

- We can also state that cluster 1(as per index) in **kmeans_cluster_3** had given the highest ranking in the minimum payment amount while making purchases and simultaneously the same cluster had the least ranking in the spending feature as compared with other clusters.

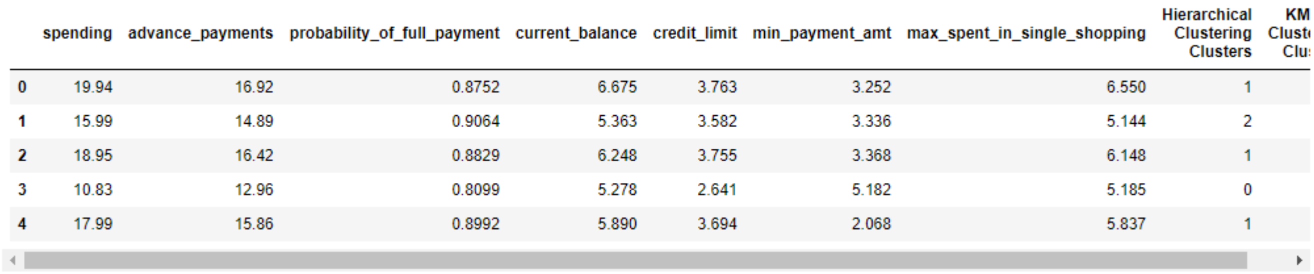
- We noticed that probability value of payment done in full was almost equal in cluster 0 & cluster 2(as per index) in **kmeans_cluster_3** with cluster 2 having the upper hand while the remaining cluster that was cluster 1(as per index) had lowest value for probability of full payment.

Recommendations For Promotional Strategies :

- From **Agglomerative Clustering** , we determined that the **Final Optimal Number of Clusters were equal to 4** while on the other hand, from **K-Means Clustering** the **Final Optimal Number of Clusters were determined to be 3**.
- From both the clustering we observed that the cluster having highest ranking in minimum payment amount while making purchases also had the lowest ranking in spending amount by the customer which means that the customers who paid minimal amount for making purchases would unlikely had very less amount of probability of making payment in full as these customers preferred to pay minimum amount while conducting purchases. So in this case, the **recommendations** were as follows:
    - Bank should develop and advertise cashback offers on variety of purchases but with a condition of slightly high amount of minimum payment for that product during purchases which will ultimately lead to increase in monthly spending by the customers as well as increase in the minimum paid amount by customer which making purchases.
- From Agglomerative clustering, we noted that at cluster 3(as per index) in __cluster_1__ , the cluster count(customer frequency) is lowest among the others which indicates how precisely the clustering is done in this cluster. From this cluster, we determined that maximum amount spend on one single purchase is lowest in this cluster as the customers in this cluster don't tend to spend excess amount of money as compared to other clusters. **Recommendations** for this were as follows:
    - Bank should promote offers such as easy to go with cashless payments(using credit cards, net-banking etc) are very much reliable nowadays. For example, when payment is done at petrol pump via credit card then credit points are added and these points are used for redeeming gift vouchers, cashback and various products. Thus by availing these banking services, customers will spend more using the bank services which will lead to increase in their credit limit increase which will directly result in increase in maximum spent on a single shopping.
- We noted that, the advance payments were made in cash and it was directly proportional to spending. Spending amount generally include payments made in cash and payment done via credit card and Upi(indirectly credit card/net banking). So, recommendations were as follows :
    - For marketing more promotional offers should be given to those customers who will avail this particular bank credit cards and Upi(technically linked with card or user bank account). This would made the customers automatically reaching the bank for opening account in that respective bank and for those who were already customers of that bank, then they would request the bank to issue credit card for them so that they could avail promotional offers while making purchases in their daily life.

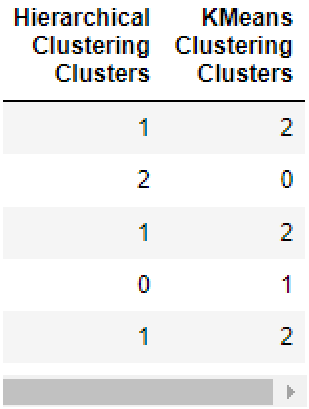
After this, we added the optimum clusters derived from Hierarchical Clustering and K-Means clustering to the original data and export it to a '.csv' file.

Optimal Clusters after adding into the original dataset looked as follows :



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Hierarchical Clustering Clusters	KM Clust Clu
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	

Figure 45 : Final Output – 1



Hierarchical Clustering Clusters	KMeans Clustering Clusters
1	2
2	0
1	2
0	1
1	2

Figure 46 : Final Output – 2

The .csv file where it exported was named as **“Clustering Project.csv”.**

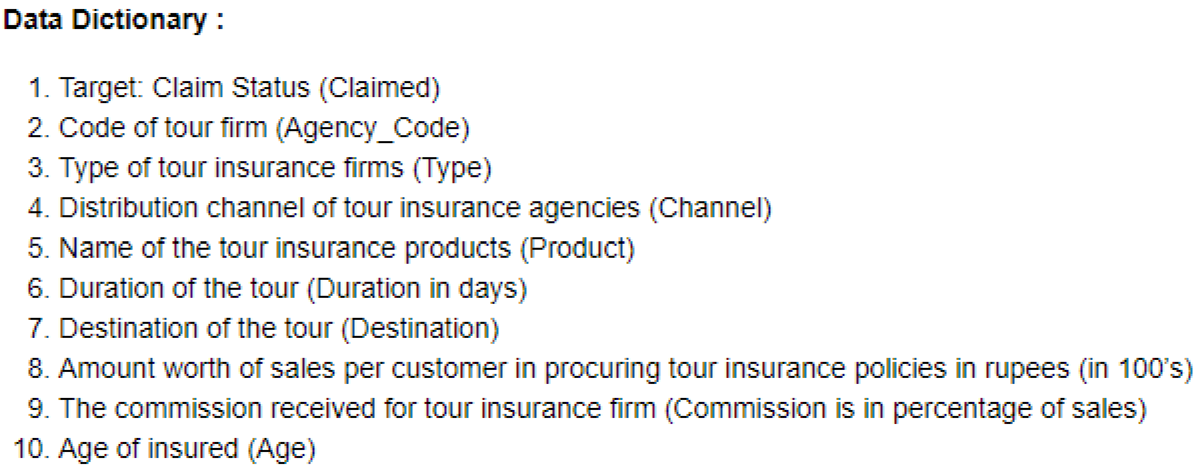
Problem 2 – CART-RF

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

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

At first we loaded the Data Dictionary for understanding of column name for data set "Bank Marketing Part 1".

The Data Dictionary is as follows :



Data Dictionary :

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)

Figure 47 : Data Dictionary (Problem 2)

After loading data dictionary and reading the data, we got to know some insights as follows:

- We inferred that there were three types of data types in this dataset which were integer, object and float.

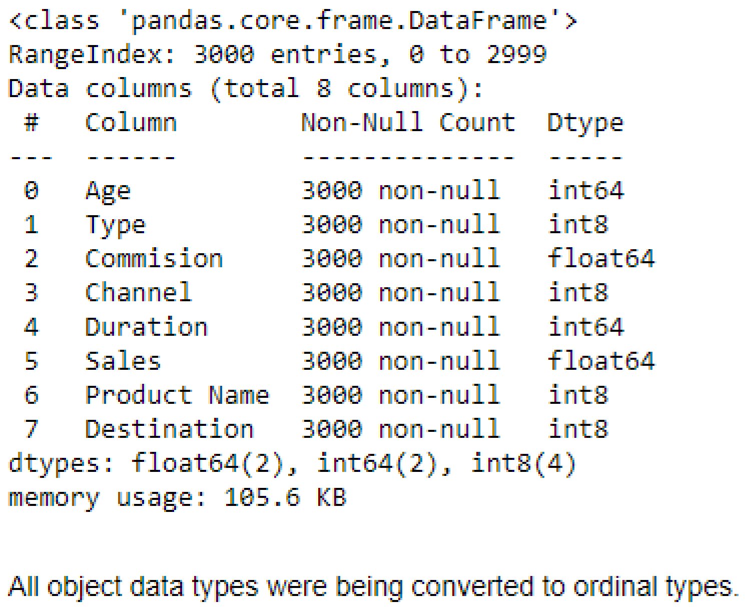
- There were 2 columns with 'float' data type, 2 columns with 'integer' data type and 6 columns with 'object' data type.

- Many columns are of type object i.e. strings. These need to be converted to ordinal type.

Before transforming them into ordinal types , we dropped two columns that were 'Agency_Code' & 'Claimed' and then we did the transforming.

The transformed data was then put into a new dataframe known as **‘data_object’.**

The info about the new dataframe **‘data_object’** was as follows :



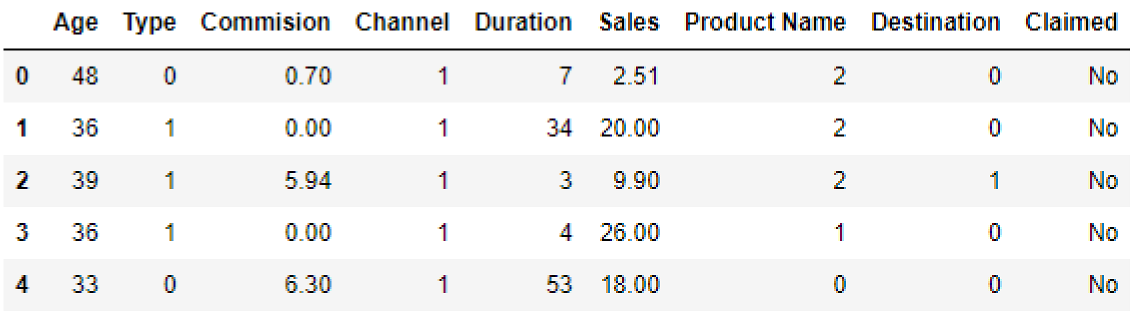
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Age           3000 non-null   int64  
 1   Type          3000 non-null   int8   
 2   Commision     3000 non-null   float64
 3   Channel       3000 non-null   int8   
 4   Duration      3000 non-null   int64  
 5   Sales         3000 non-null   float64
 6   Product Name  3000 non-null   int8   
 7   Destination   3000 non-null   int8   
dtypes: float64(2), int64(2), int8(4)
memory usage: 105.6 KB
```

All object data types were being converted to ordinal types.

Figure 48 : data_object(Info)

After that, the column **‘Claimed’** was restored back into the **data_object** because it was the target variable/dependent variable in this problem.

First 5 Observations Of data_object :



	Age	Type	Commision	Channel	Duration	Sales	Product Name	Destination	Claimed
0	48	0	0.70	1	7	2.51	2	0	No
1	36	1	0.00	1	34	20.00	2	0	No
2	39	1	5.94	1	3	9.90	2	1	No
3	36	1	0.00	1	4	26.00	1	0	No
4	33	0	6.30	1	53	18.00	0	0	No

Figure 49 : First 5 Observations Of data_object

EDA

Univariate Analysis

For Univariate analysis, we plotted a Distribution plot and a Boxplot for each column provided in the data set .

The Distribution plot was used for univariant set of observations and visualizes it through a histogram i.e. only one observation and hence we choose one particular column of the dataset.

And, Boxplot was used as a measure of how well the data is distributed in a data set. It divides the data set into three quartiles. This graph represents the minimum, maximum, median, first quartile and third quartile in the data and also shows us whether there are outliers or not.

Here the Distribution Plot and Boxplot for **Insurance Part 2 Data-1** data set :

Age :

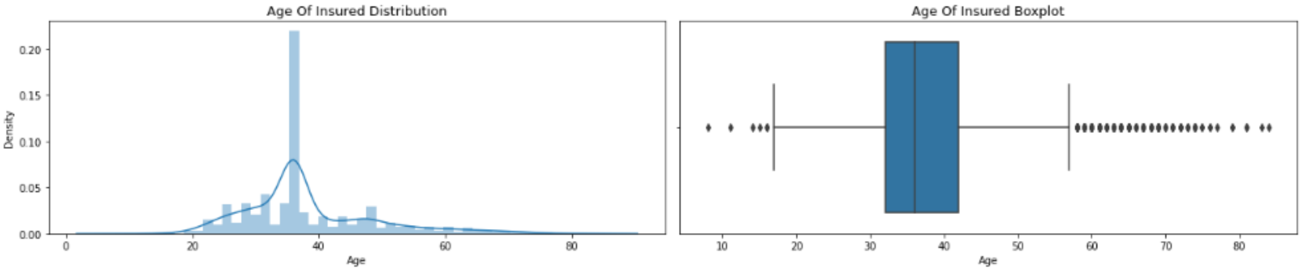


Figure 50 : Distplot & Boxplot (Age)

The ‘Age’ variable showed many outliers as per the Boxplot before the minimum and maximum range (As per the 5 Point Summary).

As per the Distplot, we inferred that ‘Age’ was a positively right skewed distribution .

Type :

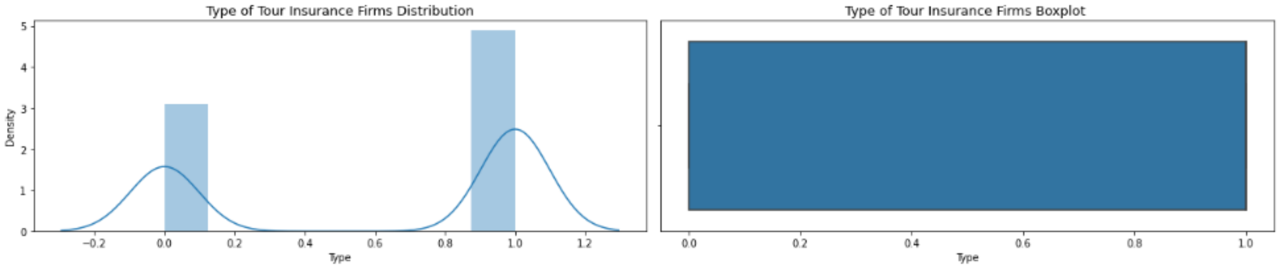


Figure 51 : Distplot & Boxplot (Type)

The Distplot tells us that the graph is negatively (left) skewed as its tail was at the left side of the distribution. Skewness value can be cross checked from the descriptive analysis section.

The ‘Type’ variable showed no outliers as per the Boxplot which means all values were laying inside the range from minimum to maximum.

Commision :

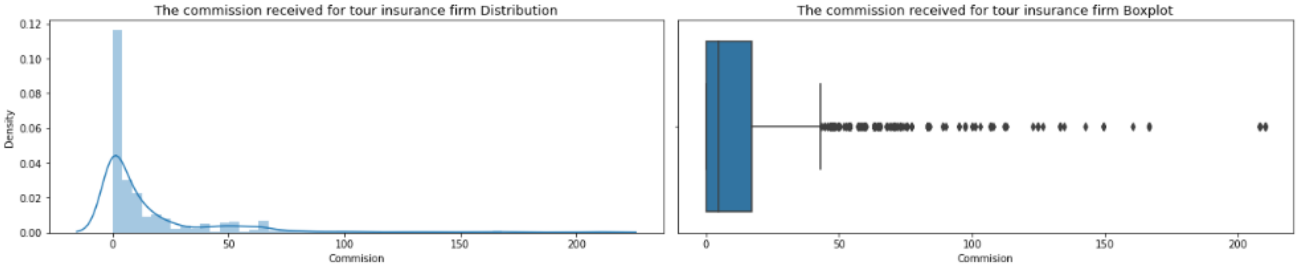


Figure 52 : Distplot & Boxplot (Commision)

The ‘Commision’ variable showed many outliers in the Boxplot which indicated that there were many values which exceed the maximum value (from 5 point summary).

As we go to the Distplot, we got to know that ‘Commision’ was positively right skewed as its tail was at the left side of the distribution .

Channel :

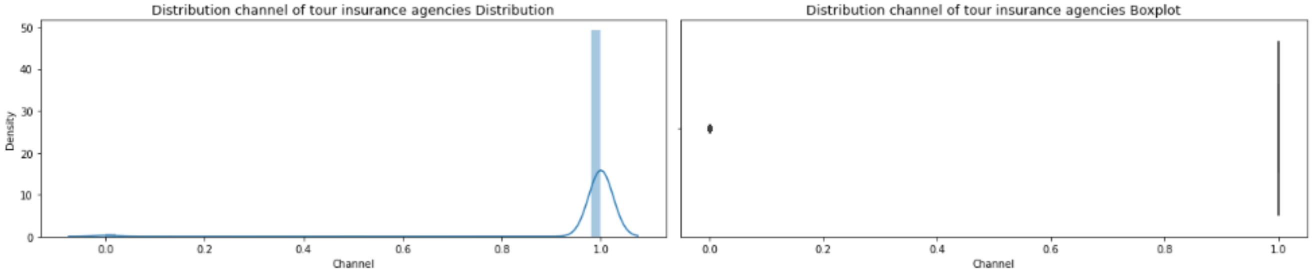


Figure 53 : Distplot & Boxplot (Channel)

The ‘Channel’ variable showed only 1 outlier as per the Boxplot which means only 1 value were laying outside the minimum range (as per the 5 Point summary). It had minimum skewness value among all other variables.

As per the Distplot & skewness value, we got to know that ‘Channel’ there were values present in low amount which breaks the normal distribution curve but it was close to normal distribution.

Hence, as per the skewness value, it was considered as negatively (left) skewed as its tail was at the left side of the distribution.

Duration :

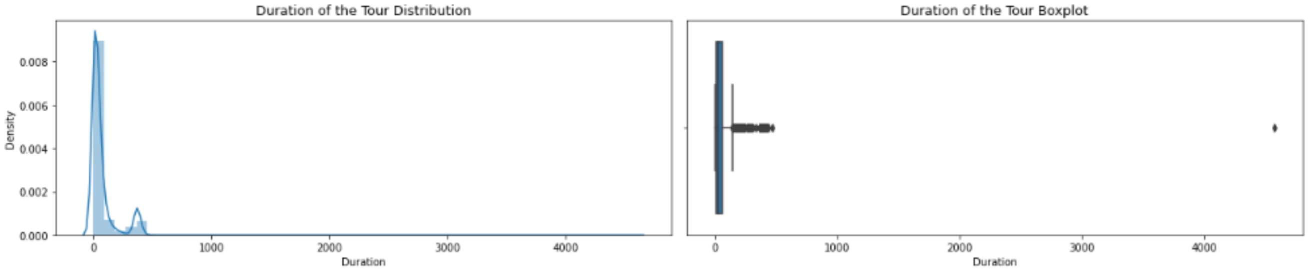


Figure 54 : Distplot & Boxplot (Duration)

The ‘Duration’ variable showed many outliers in the Boxplot which indicated that there were many values which exceed the maximum value (from 5 point summary).

As we go to the Distplot, we got to know that ‘Duration’ was positively right skewed as its tail was at the right side of the distribution. It had highest skewness among all other variables.

Sales :

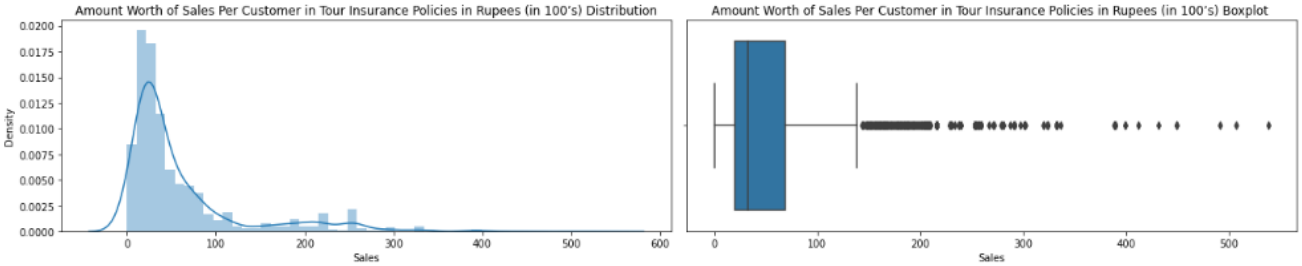


Figure 55 : Distplot & Boxplot (Sales)

The 'Sales' variable showed many outliers in the Boxplot which indicated that there were many values which exceed the maximum value (from 5 point summary).

As we go to the Distplot, we got to know that 'Sales' was positively right skewed as its tail was at the right side of the distribution.

Product Name :

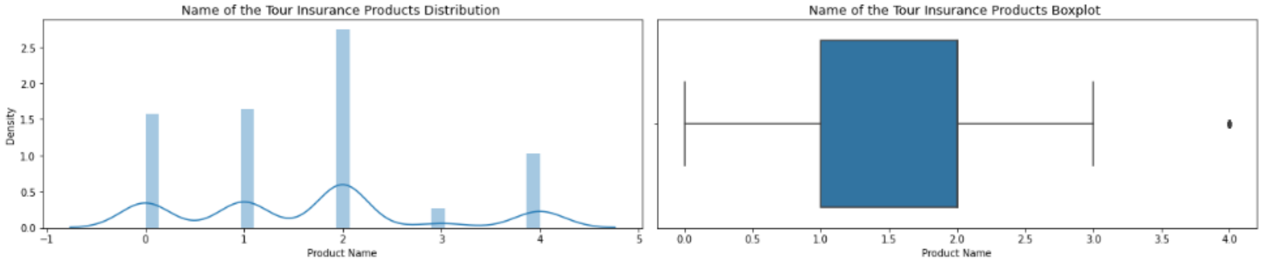


Figure 56 : Distplot & Boxplot (Product Name)

The 'Product Name' variable showed only 1 outlier as per the Boxplot which means only 1 value were laying outside the maximum range (as per the 5 Point summary).

As per the Distplot & skewness value, we got to know that 'Product Name' was almost a normally distribution (neither left nor right skewed) but as we saw to the right side of the Distplot, we saw there were values present in high amount which breaks the normal distribution curve but it was close to normal distribution.

Hence, as per the skewness value, it was considered as positively (right) skewed as its tail was at the right side of the distribution.

Destination :

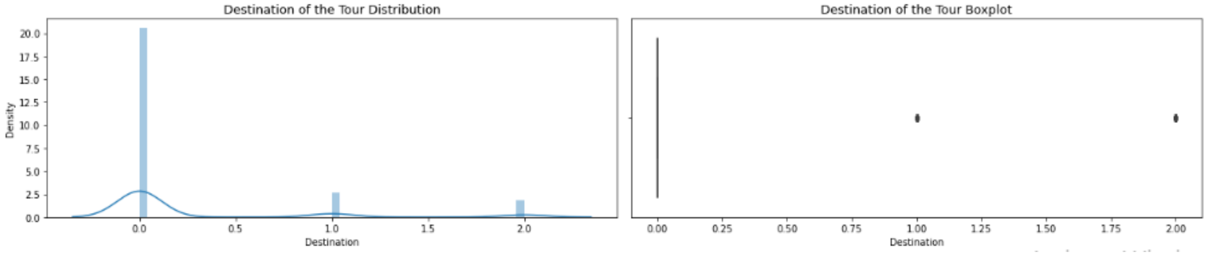


Figure 57 : Distplot & Boxplot (Destination)

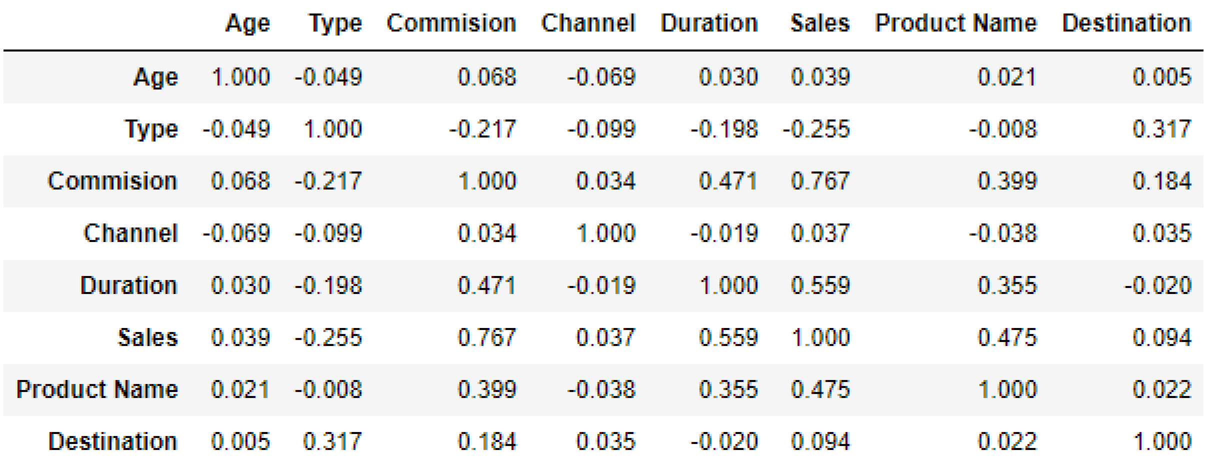
The ‘Destination’ variable showed only 2 outlier as per the Boxplot which means only 2 values were laying outside the maximum range (as per the 5 Point summary) which means that these values were far away from the majority.

As per the Distplot & skewness value, we got to know that there were values present in high amount which breaks the normal distribution curve but it was close to normal distribution.

Hence, as per the skewness value, it was considered as positively (right) skewed as its tail was at the right side of the distribution.

Bi-Variate Analysis :

- **Correlation Matrix :**
  For this data, we did the correlation matrix and find out many insights from it. It is as follows :



	Age	Type	Commision	Channel	Duration	Sales	Product Name	Destination
Age	1.000	-0.049	0.068	-0.069	0.030	0.039	0.021	0.005
Type	-0.049	1.000	-0.217	-0.099	-0.198	-0.255	-0.008	0.317
Commision	0.068	-0.217	1.000	0.034	0.471	0.767	0.399	0.184
Channel	-0.069	-0.099	0.034	1.000	-0.019	0.037	-0.038	0.035
Duration	0.030	-0.198	0.471	-0.019	1.000	0.559	0.355	-0.020
Sales	0.039	-0.255	0.767	0.037	0.559	1.000	0.475	0.094
Product Name	0.021	-0.008	0.399	-0.038	0.355	0.475	1.000	0.022
Destination	0.005	0.317	0.184	0.035	-0.020	0.094	0.022	1.000

Figure 58 : Correlation Matrix (Problem 2)

Interpretations :

- A Correlation Matrix was created above using the 'Pearson' method.
- The matrix depicts the correlation between all the possible pairs of values in a table. It is a powerful tool to summarize a large dataset and to identify and visualize patterns in the given

-For better understanding, a graphical representation in the form of heatmap was also created with respect to Correlation Matrix.

- **Heatmap** :
  A **Correlation Heatmap** is a rectangular representation of data and it repeats the same data description twice because the categories are repeated on both axis for computing analysis. Hence, the same result is obtained twice. A correlation heatmap that presents data only once without repetition that is categories are correlated only once is known as a **Triangle Correlation Heatmap**.

Triangle Correlation Heatmap for **Insurance Part 2 Data-1** data set is as follows :

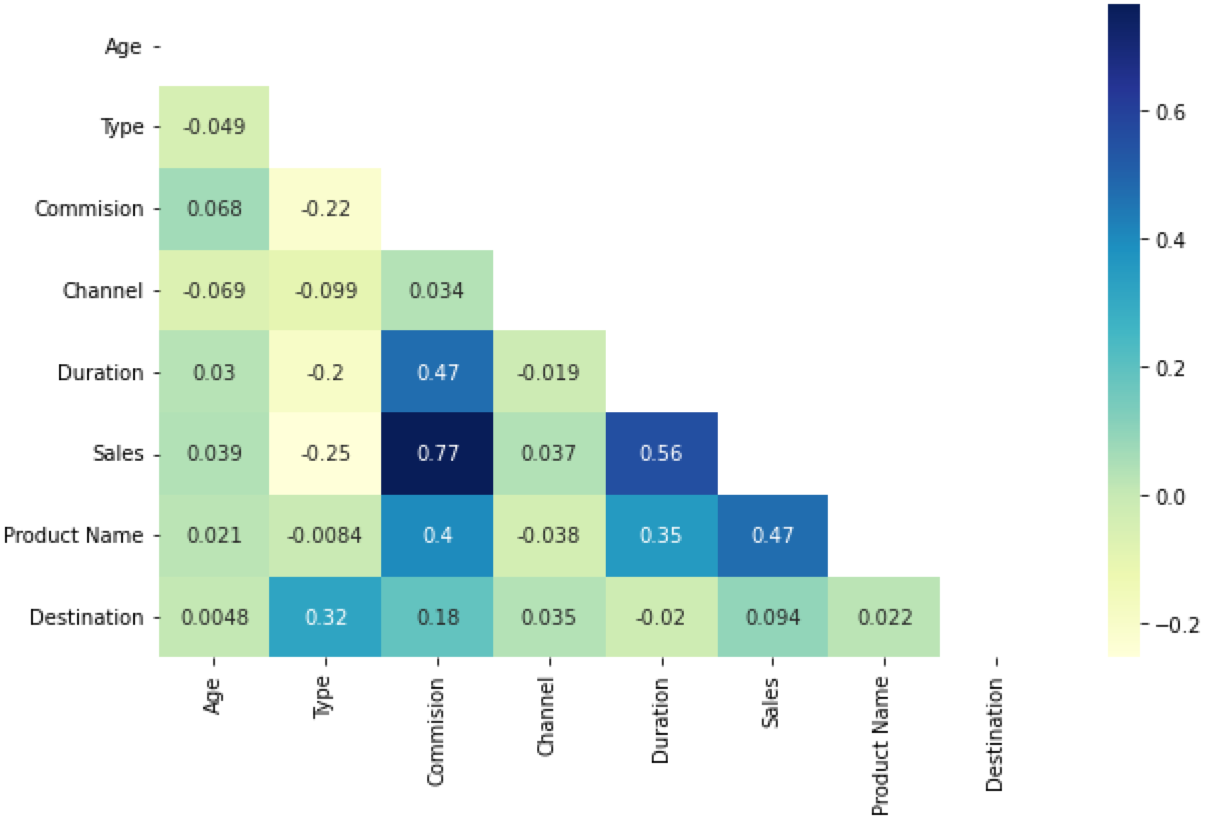


Figure 59 : Heatmap (Problem 2)

Interpretations :

- We inferred that there was 77% correlation between sales and commision. It means that as the Amount of Sales per customer for tour insurance policies increases, the Commission for Tour Insurance firms increases.
- Many of the features had very less value for correlation which indicates that the these features are does not relate to each other. In other words, they will not affect the performance of other features.
- Moving on towards the lowest correlation , 'Commission' and 'Type' had lowest correlation among each other. It meant that commission received for Tour Insurance Firm decrease as per the type of Tour Insurance Firm.
- Adding on, there is 56% correlation between 'Sales' & 'Duration'. It tells us that duration of the tour affects the amount of sales per customer for tour insurance policies in a positive way as they were directly proportional to each other.

Multi-Variate Analysis :

Pairplot :

In this we found out the **Pairplot** of the original dataset . The **pairplot()** function can also be used to showcase the subset of variables, or we can plot different types of variables on rows and columns.

The Pairplot of the following data set was as follows :

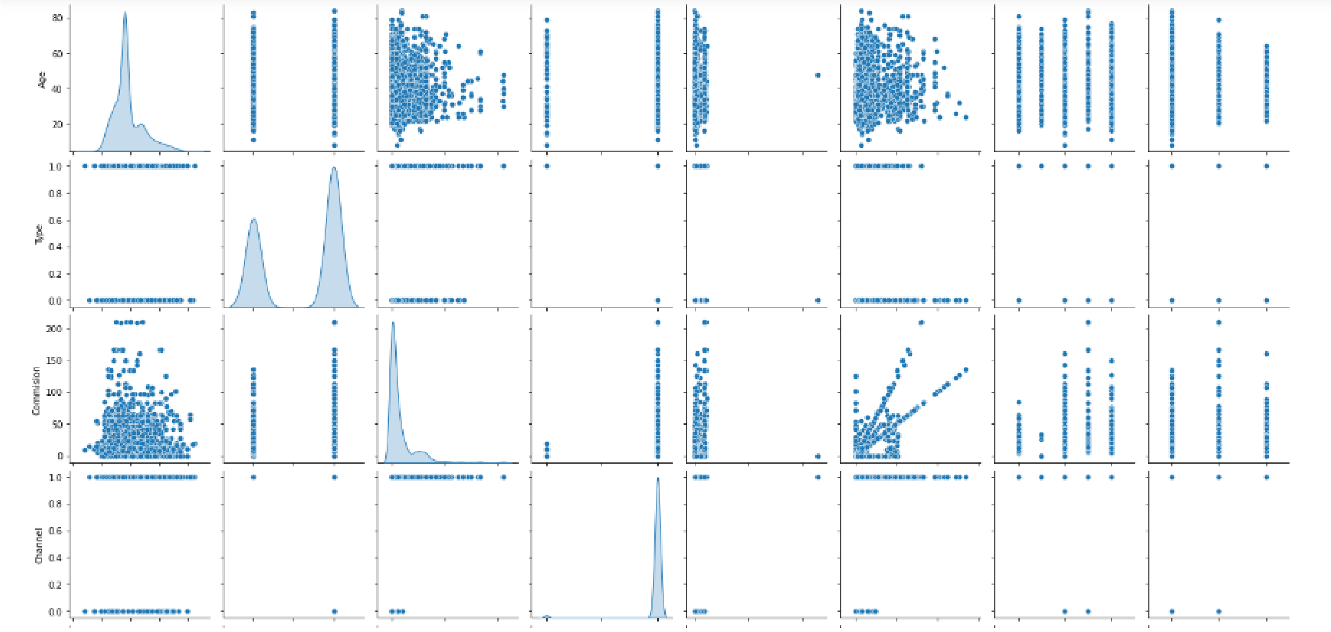


Figure 60 : Pairplot (Problem 2) – 1

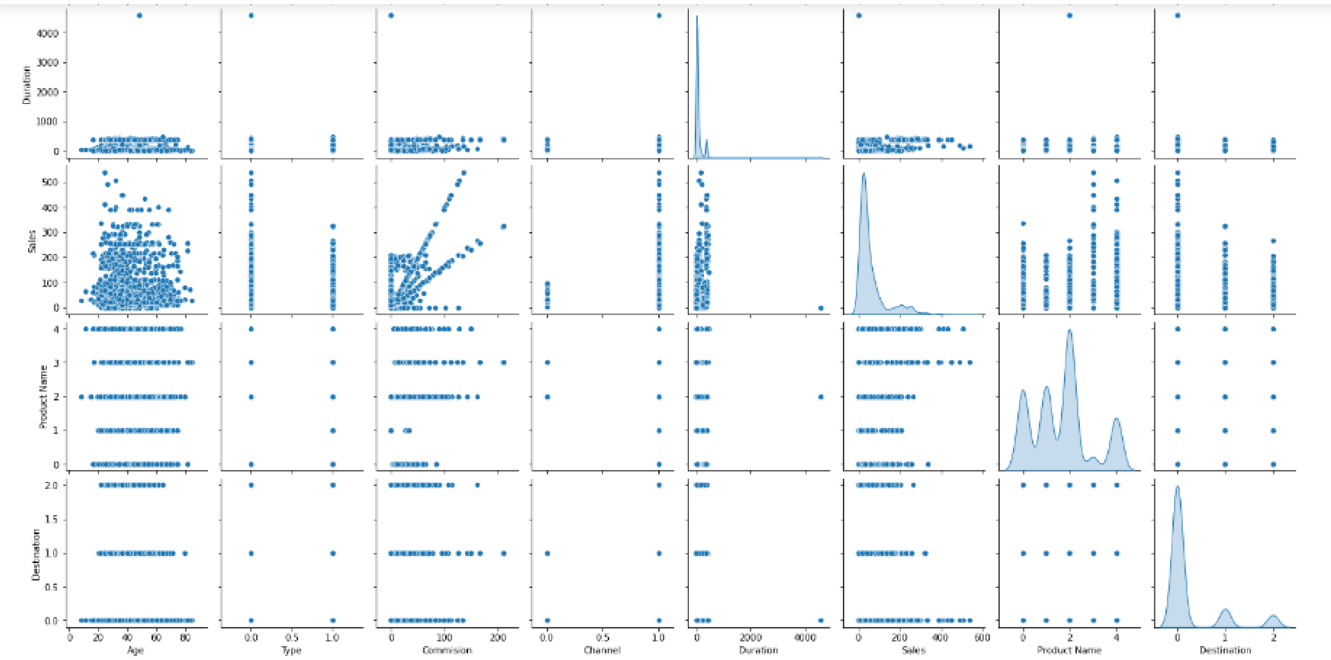


Figure 61 : Pairplot (Problem 2) -2

Interpretations :

- Many of the features had very less value for correlation which indicates that the these features are does not relate to each other. In other words, they will not affect the performance of other features.
- We observed that Age had almost null correlation with every variable in the data set.

- We inferred that there was a very strong correlation between sales and commision. It means that as the Amount of Sales per customer for tour insurance policies increases, the Commission for Tour Insurance firms increases.
- Adding on, there is moderate negative linear correlation between 'Sales' & 'Duration'. It tells us that duration of the tour affects the amount of sales per customer for tour insurance policies in a positive way as they were directly proportional to each other.
- Moving on towards the lowest correlation , 'Commission' and 'Type' had lowest relationship in their correlation among each other. It meant that commission received for Tour Insurance Firm decrease as per the type of Tour Insurance Firm.

Problem 2.2 Data Split: Split the data into test and train, build classification model CART, Random Forest.

We did Splitting the Data into Train and Test (size = 0.30) & random_state = 1

Random State :

Random state value is nothing but setting a seat value which is to ensure uniformity when we are the same random generation function across multiple systems. The integer value of random state must remain same if we are passing the same train test spilt function in a separate system so as to get the same set of observations/records.

Importance Of Random State :

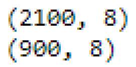
- The important thing is that every time you use any natural number, you will always get the same output the first time you make the model which is similar to random state while train test split.

After getting the concept of random state, we capture the target column ("Claimed") into separate vectors for training set and test set using drop() and pop() function.

The test size was considered to be 30% for testing and 70% was considered for training as a standard for better modeling.

After separating the vectors for training set & test set , we again split the data into training and test set for independent attributes.

Thereafter we checked the shape of the train and test set. It was as follows :



```
(2100, 8)
(900, 8)
```

- It means that 2100 were the Training for independent Variables(X_train).
- It means that 900 were the Testing Independent Variables(X_test).
- It means that 8 were the Training Dependent Variables(train_labels).
- It means that 8 were the Testing Dependent Variables(test_labels).

Building classification model CART - Decision Tree

We imported the necessary libraries and packages for building the models.

- At first, we builded a decision tree model using criterion = ‘gini’ and random_state =1 only and fit that model into Training independent and dependent set.

- The Importance of features in the tree building (The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance).
- After fitting the model , we plotted the tree. The output was as follows:

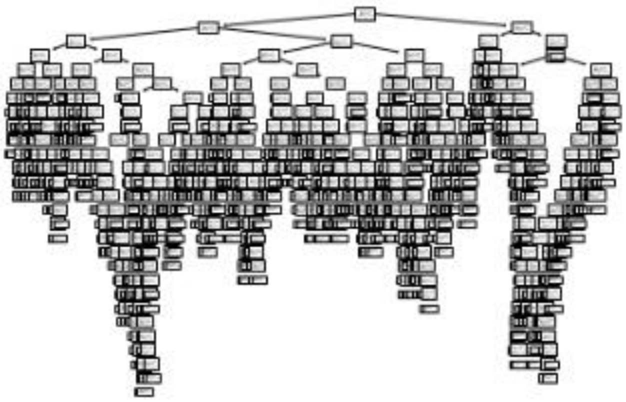
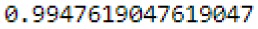
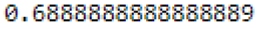


Figure 62 : Decision Tree (dt_model)

- For understanding it , we checked the scored for that decision tree model.
- The score for X_train and train_labels was 

```
0.9947619047619047
```

.
- And the score for X_test and test_labels was 

```
0.6888888888888889
```

.
- By checking the score , it was clear that the model build was over-fitted. So, we must regularize it using different parameters.

Pruning/Regularization a decision tree

- For pruning we created a new decision tree model by using Decision Tree Classifier and adding some certain parameters such as max_depth, min_sample_leaf, min_sample_split etc and then fitted that model into Training independent and dependent set. This pruned decision tree was called as **reg_dt_model1**.
- Then we plotted that new pruned decision tree . The output was as follows:

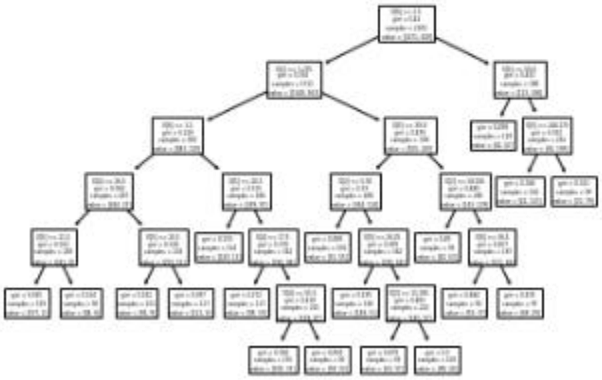


Figure 63 : Regularized Decision Tree(red_dt_model1)

Interpretation :

- By using multiple parameters such as max_depth, min_samples_split, min_samples_leaf, we were able to build a very good neat and clean regularized decision tree model.

- Now, the regularized/pruned decision tree looked simpler, and much more easy to understand.
- For understanding it , we checked the scored for that regularized decision tree model.
- The score for X_train and train_labels was 0.7766666666666666 .
- And the score for X_test and test_labels was 0.7611111111111111 .
- Comparing the scores, we determined that this regularized decision tree model is well build.

Finding Best Decision Tree Model

- Pruning helped us a lot in making the decision tree model better but we also needed to find the best decision tree model.
- For that, we needed the use GridSearch_CV function in order to achieve our goal.
- We added a param in which all the parameters were set up and a new decision tree model was build known as dtree_model.
- As done above, we again fitted this model into Training independent and dependent set.
- Thereafter , using GridSearch_cv, this decision tree was run again with score parameter = 'accuracy' and fit the same as required by the problem statement .
- Because of this , we were able to build best decision tree model and after checking its score.
- The score for X_train and train_labels was 0.7685714285714286 .
- And the score for X_test and test_labels was 0.7533333333333333 .
- Therefore, we can state that this is best decision tree model for this dataset.

Feature Importance For Decision Tree

We plotted a bar graph showing the feature importance of the all variables and in addition we drawed out values of each feature importance for cross verification.

Feature Importance of Over-Fitted Decision Tree Model :

- At first, we did this for the over-fitted decision tree model we build at the beginning. The results were as follows :

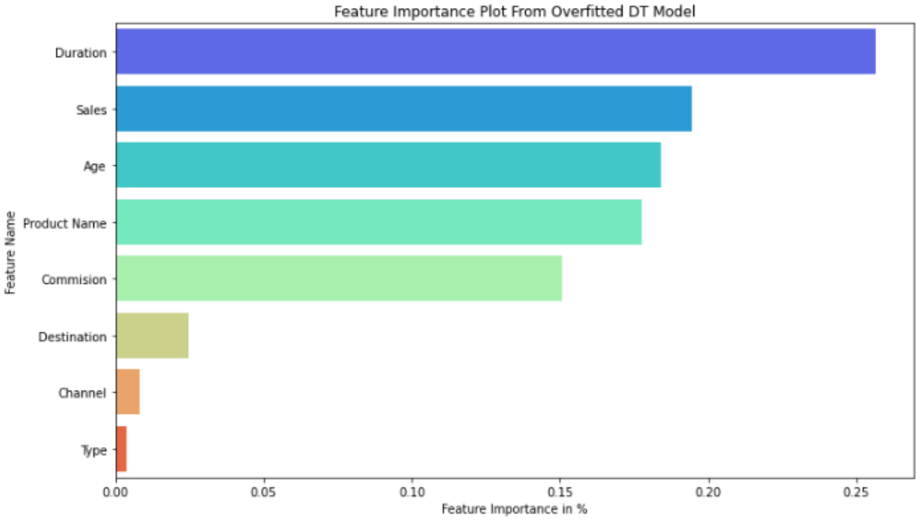
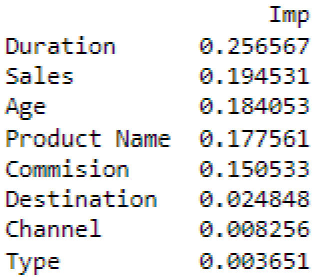


Figure 64 : Feature Importance Bar Plot (Over-Fitted Decision Tree Model)

And the values of Feature Importance of Over-Fitted Decision Tree model were as follows :



```
                   Imp
Duration      0.256567
Sales         0.194531
Age           0.184053
Product Name  0.177561
Commision     0.150533
Destination   0.024848
Channel       0.008256
Type          0.003651
```

Figure 65 : Values of Feature Importance (Over-Fitted Decision Tree Model)

Interpretations :

- The Graphical Bar Plot depicted the feature importance for decision tree for **dt_model** which we build at the beginning which was over-fitted.
- According to this bar plot, we interpreted that most important feature was 'Duration' at the top and just next to it was 'Sales' feature.
- We also observed that the least important feature among all was 'Type' and after that comes 'Channel' feature.

Feature Importance of Regularized Decision Tree Model :

- In this, we did plot a bar graph for feature importance for the regularized/pruned decision tree model we build . The results were as follows :

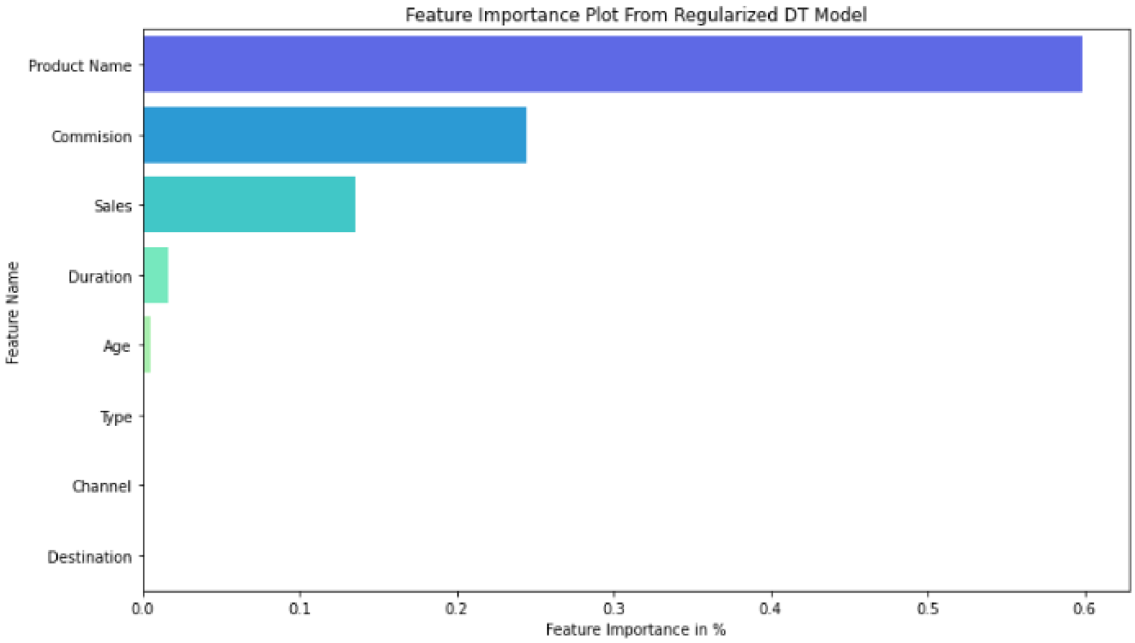
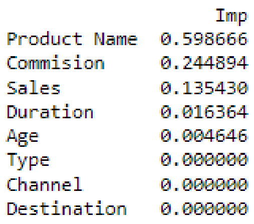


Figure 66 : Feature Importance Bar Plot (Regularized Decision Tree Model)

And the values of Feature Importance of Regularized Decision Tree model were as follows :



```
                  Imp
Product Name  0.598666
Commision     0.244894
Sales         0.135430
Duration      0.016364
Age           0.004646
Type          0.000000
Channel       0.000000
Destination   0.000000
```

Figure 67 : Values For Feature Importance (Regularized Decision Tree Model)

Interpretations :

- The Graphical Bar Plot depicted the feature importance for regularized decision tree for **reg_dt_model1** which we build using specific parameters for better understanding.
- We interpreted that most important feature was 'Product Name' at the top and just next to it was 'Commision' feature.
- We also observed that three features that are 'Type', 'Channel', 'Destination' were having 0 as value in feature importance function which means all three were the least important feature.

Feature Importance of Best Decision Tree Model :

- In this, we did plot a bar graph for feature importance for the best decision tree model we build . The results were as follows :

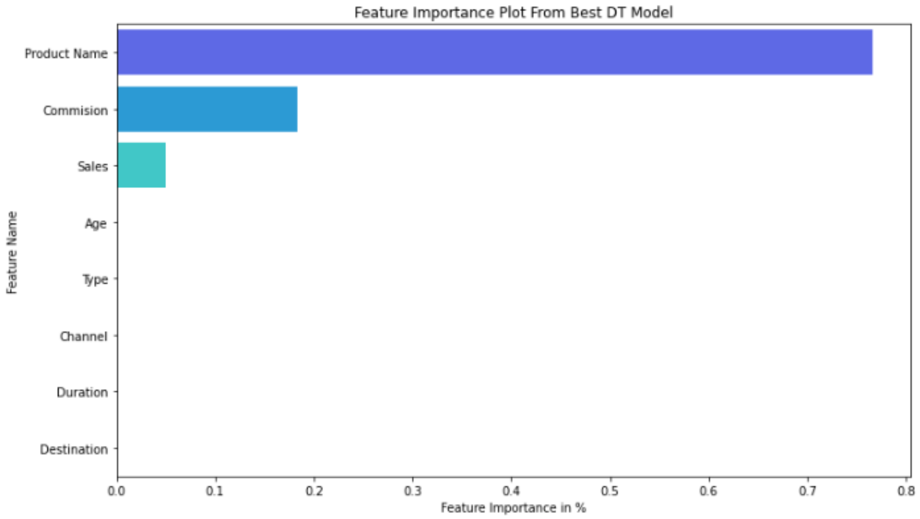
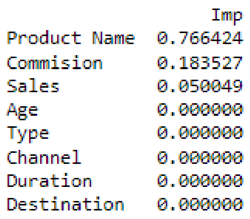


Figure 68 : Feature Importance Bar Plot (Best Decision Tree Model)

And the values of Feature Importance of Best Decision Tree model were as follows :



```
                  Imp
Product Name  0.766424
Commision     0.183527
Sales         0.050049
Age           0.000000
Type          0.000000
Channel       0.000000
Duration      0.000000
Destination   0.000000
```

Figure 69 : Values For Feature Importance (Best Decision Tree Model)

Interpretations :

- The Graphical Bar Plot showed the Feature Importance of Best Decision Tree for best_dtree_model which we build using GridSearch_CV function in order to get the best decision tree model.
- We interpreted that only 3 values are present in this Feature Importance of Best Decision Tree and rest of the features had 0 as value in the feature importance. It means that only 3 features that were 'Product Name' , 'Commision', 'Sales' were the only ones which were considered by the Feature Importance for Best Decision Tree.
- We also observed that except those 3, all others were considered to be least important in the feature importance for Best Decision Tree.

Building classification model - Random Forest

We installed necessary libraries and packages such as RandomForestClassifier in order to build a random forest.

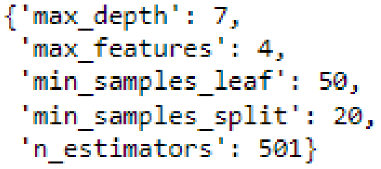
- In the beginning, we used RandomForestClassifier with n_estimators 100, max_samples 9, and fit it on the training data. This random forest was named as **rfcl**.
- After that , we checked the score for the same random forest.
- The score for X_train and train_labels was 0.99380952
- And the score for X_test and test_labels was 74.888889

Interpretations :

- We inferred that, the random forest model is clear case of imbalance/over-fitting. Thus, we need to balance it.
- For that purpose, we used the GridSearch_CV function in order to get best model for Random Forest.

- This time we created a new dataframe called param_grid and we put certain parameters inside it in order to get the best model of random forest named as **rfcl1**
- Adding on, using grid search this rfcl1, was passed with GridSearch_Cv and fit that grid search model.
- We then fitted the same random forest and checked for best_params.

- Furthermore, using best_estimator parameter, we then builded the best random forest model called as **best_grid**. They were as follows :



```
{'max_depth': 7,
 'max_features': 4,
 'min_samples_leaf': 50,
 'min_samples_split': 20,
 'n_estimators': 501}
```

Figure 70 : best_params(random forest)

- Then we checked for the scores for the builded random forest.

- The score for X_train and train_labels was 0.789047619047619 .
- And the score for X_test and test_labels was 0.7466666666666667 .
- These scores clarified that the new builded random forest (best_grid) was the best random forest.

Feature Importance For Random Forest :

We build the bar plot for finding the Feature Importance for the following variables for the random forest model that was builded at the very first named **rfcl**. It was as follows :

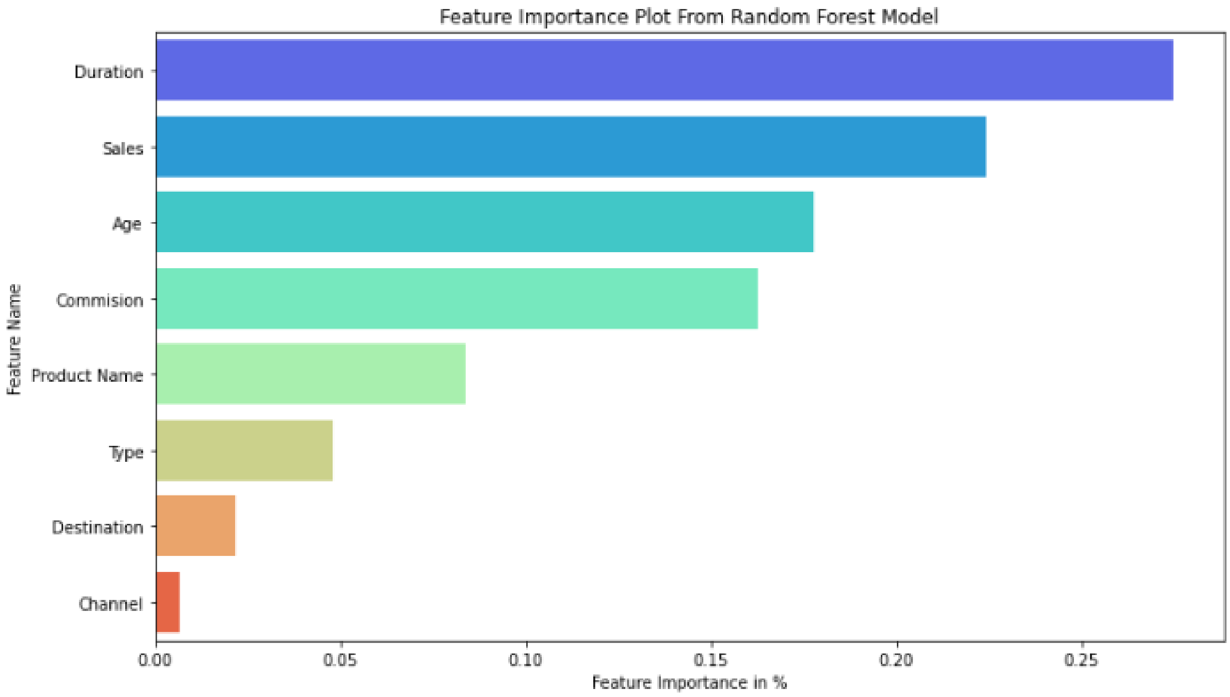


Figure 71 : Feature Importance Bar Plot(Random Forest)

Interpretation

- Duration had the highest feature importance value as per the bar plot while Sales was just next to it.
- Channel had the lowest feature importance value.

Feature Importance For Best Random Forest :

We build the bar plot for finding the Feature Importance for the following variables for the best random forest model that was builded using GridSearch_CV named **best_grid**. It was as follows :

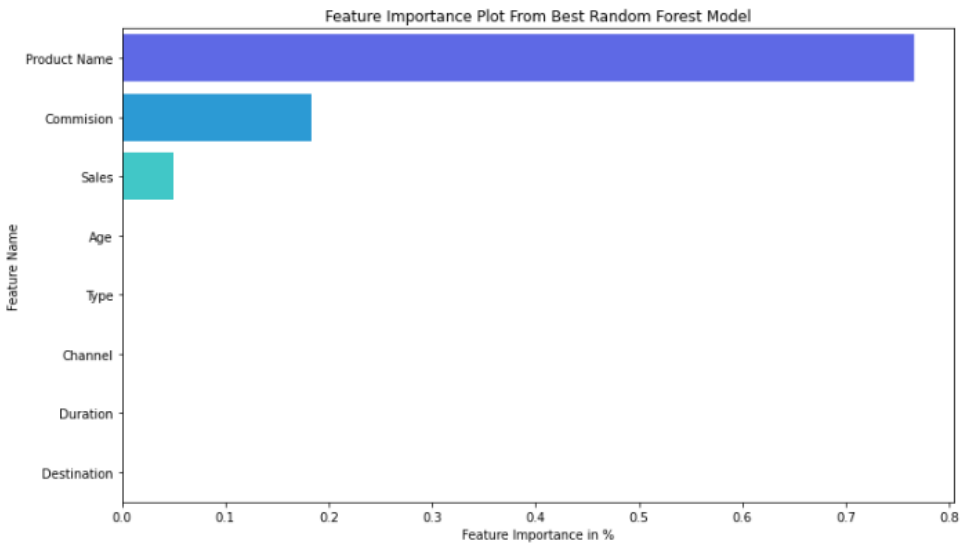


Figure 72 : Feature Importance Bar Plot (Best Random Forest)

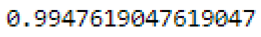
Interpretations :

- From the best random forest model(**best_grid**) we found that Product Name had the highest feature importance value.
- All variables had zero value except for Product Name, Commision, and Sales.

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

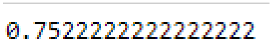
After build both Decision Tree and Random Forest , we then checked the performance metrics using the predict function().

Predicting Train and Test data with the First RF Model :

The Train Accuracy for this RF model was 

```
0.9947619047619047
```

.

The Test Accuracy for this RF model was 

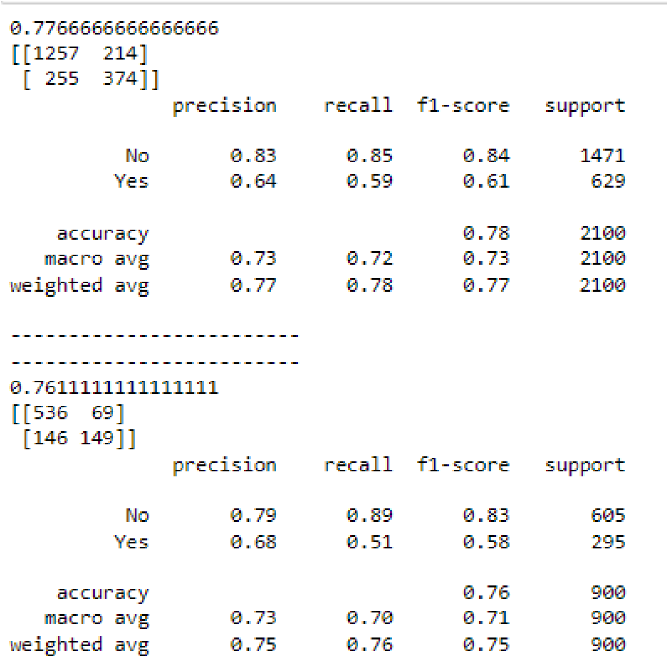
```
0.7522222222222222
```

.

As per the results, we could say that the train and test accuracy for random forest were not similar.

Classification Report For Regularized Decision Tree :

The classification report for **reg_dt_model1** (pruned decision tree) was as follows :



```
0.7766666666666666
[[1257  214]
 [ 255  374]]
              precision    recall  f1-score   support

          No       0.83      0.85      0.84      1471
         Yes       0.64      0.59      0.61       629

    accuracy                           0.78      2100
   macro avg       0.73      0.72      0.73      2100
weighted avg       0.77      0.78      0.77      2100

-------------------------
-------------------------
0.7611111111111111
[[536  69]
 [146 149]]
              precision    recall  f1-score   support

          No       0.79      0.89      0.83       605
         Yes       0.68      0.51      0.58       295

    accuracy                           0.76       900
   macro avg       0.73      0.70      0.71       900
weighted avg       0.75      0.76      0.75       900
```

Figure 73 : Classification Report(Regularized Decision Tree)

Interpretations :

- At the top , it showed that Train Accuracy Score for this pruned decision tree model with was equal to 77.66%. The rest of the classification report tells us the precision, recall and f1score and support value for the same train set.
- At the middle , it showed that Test Accuracy Score for this pruned decision tree model with was equal to 76.11%. The rest of the classification report tells us the precision, recall and f1score and support value for the same train set.

Confusion Matrix For Regularized Decision Tree :

The confusion matrix for the train set was as follows :

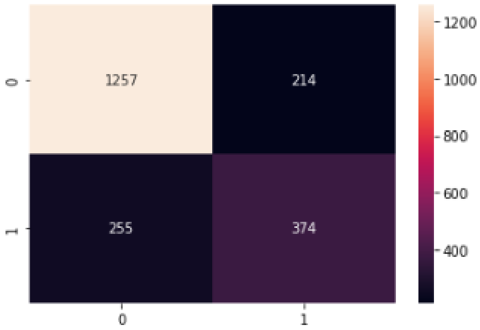


Figure 74 : Confusion Matrix (Train Set - Regularized Decision Tree)

Interpretations :

- Visual Representation Of Confusion Matrix for Train Data for Decision Tree was shown here.
- 1257 were the people who did not claimed the insurance while 374 were the people who claimed the insurance.

The confusion matrix for the test set was as follows :

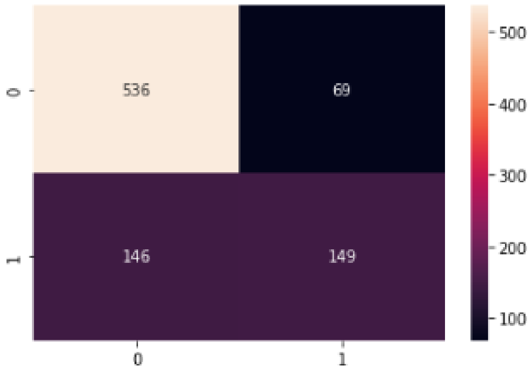


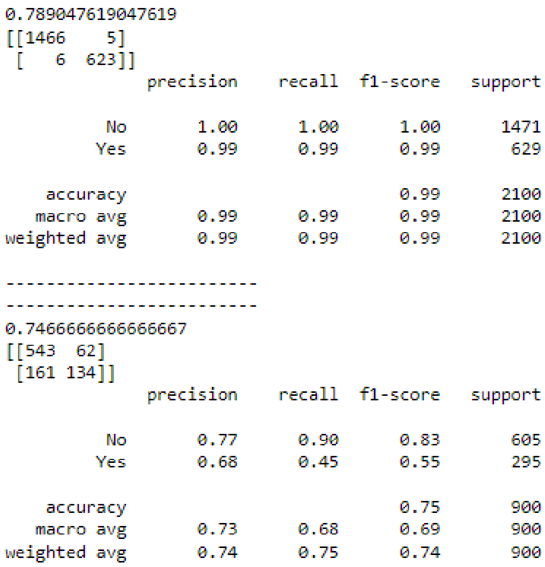
Figure 75 : Confusion Matrix (Test Set - Regularized Decision Tree)

Interpretations :

- Visual Representation Of Confusion Matrix for Test Data for Decision Tree was shown here.
- 536 were the people who did not claimed the insurance while 149 were the people who claimed the insurance.

Classification Report For Random Forest :

The classification report for **best_grid** (pruned decision tree) was as follows :



```
0.789047619047619
[[1466    5]
 [   6  623]]
              precision    recall  f1-score   support

          No       1.00      1.00      1.00      1471
         Yes       0.99      0.99      0.99       629

    accuracy                           0.99      2100
   macro avg       0.99      0.99      0.99      2100
weighted avg       0.99      0.99      0.99      2100

-------------------------
-------------------------
0.7466666666666667
[[543  62]
 [161 134]]
              precision    recall  f1-score   support

          No       0.77      0.90      0.83       605
         Yes       0.68      0.45      0.55       295

    accuracy                           0.75       900
   macro avg       0.73      0.68      0.69       900
weighted avg       0.74      0.75      0.74       900
```

Figure 76 : Classification Report (Best Random Forest)

Interpretations :

- At the top , it showed that Train Accuracy Score for this **best random forest model** with was equal to 78.90%. The rest of the classification report tells us the precision, recall and f1score and support value for the same train set.
- At the middle , it showed that Test Accuracy Score for this **best random forest model** with was equal to 74.67%. The rest of the classification report tells us the precision, recall and f1score and support value for the same train set.

Confusion Matrix For Best Random Forest :

- The confusion matrix for the train set was as follows :

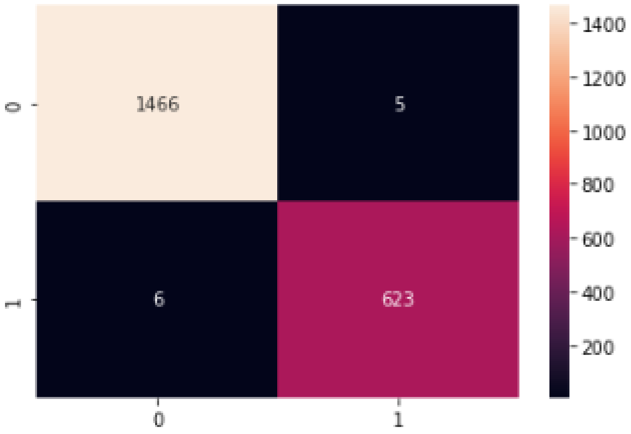


Figure 77 : Confusion Matrix (Train Set) : Best Random Forest

Interpretations :

- Visual Representation Of Confusion Matrix for Train Data for Best Random Forest was shown here.
- 1466 were the people who did not claimed the insurance while 623 were the people who claimed the insurance.

- The confusion matrix for the test set was as follows :

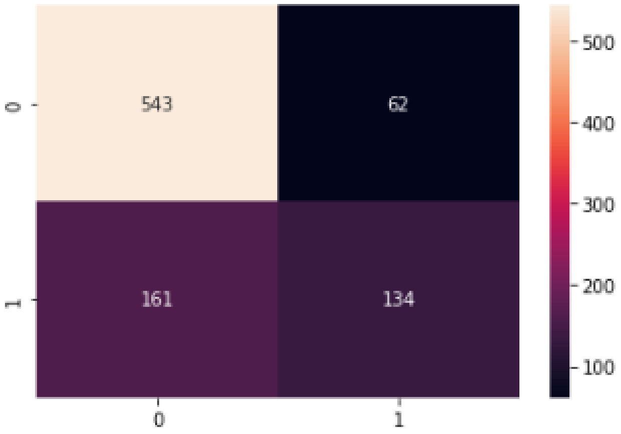


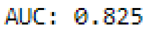
Figure 78 : Confusion Matrix (Test Set) - Best Random Forest

Interpretations :

- Visual Representation Of Confusion Matrix for Test Data for Best Random Forest was shown here.
- 543 were the people who did not claimed the insurance while 134 were the people who claimed the insurance.

ROC_AUC Score & ROC Curve :

We imported roc_auc and roc_curve from sklearn.metrics for finding roc_auc score and roc_curve.



The AUC score for the Regularized Decision Tree was `AUC: 0.825` .

The ROC Curve for the same was as follows :

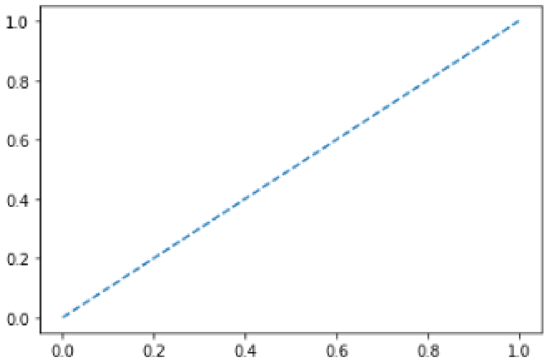
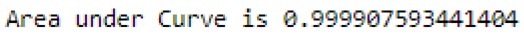


Figure 79 : ROC Curve(Regularized Decision Tree)

Conclusion :

After seeing the output , we concluded that our regularized decision tree model was build very good and majority of the training and test data gave same score when the model was run.



The AUC score for the Best Random Forest was `Area under Curve is 0.999907593441404` .

The ROC Curve for the same was as follows :

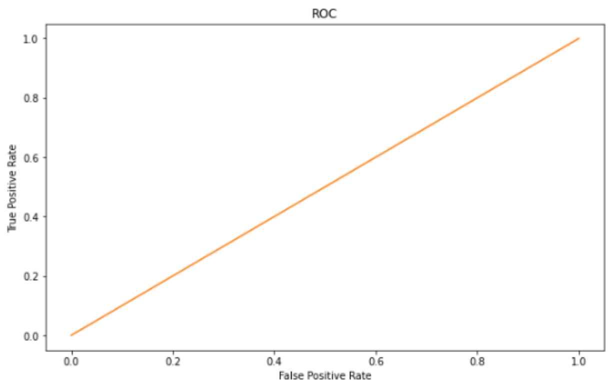


Figure 80 : ROC Curve (Best Random Forest)

Conclusion :

After seeing the output , we concluded that our regularized decision tree model was build very good with Area under curve = 99 % and majority of the training and test data gave same score when the model was run.

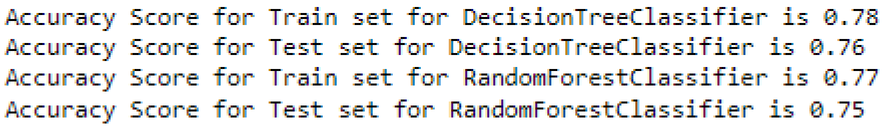
Problem 2.4 Final Model: Compare all the models and write an inference which model is best/optimized.

Firstly, we compared Accuracies from all the models for Train and Test Sets for Regularized Decision Tree

Comparing Accuracies from all the models for Train and Test Sets for Regularized Decision Tree & Random Forest.

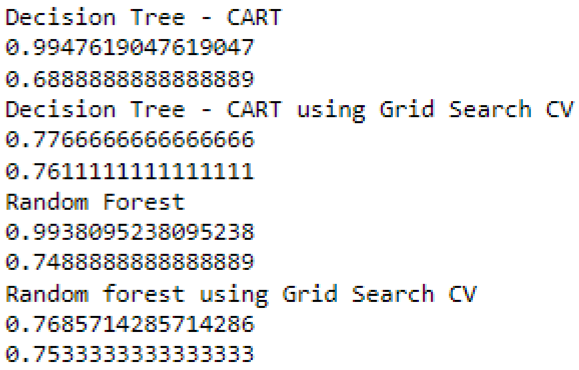
We used for loop for making comparing accuracy for all the models for train and test data for Regularized Decision Tree as well as for Random Forest.

The output was as follows :



```
Accuracy Score for Train set for DecisionTreeClassifier is 0.78
Accuracy Score for Test set for DecisionTreeClassifier is 0.76
Accuracy Score for Train set for RandomForestClassifier is 0.77
Accuracy Score for Test set for RandomForestClassifier is 0.75
```

Figure 81 :Accuracy Score for all models



```
Decision Tree - CART
0.9947619047619047
0.6888888888888889
Decision Tree - CART using Grid Search CV
0.7766666666666666
0.7611111111111111
Random Forest
0.9938095238095238
0.7488888888888889
Random forest using Grid Search CV
0.7685714285714286
0.7533333333333333
```

Figure 82 : Accuracy Score for all models (1)

Conclusion :

- From this we got all accuracy scores for all models.
- We inferred that, the most best and optimized model for this data(train and test) was for the Best Random Forest model.

Comparing Confusion Matrices from All the models for the Train Set :

The comparison for both Regularized Decision Tree & Best Random Forest for the train set was as follows :

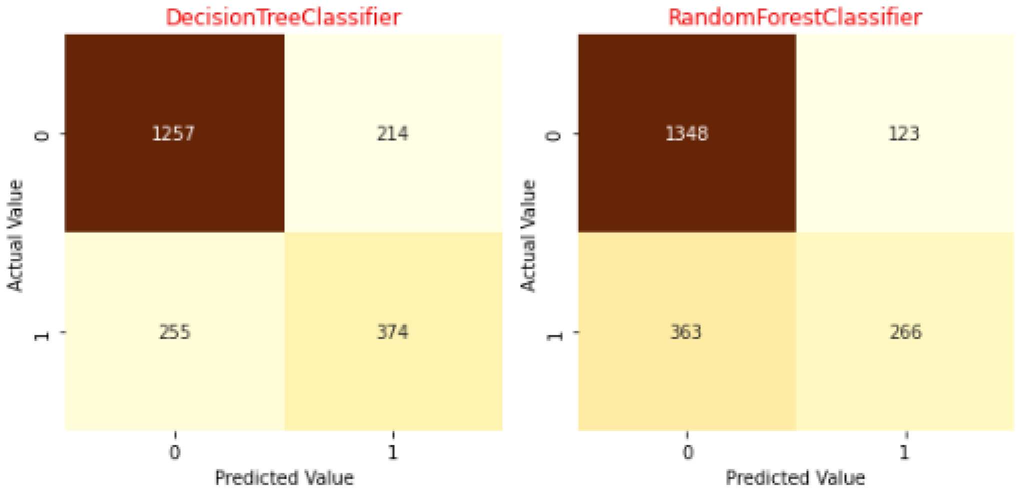


Figure 83 : Comparing Confusion Matrix for all models(Train Set)

Interpretations :

- By comparing both matrices, we concluded that, we can see values of True Negative for both with Best Decision Tree having value 1257 and Best Random Forest having value 1348.
- Also for the True Positive Value for both were Regularized Decision Tree had 374 value and Best Random Forest had 266 value.
- This means that Best Random Forest gave a more accurate output as compared to Regularized Decision Tree.

Comparing Confusion Matrices from All the models for the Test Set :

The comparison for both Regularized Decision Tree & Best Random Forest for the test set was as follows :

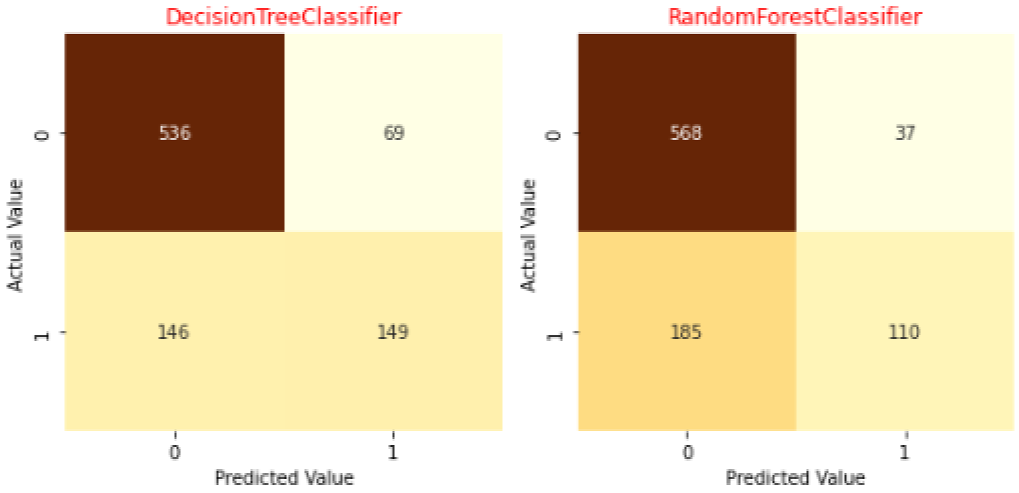


Figure 84 : Comparing Confusion Matrices for all models(Test Data)

Interpretations :

- By comparing both matrices, we concluded that, we can see values of True Negative for both with Best Decision Tree having value 536 and Best Random Forest having value 568.
- Also for the True Positive Value for both were Regularized Decision Tree had 149 value and Best Random Forest had 110 value.
- This means that Best Random Forest gave a more accurate output as compared to Regularized Decision Tree.

Conclusion :

- After reading and observing all outputs and interpretations, we concluded that, Best Random Forest is best optimized according to the insights and interpretations.

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

From all the insights we got using different functions on the data set, we can give recommendations as follows :

- The Random Forest is a better option for finding the best model for business problems.
- Product Name was the best feature importance and company should work on that in order to grow their productions.
- After that , they must find better ways like giving offers and discounts in order to attract customers and make new customers which will help in business growth.