greatlearning
*Learning for Life*


# BUSINESS REPORT

# PROBLEM-1

# CUSTOMER SEGMENTATION TO GIVE PROMOTIONAL OFFERS TO ITS CUSTOMERS FOR A LEADING BANK.

# PROBLEM-2

# SUPERVISED MODEL BUILDING TO PREDICT THE INSURANCE CLAIM STATUS.

PREPARED BY - RUPESH KUMAR
PROGRAM- PGP-DSBA ONLINE
BATCH - A5 DSBA 21C
SUBMISSION DATE: 29/08/2021

# CONTENTS

# CONTENTS

# CONTENTS

# CONTENTS

# CONTENTS

# CONTENTS

# CONTENTS



## LIST OF FIGURES

### FIGURES OF PROBLEM 1 DATASET

# LIST OF FIGURES

# LIST OF FIGURES

# LIST OF FIGURES



# LIST OF TABLES

# LIST OF TABLES

# LIST OF TABLES

# LIST OF TABLES

# LIST OF TABLES

## Problem Statement 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.

## Executive Summary

A Bank deals with different types of customers. The dataset consists of various characteristics of the customers based on the behaviour for usage of credit card available in the records of the bank. Based on the different attributes/characteristics the customers of the bank is defined. In this problem statement we will explore the different activities of users during the past few months and recognise their pattern and segments them on the credit card usage to give promotional offers to its customers.

## Introduction

The purpose of this whole exercise is to explore the dataset. Do the exploratory data analysis & apply various clustering techniques to segments the customers on their credit card usage. Explore the dataset using central tendency and other parameters. The data consists of 210 different customers with 7 unique activities.. Analyse the different attributes of the customers which can help in analysing the behaviour of the customers on their credit card usage. This assignment should help the bank in exploring the summary statistics, clustering model will help bank to recognise the activities of users during the past few months. and identify the segments based on credit card usage to give promotional offers to its customers.

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

EDA - Data Description ,data preprocessing,Data Visualization

## Records of the Dataset

| | spending | advance_payments | probability_of_full_payment | current_balance | credit_limit | min_payment_amt | max_spent_in_single_shopping |
|---|---|---|---|---|---|---|---|
| 0 | 19.94 | 16.92 | 0.8752 | 6.675 | 3.763 | 3.252 | 6.550 |
| 1 | 15.99 | 14.89 | 0.9064 | 5.363 | 3.582 | 3.336 | 5.144 |
| 2 | 18.95 | 16.42 | 0.8829 | 6.248 | 3.755 | 3.368 | 6.148 |
| 3 | 10.83 | 12.96 | 0.8099 | 5.278 | 2.641 | 5.182 | 5.185 |
| 4 | 17.99 | 15.86 | 0.8992 | 5.890 | 3.694 | 2.068 | 5.837 |
| 5 | 12.70 | 13.41 | 0.8874 | 5.183 | 3.091 | 8.456 | 5.000 |
| 6 | 12.02 | 13.33 | 0.8503 | 5.350 | 2.810 | 4.271 | 5.308 |
| 7 | 13.74 | 14.05 | 0.8744 | 5.482 | 3.114 | 2.932 | 4.825 |
| 8 | 18.17 | 16.26 | 0.8637 | 6.271 | 3.512 | 2.853 | 6.273 |
| 9 | 11.23 | 12.88 | 0.8511 | 5.140 | 2.795 | 4.325 | 5.003 |

TAB:1 RECORDS OF THE DATASET

## Data Dictionary for Problem Statment 1

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)

TAB:2 DATA DICTIONARY

## Summary of the Dataset

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

TAB:3 SUMMARY OF THE DATASET.

- From the above table we can infer the count,mean, std , 25% , 50% ,75% and min & max values of the all numeric variables present in the dataset.here is no bad values found in the dataset too.

## Shape of the dataset

| No. of Rows | No. of cols |
|---|---|
| 210 | **7** |

TAB:4 SHAPE OF THE DATAFRAME

- Shape attribute tells us number of observations and variables we have in the data set. It is used to check the dimension of data. The bank_marketing_Data.csv data set has 210 observations (rows) and 7 variables (columns) in the dataset.

## Appropriateness of Datatypes & Information of the Dataframe.

```
<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
```

TAB:5 APPROPRIATENESS OF DATATYPES
& INFORMATION OF THE DATAFRAME

- From the above results we can see that there is no missing value present in the dataset.Their are total 210 rows & 7 columns in this dataset,indexed from 0 to 209. All Datatypes of variables are float64. Memory used by the dataset: 11.6 KB.

## Checking for Null Values

```
spending                          False
advance_payments                  False
probability_of_full_payment       False
current_balance                   False
credit_limit                      False
min_payment_amt                   False
max_spent_in_single_shopping      False
dtype: bool
```

TAB:6 NULL VALUES.

- No Null Values present in the dataset.

## Checking for Anomalies in the Dataset.

```
array([19.94, 15.99, 18.95, 10.83, 17.99, 12.7 , 12.02, 13.74, 18.17,
       11.23, 18.55, 14.09, 12.15, 18.98, 12.1 , 12.79, 16.14, 10.8 ,
       13.22, 12.37, 13.07, 17.98, 12.62, 15.11, 15.56, 12.78, 11.02,
       11.35, 13.78, 11.84, 12.55, 15.88, 11.82, 11.19, 11.14, 12.22,
       11.81, 19.51, 18.72, 13.84, 16.87, 20.03, 10.79, 11.18, 13.16,
       19.06, 18.96, 18.83, 12.73, 19.46, 19.38, 18.81, 16.23, 12.38,
       11.83, 10.93, 18.65, 14.79, 11.41, 11.27, 15.26, 14.34, 18.85,
       20.71, 14.11, 19.15, 12.19, 13.54, 12.49, 20.1 , 20.2 , 13.34,
       18.94, 15.03, 12.13, 16.82, 14.29, 14.52, 12.88, 13.94, 18.59,
       10.91, 14.49, 16.63, 15.38, 16.17, 13.2 , 13.99, 21.18, 11.87,
       18.43, 19.57, 16.16, 10.82, 17.63, 13.37, 19.31, 18.89, 15.69,
       18.36, 13.32, 12.8 , 18.75, 15.6 , 14.33, 20.24, 12.89, 11.21,
       17.32, 13.5 , 14.28, 11.48, 20.97, 12.08, 11.56, 12.46, 12.54,
       15.01, 18.3 , 11.4 , 14.46, 11.36, 14.86, 12.36, 12.05, 19.14,
       17.55, 14.59, 15.78, 14.92, 11.24, 11.34, 12.74, 19.13, 10.74,
       13.8 , 12.44, 14.16, 12.11, 14.99, 16.2 , 11.42, 14.7 , 13.02,
       11.26, 17.36, 20.88, 12.72, 18.88, 17.26, 18.27, 11.65, 15.5 ,
       15.05, 12.76, 11.43, 16.19, 11.49, 14.38, 18.45, 20.16, 19.11,
       14.69, 12.21, 16.44, 10.59, 14.37, 13.45, 12.67, 14.01, 17.12,
       16.84, 15.49, 11.75, 14.43, 12.26, 18.14, 15.36, 16.53, 18.76,
       12.3 , 19.18, 12.01, 14.88, 17.08, 14.8 , 11.55, 16.41, 13.89,
       16.77, 14.03, 16.12, 15.57])
```

### Spending

```
array([16.92, 14.89, 16.42, 12.96, 15.86, 13.41, 13.33, 14.05, 16.26,
       12.88, 16.22, 14.41, 13.45, 16.57, 13.15, 13.53, 14.99, 12.57,
       13.84, 13.71, 13.47, 13.92, 15.85, 13.67, 14.54, 13.57, 13.  ,
       13.12, 12.82, 14.06, 13.21, 14.9 , 13.4 , 13.05, 12.79, 13.32,
       16.71, 16.34, 13.94, 15.65, 16.9 , 12.93, 12.72, 13.82, 16.45,
       16.2 , 16.29, 13.75, 16.19, 16.5 , 16.72, 15.18, 13.44, 13.23,
       12.8 , 16.41, 14.52, 12.95, 12.97, 14.85, 14.37, 16.17, 17.23,
       14.1 , 13.2 , 13.85, 13.46, 16.99, 16.89, 13.95, 16.32, 14.77,
       13.73, 15.51, 14.09, 14.6 , 13.5 , 14.17, 16.05, 14.61, 15.46,
       15.38, 13.66, 13.83, 17.21, 14.84, 13.04, 13.02, 15.97, 16.74,
       15.33, 12.83, 13.78, 16.59, 14.66, 16.23, 14.75, 16.49, 16.52,
       16.18, 15.11, 14.28, 16.91, 13.77, 13.13, 15.91, 17.25, 13.31,
       14.26, 14.76, 15.89, 13.08, 14.18, 14.35, 14.67, 13.19, 16.66,
       15.98, 16.61, 12.86, 15.66, 14.91, 14.43, 12.87, 13.36, 16.31,
       12.73, 14.04, 13.59, 14.4 , 13.27, 14.56, 15.27, 14.21, 13.76,
       13.01, 15.76, 17.05, 15.73, 16.09, 13.07, 14.86, 14.68, 13.38,
       15.16, 13.22, 16.12, 17.03, 14.49, 15.25, 12.41, 14.39, 14.02,
       14.29, 15.55, 15.67, 14.94, 13.55, 13.52, 12.63, 13.6 , 15.34,
       13.34, 16.63, 14.57, 13.1 , 15.62, 14.16, 15.  , 15.15])
```

### advance_payments

```
array([0.8752, 0.9064, 0.8829, 0.8099, 0.8992, 0.8874, 0.8503, 0.8744,
       0.8637, 0.8511, 0.8865, 0.8529, 0.8443, 0.8687, 0.8793, 0.8786,
       0.9034, 0.859 , 0.868 , 0.8491, 0.8567, 0.848 , 0.8993, 0.8481,
       0.8986, 0.8823, 0.8716, 0.8189, 0.8291, 0.8594, 0.8759, 0.8521,
       0.8558, 0.8988, 0.8274, 0.8253, 0.8652, 0.8198, 0.878 , 0.881 ,
       0.8955, 0.8648, 0.8811, 0.8107, 0.8662, 0.8854, 0.9077, 0.8917,
       0.8458, 0.8977, 0.8985, 0.8906, 0.885 , 0.8609, 0.8496, 0.839 ,
       0.8698, 0.8819, 0.856 , 0.8419, 0.8696, 0.8726, 0.9056, 0.8763,
       0.8911, 0.889 , 0.8783, 0.8871, 0.8658, 0.8746, 0.8894, 0.862 ,
       0.8942, 0.8081, 0.905 , 0.8557, 0.8879, 0.8728, 0.9066, 0.8372,
       0.8538, 0.8747, 0.8706, 0.8588, 0.8883, 0.9183, 0.8989, 0.871 ,
       0.8266, 0.8795, 0.8779, 0.8644, 0.8256, 0.88  , 0.8849, 0.8815,
       0.899 , 0.9008, 0.9058, 0.875 , 0.8452, 0.8613, 0.886 , 0.8999,
       0.858 , 0.8831, 0.8897, 0.8541, 0.8167, 0.8599, 0.8852, 0.8944,
       0.8473, 0.8859, 0.8664, 0.8425, 0.8722, 0.8657, 0.9108, 0.8375,
       0.882 , 0.8818, 0.8382, 0.8676, 0.8923, 0.8416, 0.8673, 0.8563,
       0.8991, 0.9006, 0.8359, 0.8596, 0.8564, 0.8579, 0.9035, 0.8329,
       0.8794, 0.8462, 0.8584, 0.8639, 0.8734, 0.8683, 0.9153, 0.8641,
       0.8355, 0.8857, 0.8785, 0.9031, 0.8686, 0.8969, 0.887 , 0.8575,
       0.8964, 0.8335, 0.8263, 0.8951, 0.8921, 0.8735, 0.9081, 0.8799,
       0.8453, 0.888 , 0.8604, 0.8625, 0.8892, 0.8623, 0.8392, 0.8724,
       0.9009, 0.8082, 0.884 , 0.8751, 0.8333, 0.8772, 0.8861, 0.8984,
       0.8684, 0.8717, 0.8249, 0.9079, 0.8455, 0.8866, 0.8638, 0.8796,
       0.9   , 0.8527])
```

### probability_of_full_payment

```
array([6.675, 5.363, 6.248, 5.278, 5.89 , 5.183, 5.35 , 5.482, 6.271,
       5.14 , 6.153, 5.717, 5.417, 6.449, 5.105, 5.224, 5.658, 4.981,
       5.395, 5.386, 5.204, 5.472, 5.979, 5.41 , 5.579, 5.776, 5.262,
       5.325, 5.176, 5.089, 5.479, 5.175, 5.333, 5.618, 5.314, 5.25 ,
       5.011, 5.413, 6.366, 6.219, 5.324, 6.139, 6.493, 5.317, 5.009,
       5.454, 6.416, 6.051, 6.037, 5.412, 6.006, 6.113, 6.303, 6.272,
       5.872, 5.219, 5.263, 5.046, 6.285, 5.545, 5.09 , 5.088, 5.714,
       5.63 , 6.152, 6.579, 5.42 , 6.245, 5.137, 5.348, 5.267, 6.581,
       5.389, 6.144, 5.702, 5.394, 6.017, 5.291, 5.741, 5.139, 5.585,
       5.715, 6.053, 5.884, 5.762, 5.236, 5.119, 6.573, 5.763, 5.22 ,
       5.132, 5.98 , 6.384, 5.845, 5.18 , 6.033, 5.32 , 6.341, 5.477,
       6.227, 5.527, 6.445, 6.666, 5.541, 5.16 , 6.111, 5.832, 5.504,
       6.315, 5.495, 5.279, 6.064, 5.351, 5.397, 6.563, 5.099, 5.451,
       5.52 , 5.789, 5.136, 5.388, 5.678, 5.076, 6.549, 6.191, 6.259,
       5.091, 5.791, 5.674, 5.384, 5.053, 5.24 , 6.183, 5.145, 5.376,
       5.319, 5.57 , 5.826, 5.008, 5.205, 5.186, 5.662, 6.145, 6.45 ,
       5.226, 6.084, 5.978, 6.173, 5.108, 5.877, 5.712, 5.073, 5.833,
       5.304, 6.107, 6.513, 6.154, 5.563, 5.357, 4.899, 5.569, 5.516,
       4.984, 5.609, 5.85 , 5.998, 5.159, 5.757, 5.138, 5.444, 4.902,
       5.408, 6.059, 5.701, 5.875, 6.172, 5.243, 6.369, 5.405, 5.554,
       5.656, 5.167, 5.718, 5.439, 5.927, 5.438, 5.709, 5.92 ])
```

### current_balance

```
array([3.763, 3.582, 3.755, 2.641, 3.694, 3.091, 2.81 , 3.114, 3.512,
       2.795, 3.674, 3.186, 2.837, 3.552, 2.941, 3.054, 3.562, 2.821,
       3.07 , 2.911, 2.96 , 2.994, 3.687, 3.462, 3.408, 3.026, 2.701,
       2.668, 3.156, 2.836, 2.968, 3.507, 2.777, 2.675, 2.794, 2.967,
       2.716, 3.801, 3.684, 3.379, 3.463, 3.857, 2.648, 2.975, 3.719,
       3.897, 3.786, 2.882, 3.892, 3.791, 3.693, 3.472, 2.989, 2.84 ,
       2.717, 3.594, 3.291, 2.775, 2.763, 3.242, 3.19 , 3.806, 3.814,
       3.302, 3.815, 2.981, 3.785, 3.864, 3.074, 3.825, 3.212, 2.745,
       3.486, 3.337, 3.113, 3.119, 3.15 , 3.86 , 3.465, 3.268, 3.387,
       3.232, 3.383, 4.033, 3.312, 2.693, 2.953, 3.771, 3.772, 3.395,
       2.63 , 3.573, 3.128, 3.81 , 3.769, 3.514, 3.639, 3.485, 3.073,
       3.126, 3.869, 3.286, 3.199, 3.962, 2.687, 3.403, 3.158, 3.298,
       2.758, 3.991, 2.936, 2.683, 3.017, 2.879, 3.168, 3.245, 3.221,
       3.377, 2.755, 3.258, 3.042, 3.67 , 2.847, 3.561, 3.737, 2.804,
       3.69 , 3.333, 3.434, 3.412, 2.715, 2.849, 2.956, 2.909, 3.902,
       2.642, 3.155, 2.897, 3.129, 3.464, 2.85 , 3.466, 2.71 , 3.419,
       3.574, 4.032, 3.049, 3.764, 3.651, 3.396, 3.328, 2.719, 3.421,
       2.695, 3.773, 3.93 , 3.259, 2.893, 3.505, 2.787, 3.153, 3.065,
       3.135, 3.566, 3.484, 3.032, 3.371, 3.201, 2.678, 3.272, 2.833,
       3.563, 3.393, 3.467, 3.796, 2.974, 3.681, 2.776, 3.683, 3.288,
       2.845, 3.525, 3.438, 3.231])
```

credit_limit

```
array([3.252 , 3.336 , 3.368 , 5.182 , 2.068 , 8.456 , 4.271 , 2.932 ,
       2.853 , 4.325 , 1.738 , 3.92  , 3.638 , 2.144 , 2.201 , 5.483 ,
       1.355 , 4.773 , 4.157 , 3.26  , 3.919 , 5.304 , 2.257 , 3.306 ,
       3.128 , 4.972 , 1.176 , 6.735 , 4.337 , 7.524 , 3.136 , 3.598 ,
       4.419 , 0.7651, 4.471 , 5.813 , 6.388 , 5.469 , 4.898 , 2.962 ,
       2.188 , 2.259 , 3.696 , 3.063 , 5.462 , 4.051 , 0.8551, 2.248 ,
       4.334 , 2.553 , 3.533 , 5.324 , 4.308 , 3.678 , 3.237 , 3.769 ,
       5.472 , 5.195 , 5.398 , 4.391 , 2.704 , 4.957 , 4.309 , 4.543 ,
       1.313 , 2.843 , 4.451 , 2.7   , 3.084 , 3.631 , 2.587 , 4.421 ,
       1.955 , 5.173 , 5.995 , 2.908 , 1.933 , 4.825 , 4.004 , 2.699 ,
       1.481 , 2.352 , 2.124 , 6.001 , 4.179 , 4.116 , 2.04  , 4.462 ,
       4.286 , 8.315 , 5.234 , 5.78  , 2.221 , 3.332 , 3.597 , 2.984 ,
       1.472 , 4.266 , 4.853 , 3.747 , 4.67  , 3.477 , 3.6   , 3.639 ,
       1.599 , 5.064 , 4.933 , 7.035 , 4.873 , 4.188 , 2.725 , 3.328 ,
       5.901 , 6.185 , 6.169 , 3.824 , 2.249 , 6.685 , 5.876 , 4.677 ,
       1.415 , 4.062 , 4.987 , 3.082 , 2.688 , 1.791 , 2.837 , 5.588 ,
       2.754 , 2.802 , 4.048 , 2.129 , 3.22  , 3.691 , 4.988 , 4.076 ,
       6.682 , 3.985 , 5.366 , 4.185 , 5.593 , 1.142 , 3.521 , 3.347 ,
       2.504 , 4.857 , 2.109 , 4.702 , 1.56  , 4.924 , 3.072 , 4.132 ,
       2.958 , 2.823 , 1.767 , 3.373 , 5.335 , 1.999 , 3.526 , 5.016 ,
       4.102 , 1.649 , 4.539 , 2.443 , 5.209 , 4.711 , 2.828 , 0.903 ,
       5.388 , 2.462 , 2.235 , 1.91  , 2.936 , 3.586 , 1.661 , 1.969 ,
       4.975 , 1.464 , 3.531 , 2.3   , 2.217 , 2.858 , 4.675 , 1.502 ,
       3.412 , 2.461 , 4.378 , 2.269 , 3.975 , 4.756 , 3.619 , 1.367 ,
       5.532 , 3.12  , 5.637 , 3.357 , 6.992 , 1.018 , 2.956 , 3.112 ,
       6.715 , 4.217 , 3.986 , 4.92  , 1.717 , 2.27  , 2.64  ])
```

min_payment_amt

```
array([6.55 , 5.144, 6.148, 5.185, 5.837, 5.   , 5.308, 4.825, 6.273,
       5.003, 5.894, 5.299, 5.338, 6.453, 5.056, 4.958, 5.175, 5.063,
       5.088, 5.316, 5.001, 5.395, 5.919, 5.231, 5.18 , 5.847, 4.782,
       5.163, 5.132, 4.957, 4.872, 5.044, 5.176, 5.091, 5.178, 5.219,
       5.049, 5.221, 5.352, 6.185, 6.097, 4.805, 5.967, 6.32 , 5.194,
       4.828, 6.163, 5.75 , 5.879, 5.067, 6.009, 5.965, 6.053, 5.922,
       5.045, 5.307, 6.102, 5.111, 5.314, 5.15 , 6.2  , 6.451, 4.87 ,
       5.002, 6.449, 6.187, 5.949, 5.439, 5.22 , 5.841, 5.487, 4.607,
       5.012, 5.877, 4.956, 5.396, 5.795, 5.703, 4.781, 6.231, 5.905,
       5.089, 5.929, 6.238, 5.966, 5.046, 6.362, 6.448, 5.44 , 4.914,
       5.992, 5.752, 5.224, 6.188, 5.275, 6.316, 4.961, 5.182, 5.147,
       5.491, 5.962, 5.038, 5.263, 5.351, 4.605, 6.498, 6.06 , 5.661,
       5.136, 4.869, 5.158, 5.924, 4.963, 5.27 , 5.527, 4.649, 5.092,
       5.222, 5.971, 6.321, 6.109, 5.791, 6.197, 5.135, 5.528, 5.36 ,
       4.83 , 5.31 , 5.794, 6.079, 5.533, 4.794, 5.3  , 5.097, 4.745,
       5.746, 4.519, 5.228, 4.783, 4.703, 6.011, 5.88 , 6.229, 5.484,
       5.309, 5.618, 4.738, 5.443])
```

max_spent_in_single_shopping

- **TAB:7 CHECKING FOR ANOMALIES FOR VARIABLES IN THE DATASET**

- No Anomalies found in the Dataset.

## CHECKING FOR DUPLICATE ROWS.

- Number of Duplicated Row in the Dataset = 0

[ 0,0,0,0,0,0,0,0,----------0,0,0,0 ]

TAB:8 DUPLICATE ROWS.

## Univariate Analysis

### *Histogram & Boxplot

A histogram takes as input a numeric variable only. The variable is cut into several bins, and the number of observation per bin is represented by the height of the bar. It is possible to represent the distribution of several variable on the same axis using this technique.

A boxplot gives a nice summary of one or several numeric variables. The line that divides the box into 2 parts represents the median of the data. The end of the box shows the upper and lower quartiles. The extreme lines show the highest and lowest value excluding outliers.

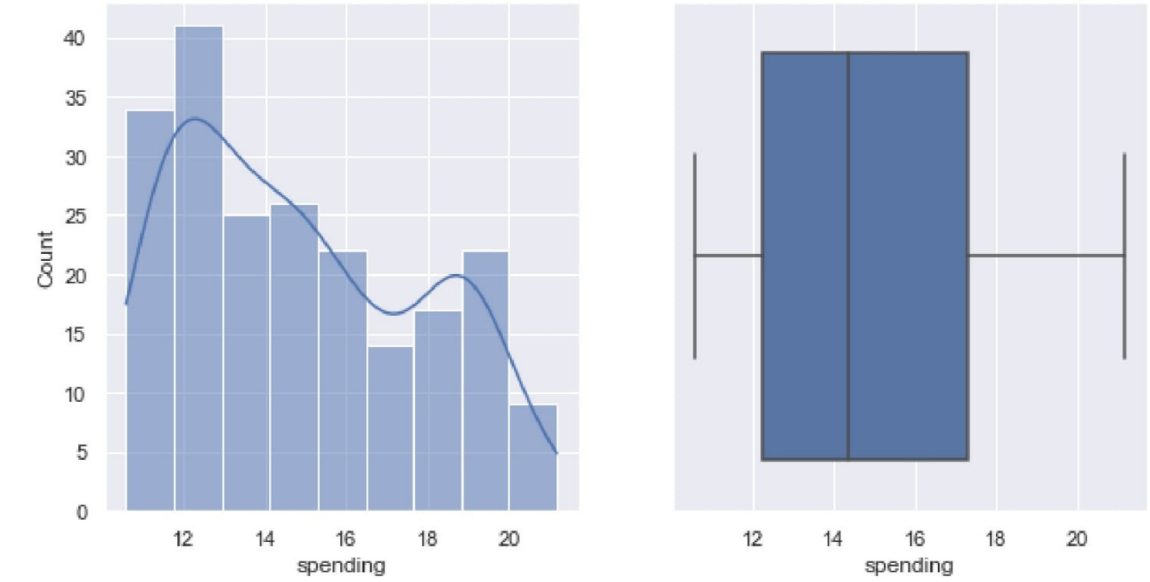



FIG:1 HISTOGRAM & BOXPLOT OF SPENDING

### Description

count 210.000000
mean 14.847524
std 2.909699
min 10.590000
25% 12.270000
50% 14.355000
75% 17.305000
max 21.180000
Name: spending,
dtype: float64

### Skewness

0.39702715402072153

## Insight

- spending:amount spent by the customer per month (in 1000s) ranges from a minimum of 10.59 to maximum of 21.18.
- The average spending:amount spent by the customer per month (in 1000s) is around 14.84.
- The standard deviation of the spending:amount spent by the customer per month (in 1000s) is 2.909.
- 25% , 50% (median) and 75 % of the spending:amount spent by the customer per month (in 1000s) are 12.27 , 14.35 and 17.30.
- Skewness indicating that the ditribution is slightly right skewed.

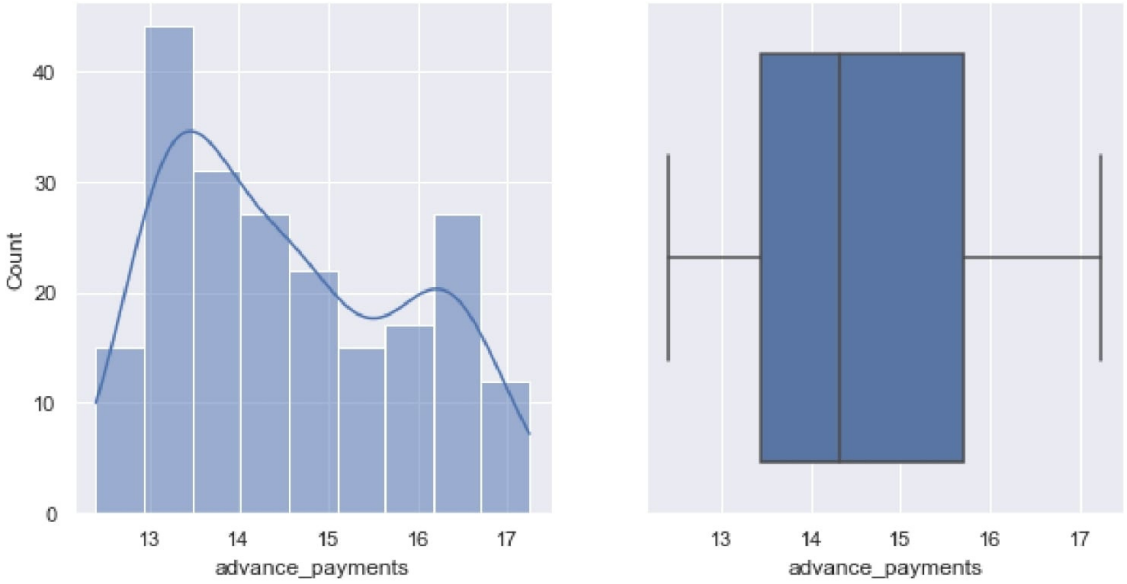



FIG:2 HISTOGRAM & BOXPLOT OF ADVANCE PAYMENTS

**Description**

count 210.000000
mean 14.559286
std 1.305959
min 12.410000
25% 13.450000
50% 14.320000
75% 15.715000
max 17.250000
Name:
advance_payments,
dtype: float64

**Skewness**

0.38380604212562563

## Insights

- advance_payments:amount paid by the customer in advance by cash (in 100s) ranges from a minimum of 12.41 to maximum of 17.25.
- The average advance_payments:amount paid by the customer in advance by cash (in 100s) is around 14.55.
- The standard deviation of the advance_payments:amount paid by the customer in advance by cash (in 100s) is 1.305.
- 25% , 50% (median) and 75 % of the advance_payments:amount paid by the customer in advance by cash (in 100s) are 13.45, 14.32 , and 15.71.
- Skewness indicating that the ditribution is slightly right skewed.

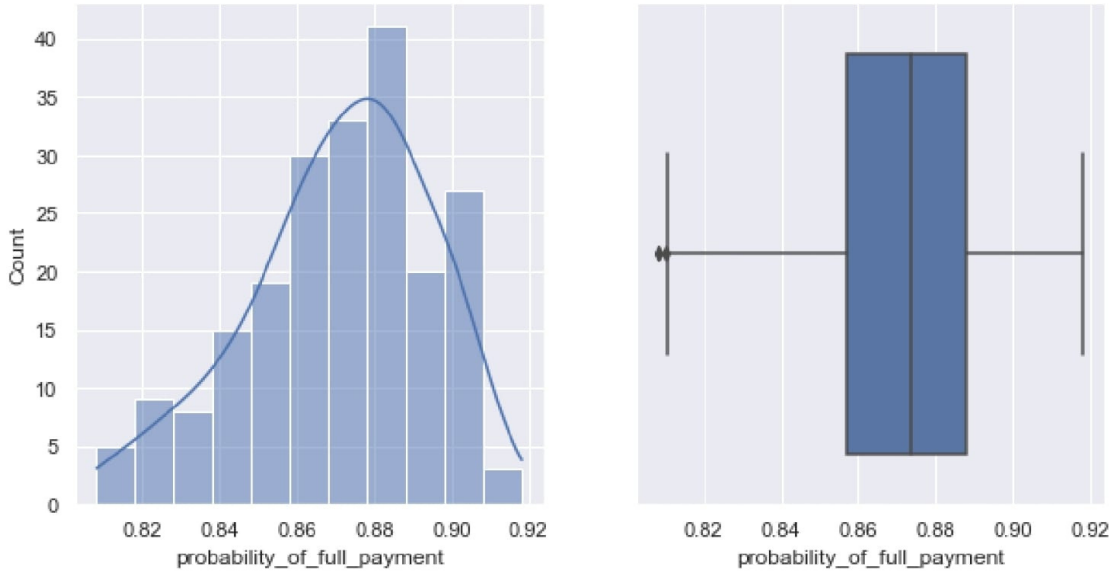



FIG:3 HISTOGRAM & BOXPLOT OF PROBABILITY OF FULL PAYMENT

| Description | | Skewness |
|---|---|---|
| count | 210.000000 | -0.5341035521949097 |
| mean | 0.870999 | |
| std | 0.023629 | |
| min | 0.808100 | |
| 25% | 0.856900 | |
| 50% | 0.873450 | |
| 75% | 0.887775 | |
| max | 0.918300 | |
| Name: probability_of_full_payment, dtype: float64 | | |

## Insights

- probability_of_full_payment: Probability of payment done in full by the customer to the bank ranges from a minimum of 0.8081 to maximum of 0.9183.
- The average probability_of_full_payment: Probability of payment done in full by the customer to the bank is around 0.8709.
- The standard deviation of the probability_of_full_payment: Probability of payment done in full by the customer to the bank is 0.0235.
- 25% , 50% (median) and 75 % of the probability_of_full_payment: Probability of payment done in full by the customer to the bank are 0.856, 0.873 , and 0.887.
- Skewness indicating that the ditribution is slightly left skewed.
- probability_of_full_payment: Probability of payment done in full by the customer to the bank contains outlier.

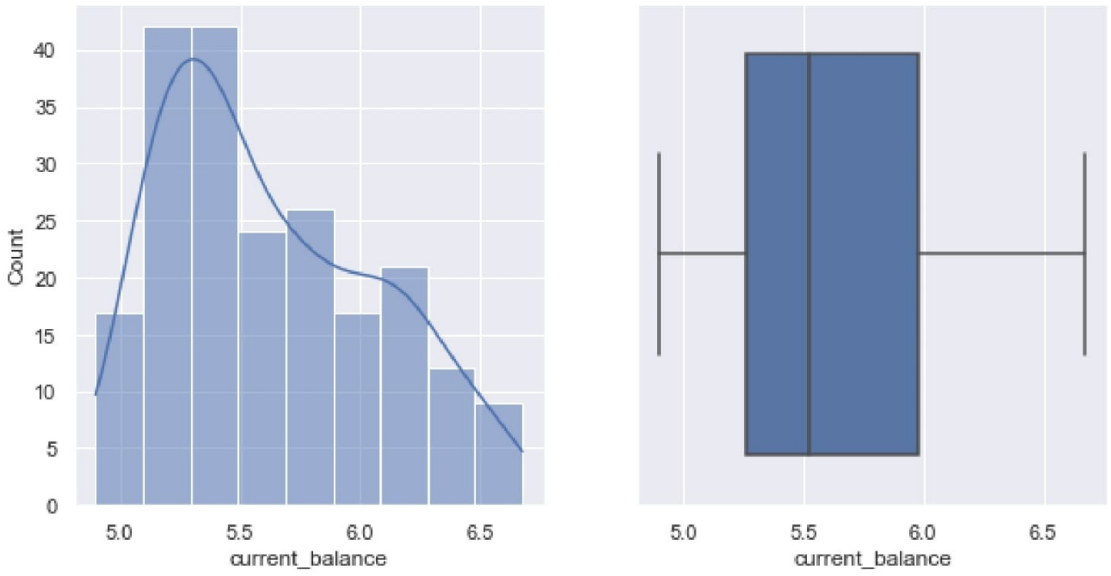



FIG:4 HISTOGRAM & BOXPLOT OF CURRENT BALANCE

**Description**

count 210.000000
mean 5.628533
std 0.443063
min 4.899000
25% 5.262250
50% 5.523500
75% 5.979750
max 6.675000
Name: current_balance,
dtype: float64

**Skewness**

0.5217206481959239

## Insights

- current_balance: Balance amount left in the account to make purchases (in 1000s) ranges from a minimum of 4.899 to maximum of 6.675.
- The average current_balance: Balance amount left in the account to make purchases (in 1000s)is around 5.628.
- The standard deviation of the current_balance: Balance amount left in the account to make purchases (in 1000s) is 0.443.
- 25% , 50% (median) and 75 % of the current_balance: Balance amount left in the account to make purchases (in 1000s) are 5.262, 5.523 , and 5.979.
- Skewness indicating that the ditribution is slightly right skewed.

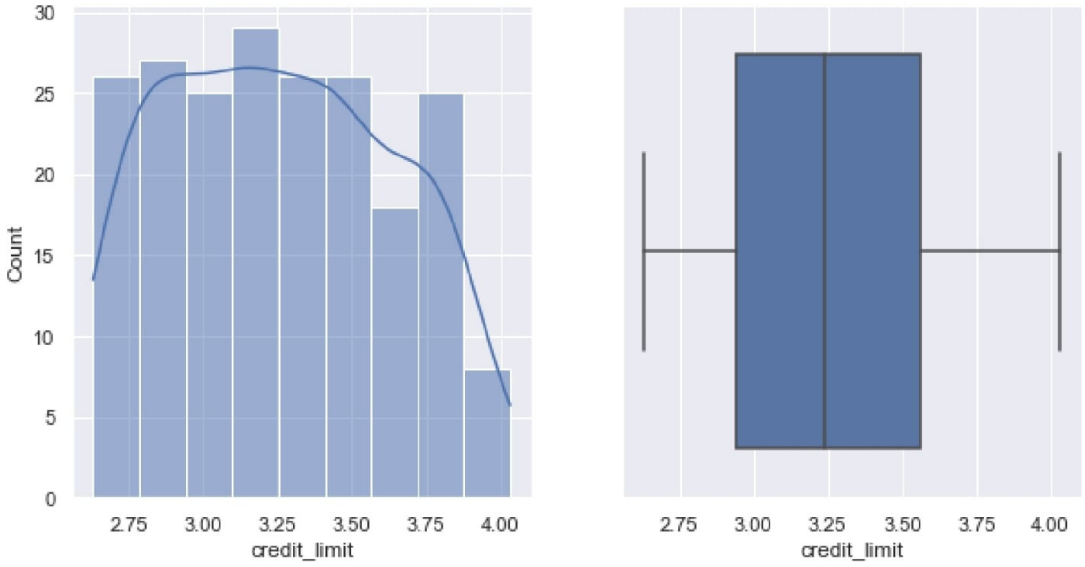



FIG:5 HISTOGRAM & BOXPLOT OF CREDIT LIMIT

**Description**

COUNT 210.000000
MEAN 3.258605
STD 0.377714
MIN 2.630000
25% 2.944000
50% 3.237000
75% 3.561750
MAX 4.033000
NAME: CREDIT_LIMIT,
DTYPE: FLOAT64

**Skewness**

0.13341648969738146

## Insights

- credit_limit: Limit of the amount in credit card (10000s) ranges from a minimum of 2.630 to maximum of 4.033.
- The average credit_limit: Limit of the amount in credit card (10000s) is around 3.258.
- The standard deviation of the credit_limit: Limit of the amount in credit card (10000s) is 0.377.
- 25% , 50% (median) and 75 % of the credit_limit: Limit of the amount in credit card (10000s) are 2.944, 3.237 , and 3.561.
- Skewness indicating that the ditribution is slightly right skewed.

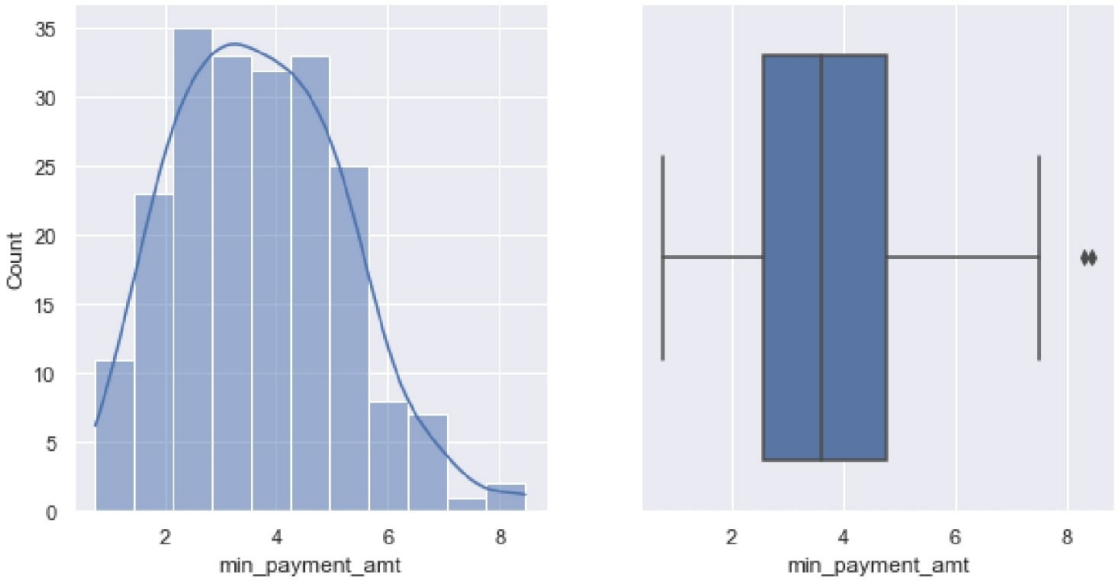



FIG:6 HISTOGRAM & BOXPLOT OF MIN PAYMENT AMOUNT

| Description | |
|---|---|
| count | 210.000000 |
| mean | 3.700201 |
| std | 1.503557 |
| min | 0.765100 |
| 25% | 2.561500 |
| 50% | 3.599000 |
| 75% | 4.768750 |
| max | 8.456000 |
| Name: min_payment_amt, dtype: float64 | |

| Skewness |
|---|
| 0.3987925792256687 |

## Insights

- min_payment_amt : minimum paid by the customer while making payments for purchases made monthly (in 100s) ranges from a minimum of 0.765 to maximum of 8.456.
- The average min_payment_amt : minimum paid by the customer while making payments for purchases made monthly (in 100s) is around 3.700.
- The standard deviation of the min_payment_amt : minimum paid by the customer while making payments for purchases made monthly (in 100s) is 1.494.
- 25% , 50% (median) and 75 % of the min_payment_amt : minimum paid by the customer while making payments for purchases made monthly (in 100s) are 2.561 , 3.599 and 4.768.
- Skewness indicating that the ditribution is slightly right skewed.
- min_payment_amt : minimum paid by the customer while making payments for purchases made monthly (in 100s) contains outliers.

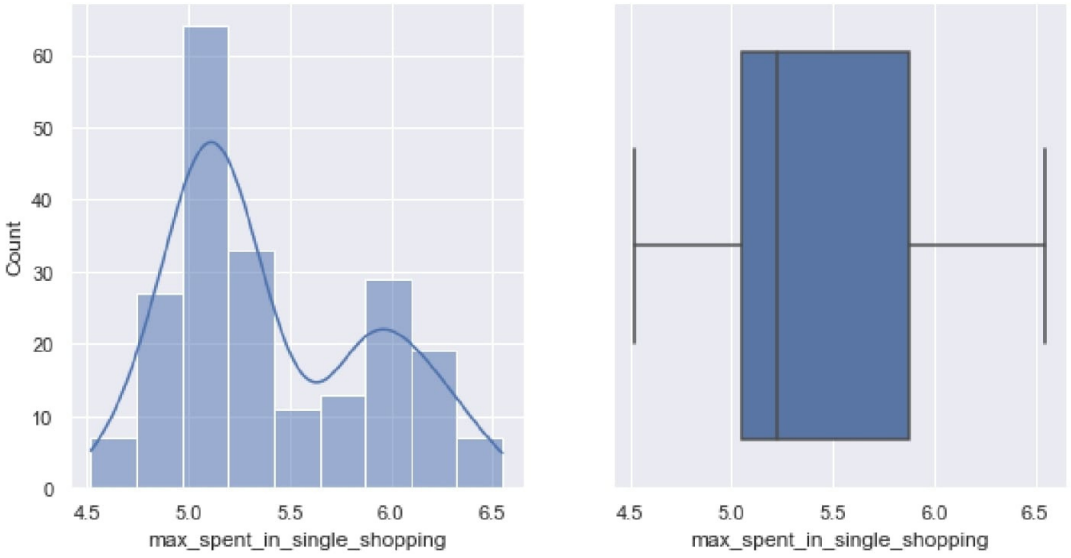



FIG:7 HISTOGRAM & BOXPLOT OF MAX SPENT IN SINGLE SHOPPING

**Description**

count 210.000000
mean 5.408071
std 0.491480
min 4.519000
25% 5.045000
50% 5.223000
75% 5.877000
max 6.550000
Name:
max_spent_in_single_shopping,
dtype: float64

**Skewness**

0.5578758322317957

## Insights

- max_spent_in_single_shopping: Maximum amount spent in one purchase (in 1000s) ranges from a minimum of 4.519 to maximum of 6.550.
- The average max_spent_in_single_shopping: Maximum amount spent in one purchase (in 1000s) is around 5.408.
- The standard deviation of the max_spent_in_single_shopping: Maximum amount spent in one purchase (in 1000s) is 0.491.
- 25% , 50% (median) and 75 % of the max_spent_in_single_shopping: Maximum amount spent in one purchase (in 1000s) are 5.045 , 5.223 and 5.877.
- Skewness indicating that the ditribution is slightly right skewed.

## Bi-variate Analysis

*Scatter Plot

A scatter plot (aka scatter chart, scatter graph) uses dots to represent values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables.

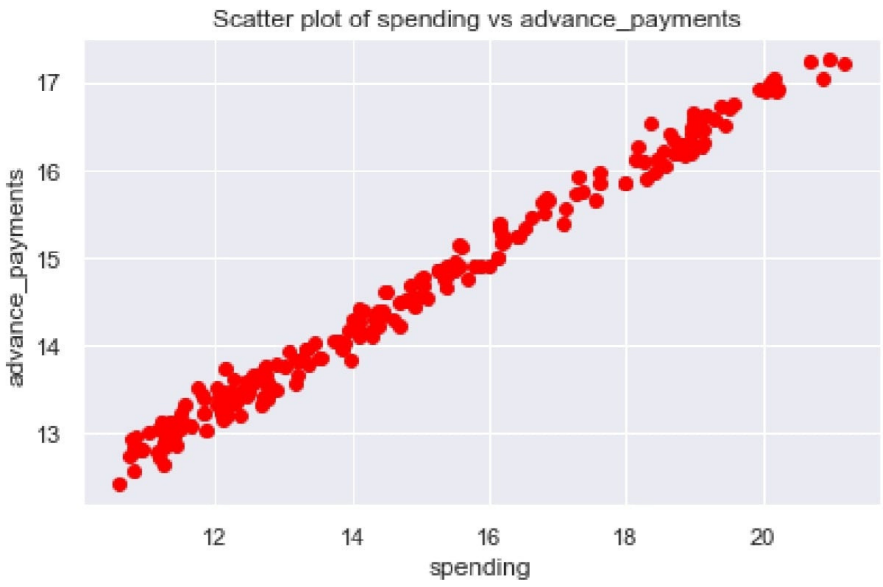



FIG:8 SCATTER PLOT OF SPENDING VS ADVANCE PAYMENTS

## Insights

- From the above plot we see that as the spending increases the advance_payments is also increasing showing a positive relationship.

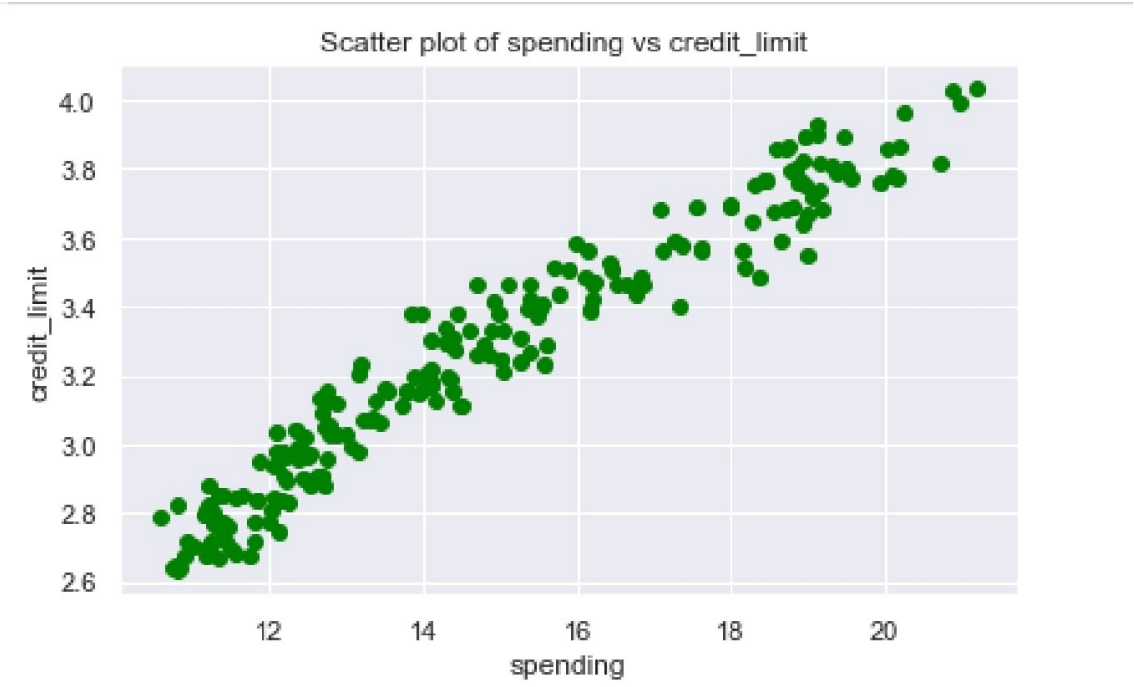



FIG:9 SCATTER PLOT OF SPENDING VS CREDIT LIMIT

## Insights

- From the above plot we see that as the spending increases the credit_limit is also increasing showing a positive relationship.

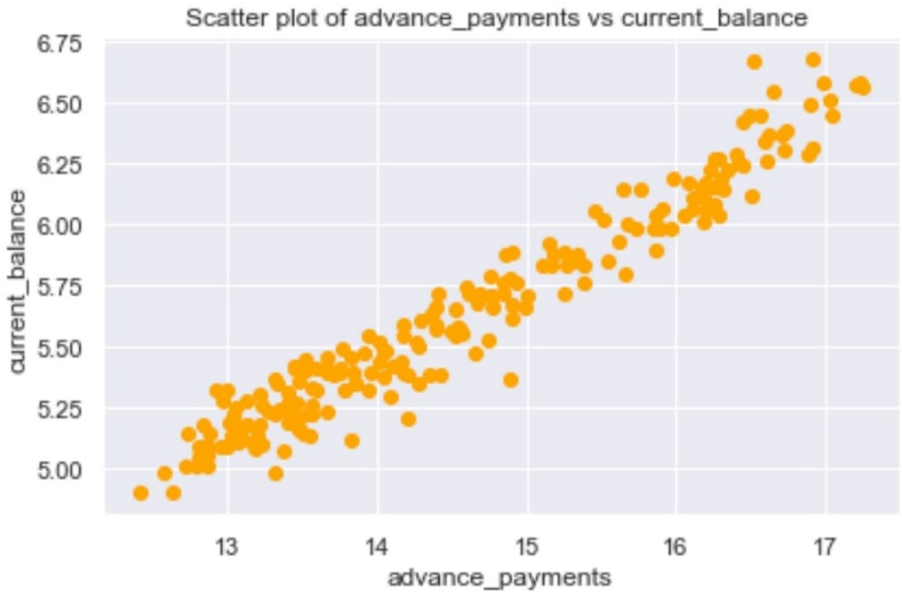



FIG:10 SCATTER PLOT OF ADVANCE PAYMENTS VS CURRENT BALANCE

## Insights

- From the above plot we see that as the advance_payments increases the current_balance is also increasing showing a positive relationship.

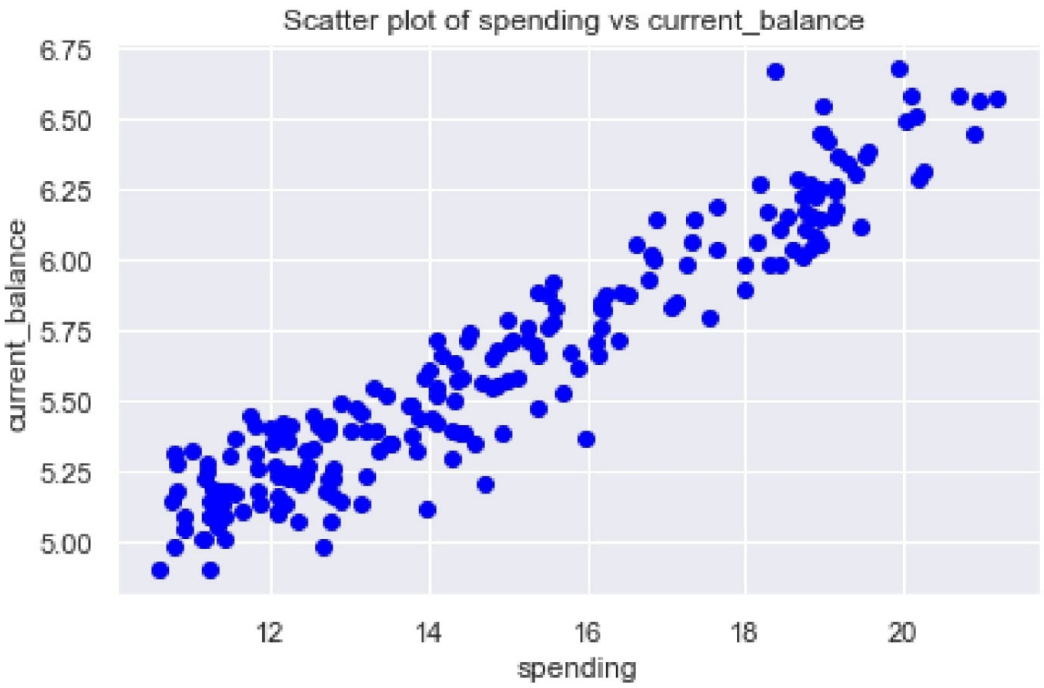



FIG:11 SCATTER PLOT OF SPENDING VS CURRENT BALANCE

## Insights

- From the above plot we see that as the spending increases the current_balance is also increasing showing a positive relationship.

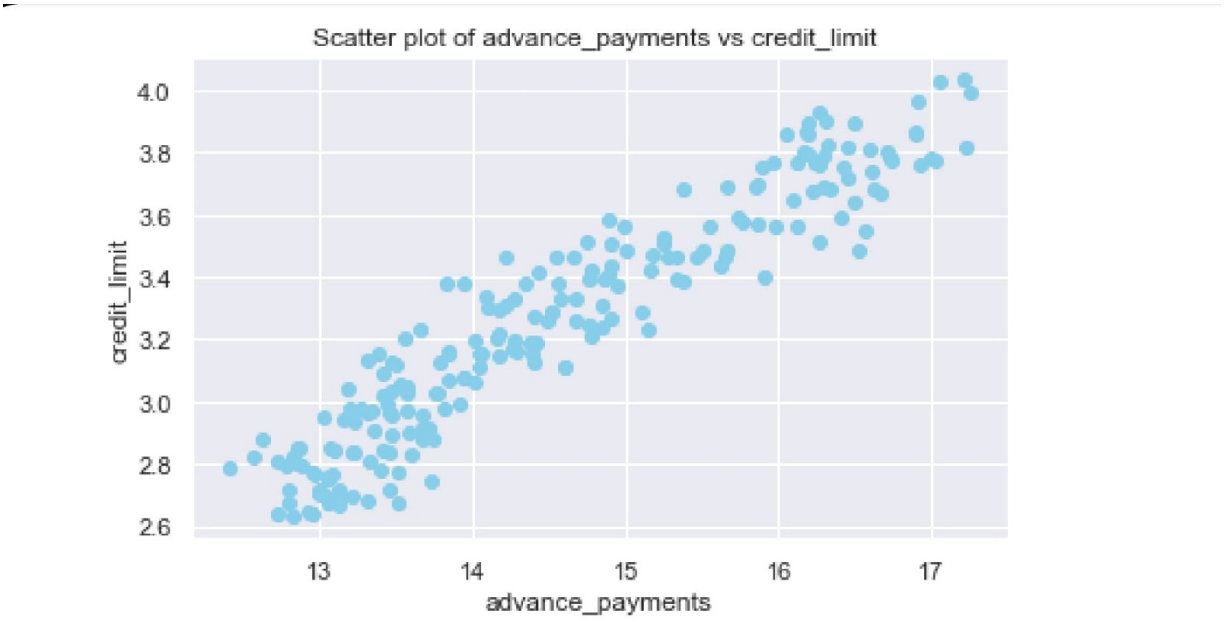



FIG:12 SCATTER PLOT OF ADVANCE PAYMENT VS CREDIT LIMIT

## Insights

- From the above plot we see that as the advance_payments increases the credit_limit is also increasing showing a positive relationship.

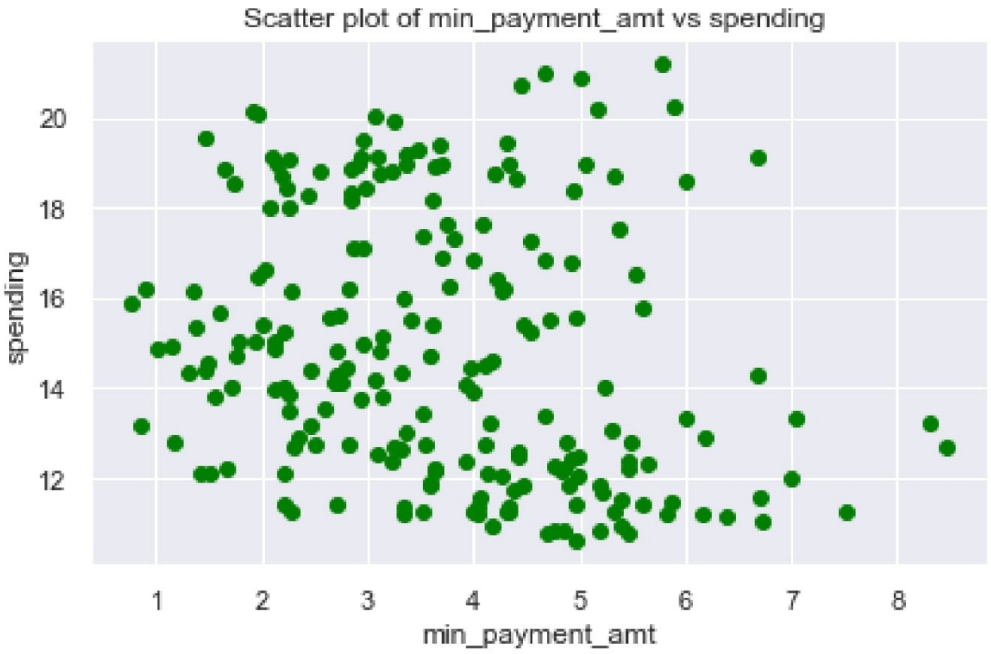



FIG:13 SCATTER PLOT OF MIN PAYMENT AMOUNT VS SPENDING

## Insights

- From the above plot we see that as the min_payment_amt and the spending is showing a poor relationship.

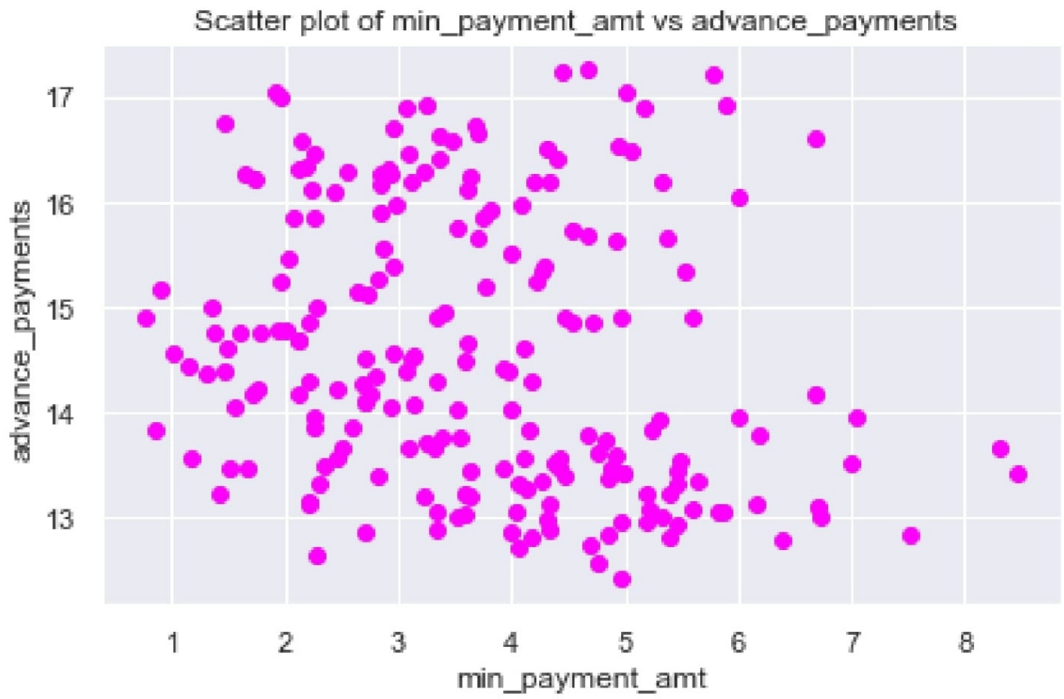



FIG:14 SCATTER PLOT OF MIN PAYMENT AMOUNT VS ADVANCE PAYMENTS

### Insights

- From the above plot we see that as the min_payment_amt and the advance_payments is showing a poor relationship.

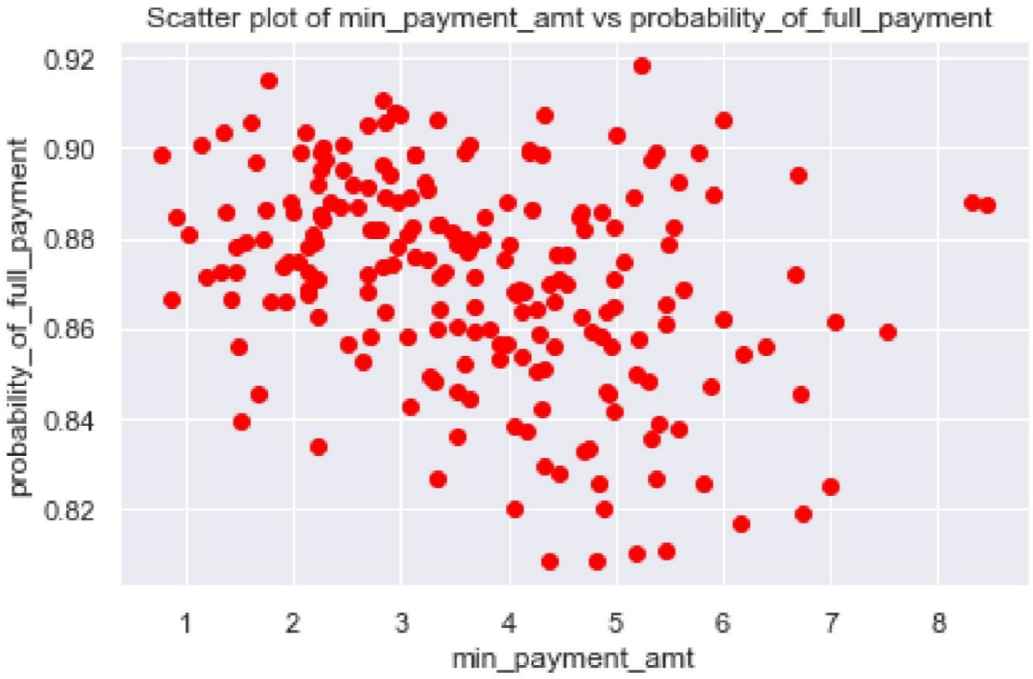



FIG:15 SCATTER PLOT OF MIN PAYMENT AMOUNT VS PROBABILITY OF FULL PAYMENT

### Insights

- From the above plot we see that as the min_payment_amt decreases the credit_limit is also decreasing showing a negetive relationship.

### Multivariate Analysis

*** Heatmap**

A correlation heatmap uses colored cells, typically in a monochromatic scale, to show a 2D correlation matrix (table) between two discrete dimensions or event types.Correlation heatmaps are ideal for comparing the measurement for each pair of dimension values.Darker Shades have higher Correaltion , while lighter shades have smaller values of correaltion as compared to darker shades values.Correlation values near to 1 or -1 are highly positively correlated and highly negatively correlated respectively. Correlation values near to 0 are not correlated to each other.

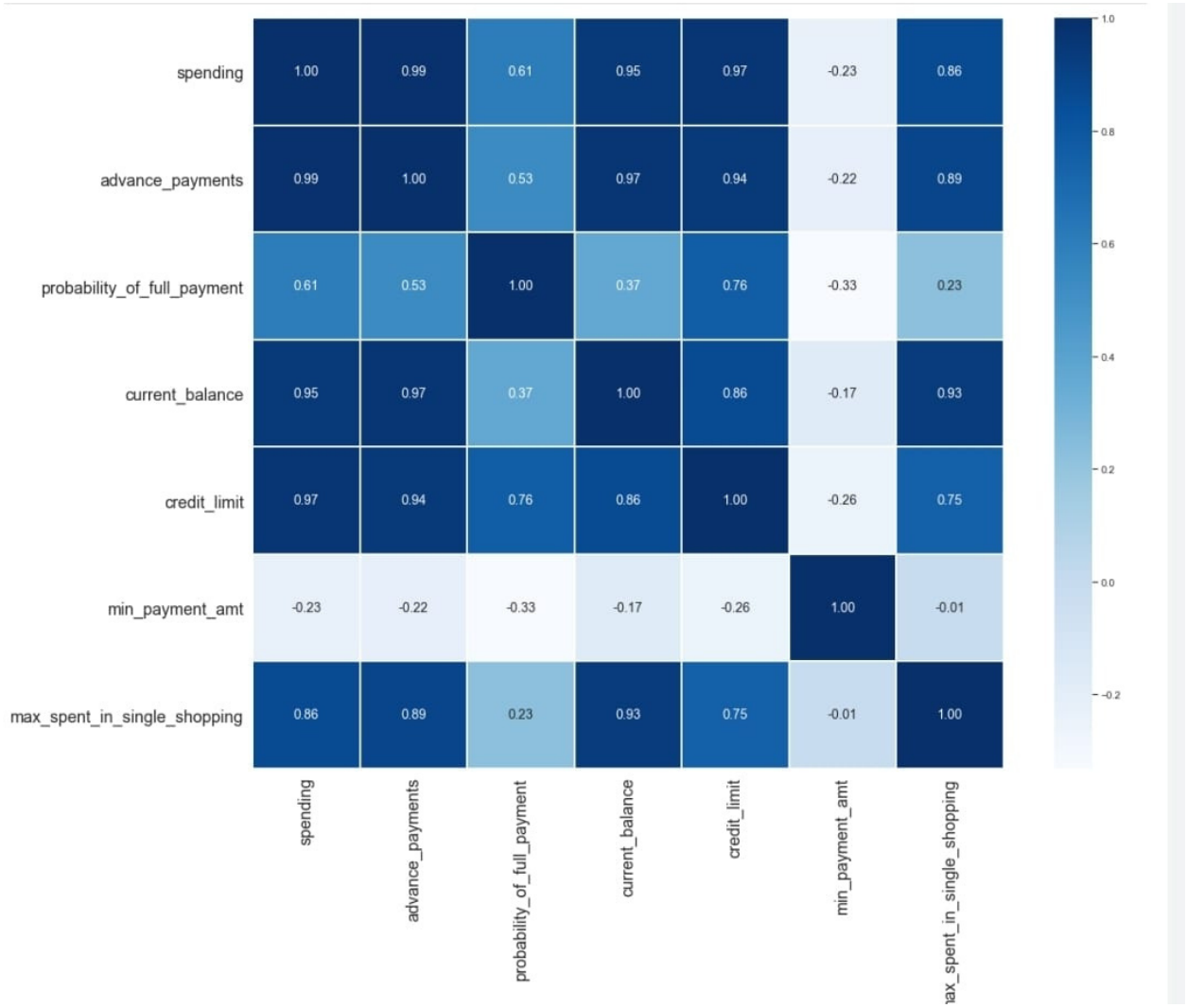



FIG:16 HEATMAP OF PROBLEM 1

## Insights

- From the above correlation table we conclude that,
- spending with advance_payments have max value of correlation is 0.99.
- spending with credit_limit have strong value of correlation 0.97.
- advance_payments with current_balance also have a strong correaltion value 0.97.
- spending with current_balance also have a strong correaltion value 0.95.
- advance_payments with credit_limit also show significant correlation value of 0.94.
- min_payment_amt with spending and min_payment_amt with advance_payments show poor correlation i.e. -0.23,-0.22.
- min_payment_amt with probability_of_full_payment have min value of correlation -0.33.

## *Pairplot

- Pairplot shows the relationship between the variables in the form of scatterplot and the distribution of the variable in the form of histogram.

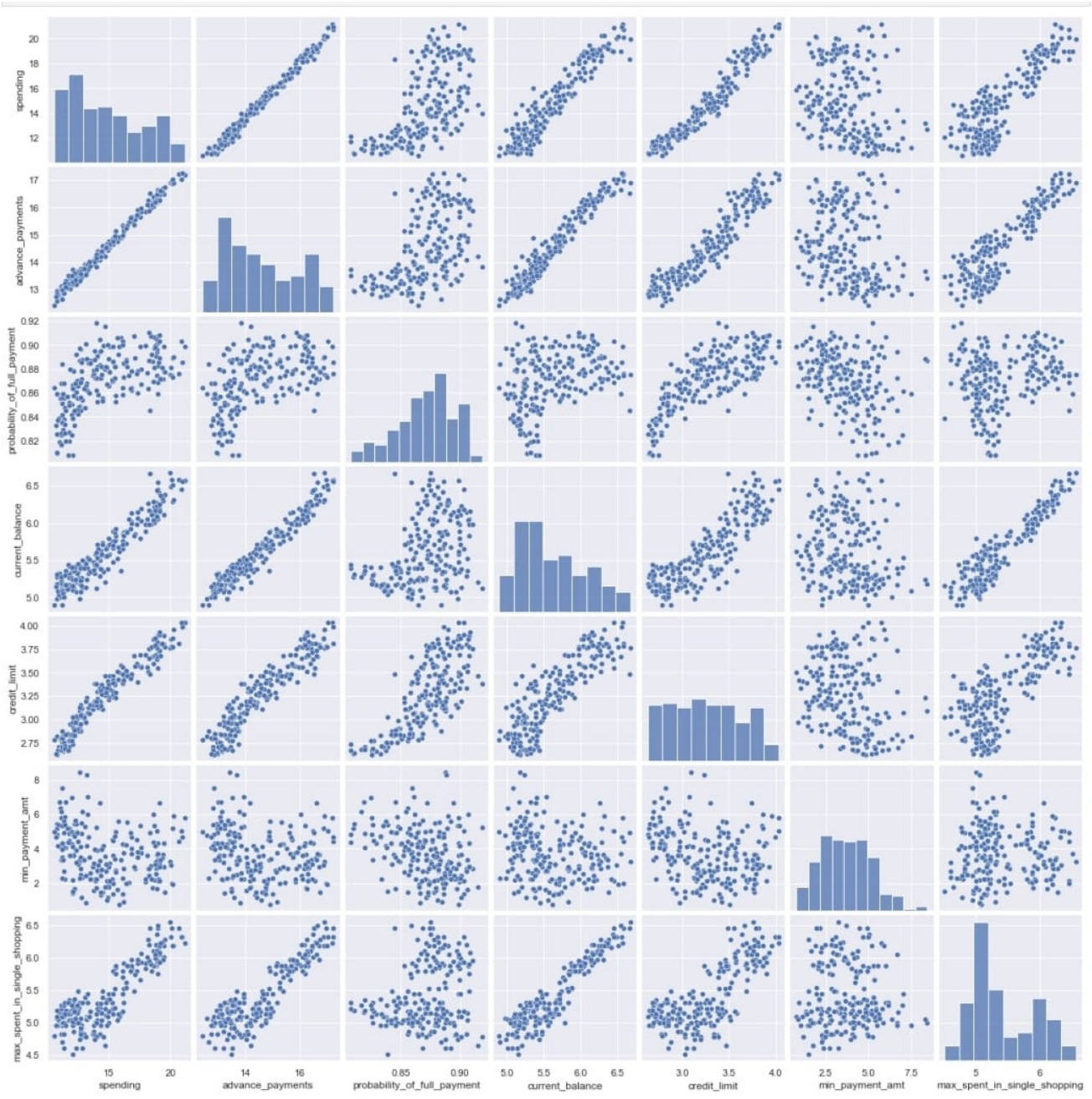



FIG:17 PAIRPLOT OF PROBLEM 1

### Insights

- From the graph, we can see that there is positive linear relationship between variables like spending and advance_payments , spending and credit_limit , advance_payments and current_balance , spending and current_balance.

## Outliers Detection in the dataset

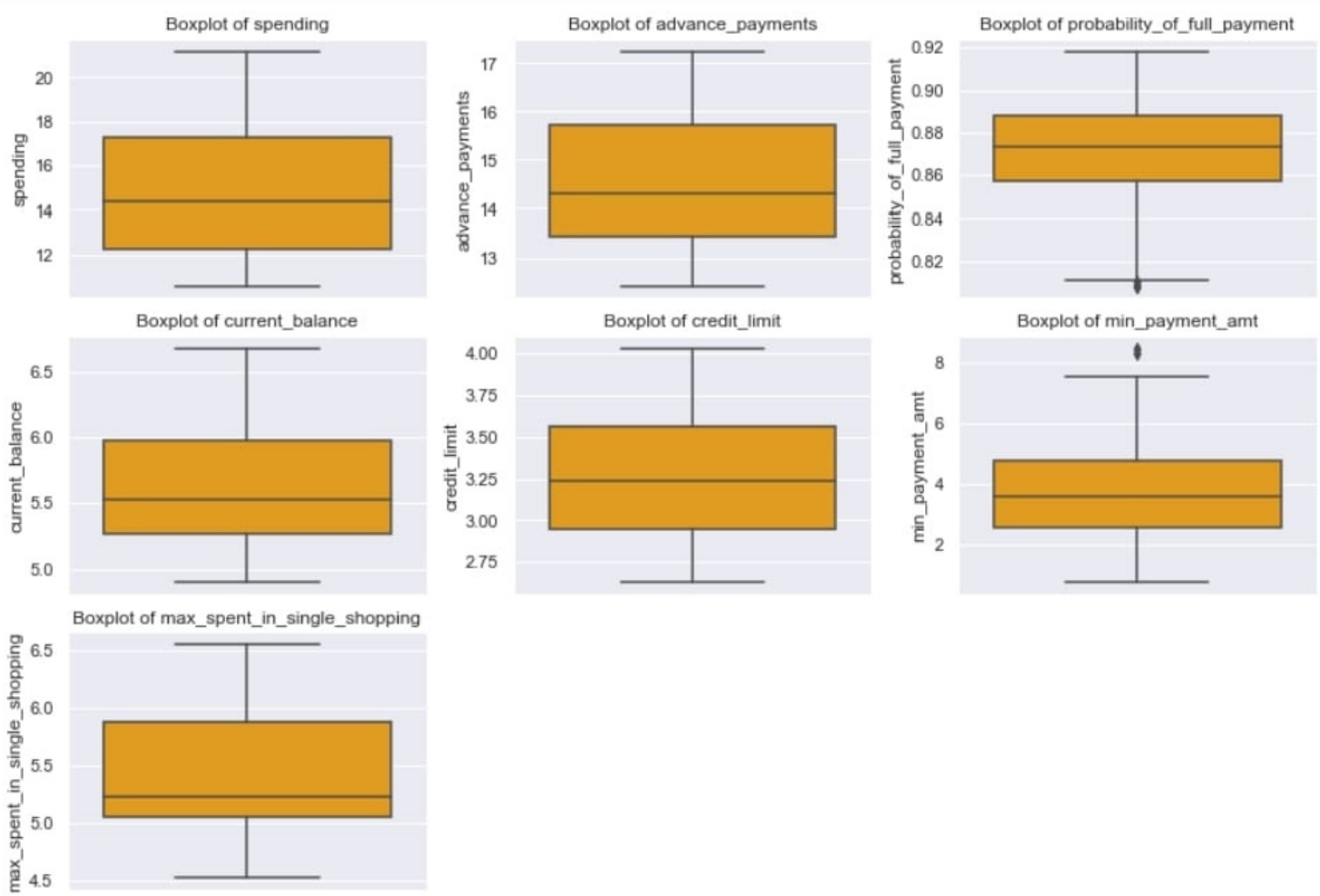



FIG:18 OUTLIER DETECTION BOXPLOT OF PROBLEM 1

## Conclusion

- As in the above plotted boxplots we found outliers in the probability_of_full_payment & min_payment_amt ,so we need to treat outliers as clustering results are affected by the presence of outliers.

## Outliers Treatment in the dataset

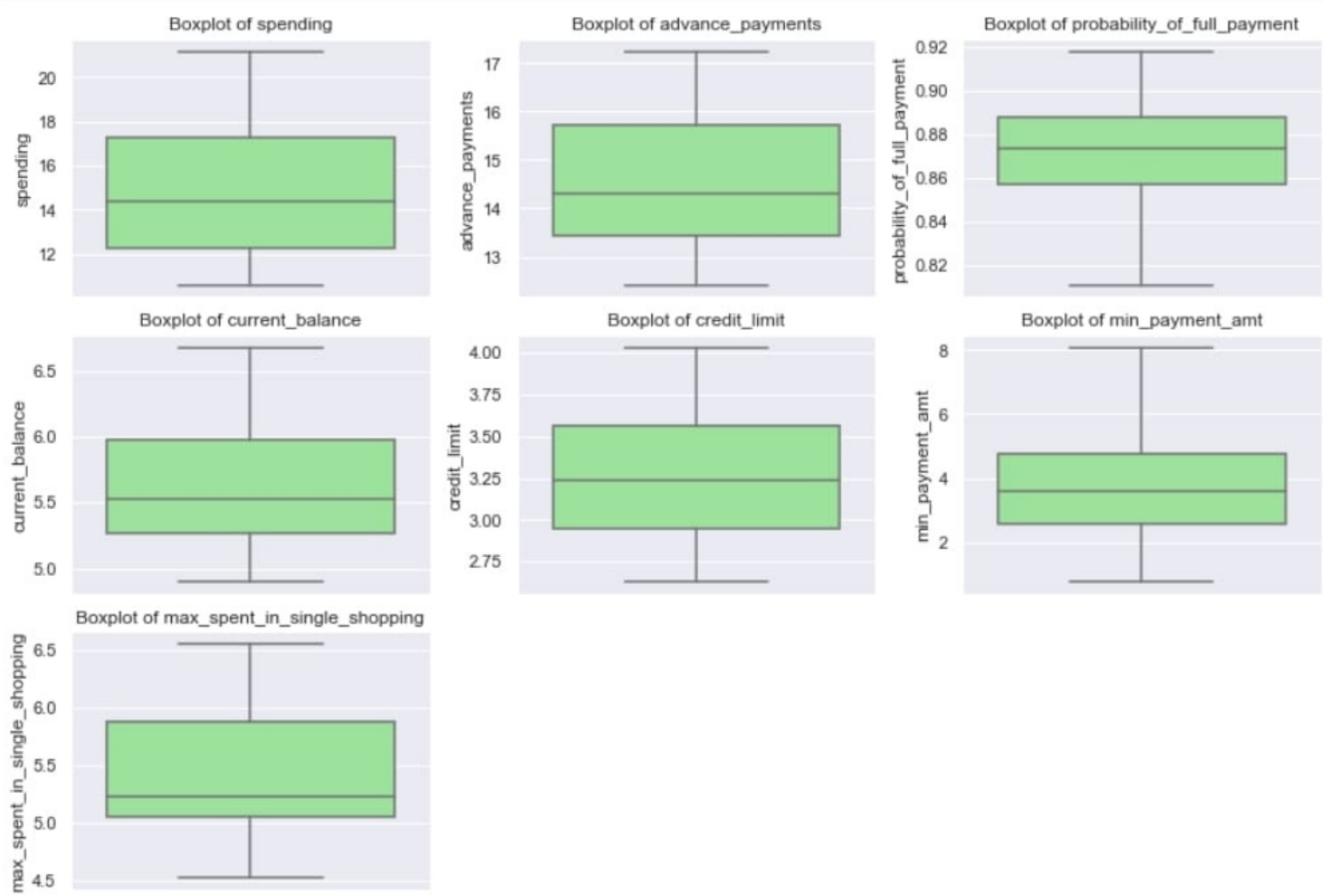



FIG:19 OUTLIER TREATMENT BOXPLOT OF PROBLEM 1.

## Conclusion:

- As we sucessfully treated the outliers and from the above plotted boxplot, we clearly infer that their is no more outliers present in the dataset. Now we can do the scaling and perform the further processes for clustering.

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

### Conclusion:

- Yes,i think scaling is necessary for clustering in this case. As we all know that agglomerative cluserting is very sensitive to the outliers & Clustering algorithms such as K-means do need feature scaling before they are fed to the algorithm. Since, clustering techniques use Euclidean Distance to form the cohorts, it will be wise to do scaling before calculating the distance.

- When we perform a distance based model or work on disatnce based algorithm scaling or normalization is a requirement. We can either go for Z-Score sacling or Min-Max normalization technique , but in this case we go with Z-Score as per the nature of the given data set as the given data set have afew outliers in ,but not so extreme and the given data set also approaches the normality.That's why we are using z-score scaling on the dataset.

| | spending | advance_payments | probability_of_full_payment | current_balance | credit_limit | min_payment_amt | max_spent_in_single_shopping |
|---|---|---|---|---|---|---|---|
| 0 | 1.754355 | 1.811968 | 0.177628 | 2.367533 | 1.338579 | -0.298625 | 2.328998 |
| 1 | 0.393582 | 0.253840 | 1.505071 | -0.600744 | 0.858236 | -0.242292 | -0.538582 |
| 2 | 1.413300 | 1.428192 | 0.505234 | 1.401485 | 1.317348 | -0.220832 | 1.509107 |
| 3 | -1.384034 | -1.227533 | -2.571391 | -0.793049 | -1.639017 | 0.995699 | -0.454961 |
| 4 | 1.082581 | 0.998364 | 1.198738 | 0.591544 | 1.155464 | -1.092656 | 0.874813 |

TAB:9 SCALED DATASET.

### 5 Number Sumarray of Scaled Data

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| spending | 210.0 | 9.148766e-16 | 1.002389 | -1.466714 | -0.887955 | -0.169674 | 0.846599 | 2.181534 |
| advance_payments | 210.0 | 1.097006e-16 | 1.002389 | -1.649686 | -0.851433 | -0.183664 | 0.887069 | 2.065260 |
| probability_of_full_payment | 210.0 | 1.642601e-15 | 1.002389 | -2.571391 | -0.600968 | 0.103172 | 0.712647 | 2.011371 |
| current_balance | 210.0 | -1.089076e-16 | 1.002389 | -1.650501 | -0.828682 | -0.237628 | 0.794595 | 2.367533 |
| credit_limit | 210.0 | -2.994298e-16 | 1.002389 | -1.668209 | -0.834907 | -0.057335 | 0.804496 | 2.055112 |
| min_payment_amt | 210.0 | 1.512018e-16 | 1.002389 | -1.966425 | -0.761698 | -0.065915 | 0.718559 | 2.938945 |

TAB:10 5 NUMBER SUMARRAY OF SCALED DATA

### Insights

- After zscore scaling the mean of the variables becomes zero and std is equals to 1.
- Scaling changes the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.

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

### Step:1 Importing the necessary python package & functions for Hierarchical Clustering.

1. Dendrogram function for visualization of the clusters formed.
2. Linkage function is for computing the distances and merging the clusters.

### Step:2 Create linkage for clustering , here we are going to use the average linkage.

1. Linkage method used here is = "average linkage method" .
2. Now the linkage stores the various distances at which the n-clusters are sequentially merged.This information used by the dendrogram to create visuals for us.

```
array([[1.90000000e+01, 2.30000000e+01, 1.93561325e-01, 2.00000000e+00],
       [3.00000000e+00, 4.40000000e+01, 2.10880708e-01, 2.00000000e+00],
       [7.00000000e+00, 3.00000000e+01, 2.11416263e-01, 2.00000000e+00],
       [9.50000000e+01, 1.26000000e+02, 2.19575820e-01, 2.00000000e+00],
       [7.00000000e+01, 1.16000000e+02, 2.41240086e-01, 2.00000000e+00],
       [1.48000000e+02, 2.07000000e+02, 2.59139938e-01, 2.00000000e+00],
       [1.27000000e+02, 1.57000000e+02, 2.71769302e-01, 2.00000000e+00],
       [7.60000000e+01, 1.31000000e+02, 2.87275484e-01, 2.00000000e+00],
       [6.70000000e+01, 1.72000000e+02, 2.91364436e-01, 2.00000000e+00],
       [4.00000000e+00, 2.20000000e+01, 2.91569402e-01, 2.00000000e+00],
       [7.10000000e+01, 1.51000000e+02, 2.97134664e-01, 2.00000000e+00],
       [1.85000000e+02, 2.06000000e+02, 3.19058136e-01, 2.00000000e+00],
       [9.90000000e+01, 1.59000000e+02, 3.34862969e-01, 2.00000000e+00],
       [9.00000000e+00, 1.37000000e+02, 3.37994916e-01, 2.00000000e+00],
       [7.50000000e+01, 1.96000000e+02, 3.45544469e-01, 2.00000000e+00],
       [5.60000000e+01, 1.97000000e+02, 3.55807106e-01, 2.00000000e+00],
       [8.80000000e+01, 9.70000000e+01, 3.59654615e-01, 2.00000000e+00],
       [6.40000000e+01, 1.80000000e+02, 3.64523798e-01, 2.00000000e+00],
       [1.11000000e+02, 1.76000000e+02, 3.64843360e-01, 2.00000000e+00],
       [5.20000000e+01, 1.09000000e+02, 3.67595472e-01, 2.00000000e+00],
       [4.90000000e+01, 1.73000000e+02, 3.70274426e-01, 2.00000000e+00],
       [1.10000000e+01, 8.50000000e+01, 3.71467423e-01, 2.00000000e+00],
       [2.00000000e+00, 6.80000000e+01, 3.74171243e-01, 2.00000000e+00],
       [6.20000000e+01, 8.40000000e+01, 3.77607236e-01, 2.00000000e+00],
       [3.70000000e+01, 2.25000000e+02, 3.96501337e-01, 3.00000000e+00],
       [1.93000000e+02, 2.22000000e+02, 4.00436309e-01, 3.00000000e+00],
       [3.50000000e+01, 1.71000000e+02, 4.03661273e-01, 2.00000000e+00],
       [5.80000000e+01, 2.16000000e+02, 4.05995890e-01, 3.00000000e+00],
       [3.90000000e+01, 1.01000000e+02, 4.12485772e-01, 2.00000000e+00],
```

TAB:11 DISTANCES AT WHICH THE N-CLUSTERS ARE SEQUENTIALLY MERGED (AGGLOMERATIVE CLUSTERING)

## Step:3 Visualization & Cutting the Dendrogram with suitable clusters.

A dendrogram is a network structure. It is constituted of a root node that gives birth to several nodes connected by edges or branches.

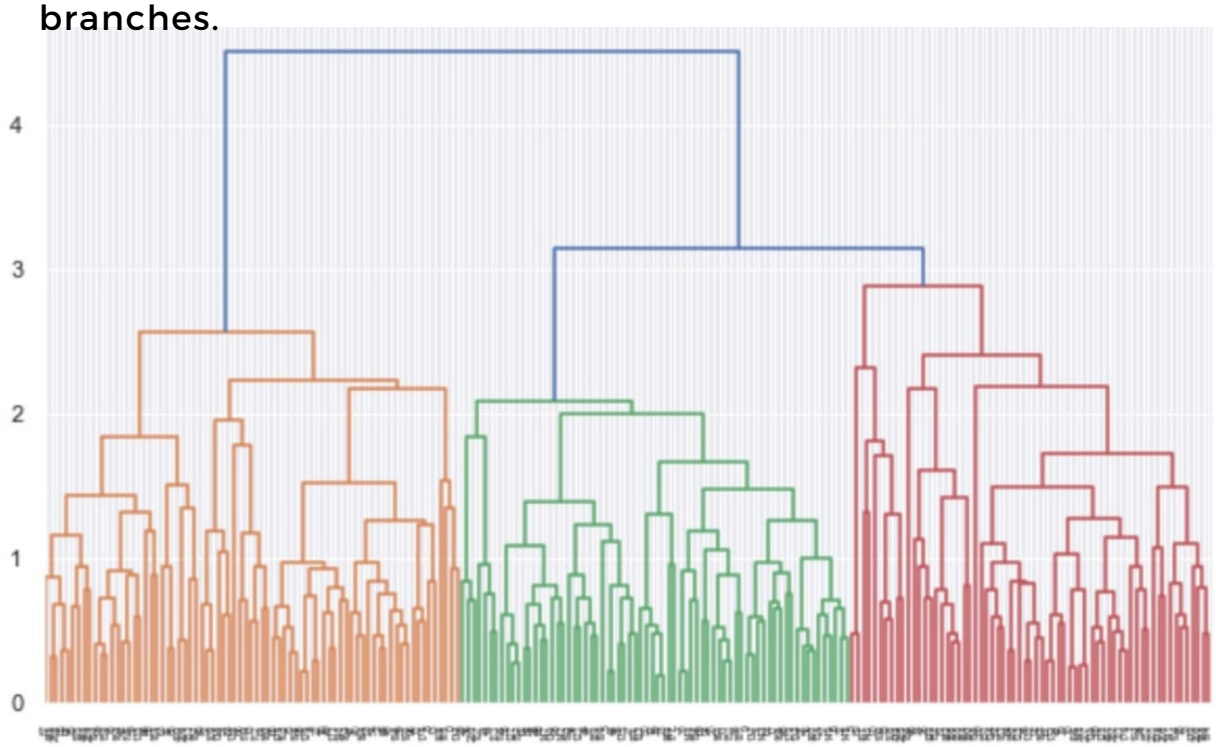



FIG:20 DENDROGRAM

- Lets see more readable and turncate visualize form of dendrogram with some additional parameters , which will give us neater & clearer output of the dendrogram.

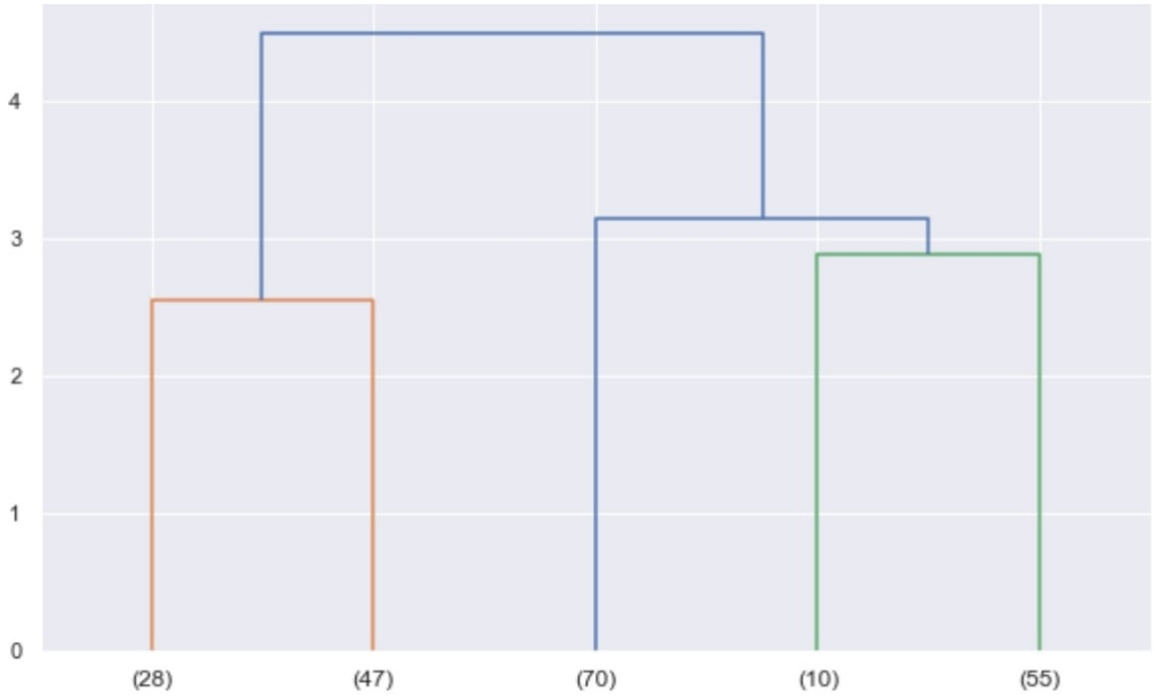



FIG:21 DENDROGRAM - WITH SOME ADDITIONAL PARAMETERS
(TURNCATED DENDROGRAM)

## Insights

From the above plot we observe the last 5 merges and the number of observations in each merge.Cluster 1 have maximum observations.

## Step:4 Importing the fcluster module to create optimum number of clusters.

- Method ------ Maxclust

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

TAB:12 MAXCLUST (AGGLOMERATIVE CLUSTERING)

## Insights

- From the maxclsut criterion fuction we obtain the 3 clusters , i.e. cluster 1 , cluster 2 & cluster 3.

## Step:5 Appending the clusters into the original dataset.

| spending | advance_payments | probability_of_full_payment | current_balance | credit_limit | min_payment_amt | max_spent_in_single_shopping | agg_clusters |
|---|---|---|---|---|---|---|---|
| 19.94 | 16.92 | 0.875200 | 6.675 | 3.763 | 3.252000 | 6.550 | 1 |
| 15.99 | 14.89 | 0.906400 | 5.363 | 3.582 | 3.336000 | 5.144 | 3 |
| 18.95 | 16.42 | 0.882900 | 6.248 | 3.755 | 3.368000 | 6.148 | 1 |
| 10.83 | 12.96 | 0.810588 | 5.278 | 2.641 | 5.182000 | 5.185 | 2 |
| 17.99 | 15.86 | 0.899200 | 5.890 | 3.694 | 2.068000 | 5.837 | 1 |
| 12.70 | 13.41 | 0.887400 | 5.183 | 3.091 | 8.079625 | 5.000 | 3 |
| 12.02 | 13.33 | 0.850300 | 5.350 | 2.810 | 4.271000 | 5.308 | 2 |
| 13.74 | 14.05 | 0.874400 | 5.482 | 3.114 | 2.932000 | 4.825 | 2 |
| 18.17 | 16.26 | 0.863700 | 6.271 | 3.512 | 2.853000 | 6.273 | 1 |
| 11.23 | 12.88 | 0.851100 | 5.140 | 2.795 | 4.325000 | 5.003 | 2 |

TAB:13 clusters into the original dataset(AGGLOMERATIVE CLSUTERING)

### Step:6 Cluster Frequency

1 75
2 70
3 65
Name: agg_clusters, dtype: int64

TAB:14 CLUSTER FREQUENCY AGGLOMERATIVE CLUSTERING)

### Insights

- Cluster 1 consists of 75 customers.
- Cluster 2 consists of 70 customers.
- Cluster 3 consists of 65 customers.

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

### Step:1 Importing the necessary python package & functions for K-Means Clustering.

### Step:2 Plot the WSS Plot (Elbow Curve) .

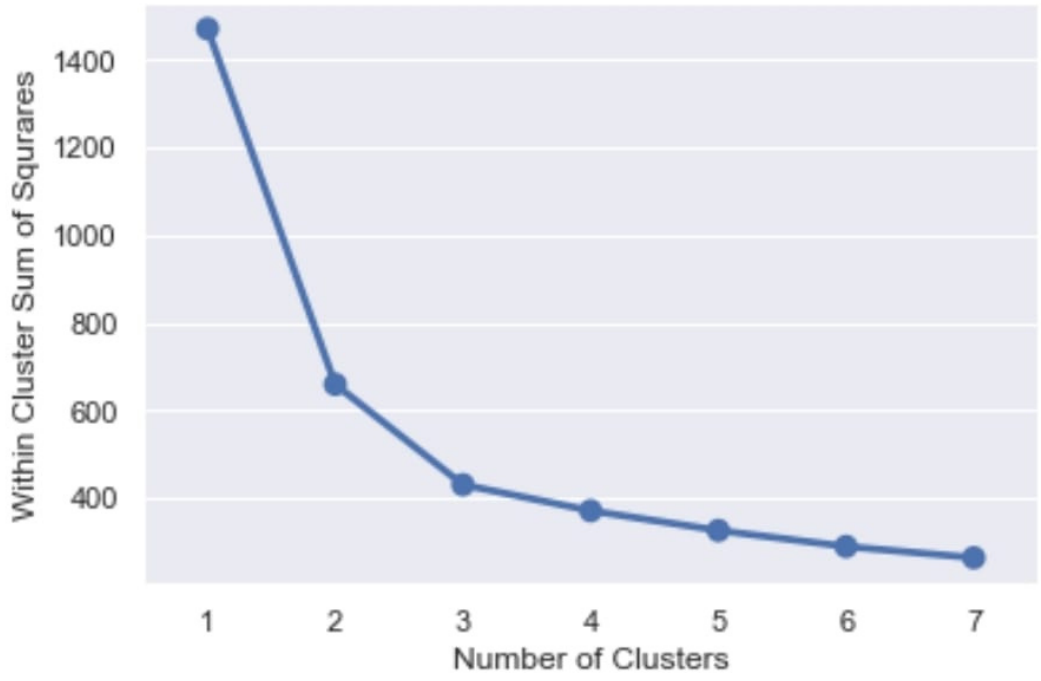



FIG:22 ELBOW CURVE / WSS PLOT

**Step:3 Checking the Within Cluster Sum of Squrares . Forming clusters with K = 1,2,3,4,5,6,7 and comparing the WSS to determine optimum number clusters.**

| K-VALUE | Within Cluster Sum of Squrares |
|---|---|
| K=1 | 1469.999999999999 |
| K=2 | 659.14740095485 |
| K=3 | 430.298481751223 |
| K=4 | 371.0356644664012 |
| K=5 | 325.9741284729876 |
| K=6 | 289.45524862464833 |
| K=7 | 263.859944426353 |

TAB:15 WITHIN CLUSTER SUM OF SQURARES (K-MEANS CLUSTERING)

**Step4: Cluster evaluation - Forming clusters with K = 3,4,5 comparing their silhouette_score**

| K-VALUE | silhouette_score |
|---|---|
| K=3 | 0.4008059221522216 |
| K=4 | 0.3373662527862716 |
| K=5 | 0.28606972536882685 |

TAB:16 SILHOUETTE_SCORE (K-MEANS CLUSTERING)

**Result:**

- From within cluster sum of squares and above plotted elbow curve we observe WSS reduces when k value increases. After third cluster the drop in the within cluster sum of square is not so effective and silhouette_score for k=3 is 0.40080 which is better than k=4 (silhouette_score-0.33736) & k=5 (silhouette_score-0.28606) , so we end up with 3 optimum number clusters for buliding K-Means clustering model.

**Step:4 Creating Clusters using K-Means**

**Step5: Cluster Output for all the observations.**

```
array([1, 2, 1, 0, 1, 0, 0, 2, 1, 0, 1, 2, 0, 1, 2, 0, 2, 0, 0, 0, 0, 0,
       1, 0, 2, 1, 2, 0, 0, 0, 2, 0, 0, 2, 0, 0, 0, 0, 0, 1, 1, 2, 1, 1,
       0, 0, 2, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 2, 0, 0, 2, 2, 1,
       1, 2, 1, 0, 2, 0, 1, 1, 0, 1, 2, 0, 1, 2, 2, 2, 2, 1, 0, 2, 1, 2,
       1, 0, 2, 1, 2, 0, 0, 1, 1, 1, 0, 1, 2, 1, 2, 1, 2, 1, 1, 0, 0, 1,
       2, 2, 1, 0, 0, 1, 2, 2, 0, 1, 2, 0, 0, 0, 2, 2, 1, 0, 2, 2, 0, 2,
       2, 1, 0, 1, 1, 0, 1, 2, 2, 2, 0, 0, 2, 0, 1, 0, 2, 0, 2, 0, 2, 2,
       0, 2, 2, 0, 2, 1, 1, 0, 1, 1, 1, 0, 2, 2, 2, 0, 2, 0, 2, 1, 1, 1,
       2, 0, 2, 0, 2, 2, 2, 2, 1, 1, 0, 2, 2, 0, 0, 2, 0, 1, 2, 1, 1, 0,
       1, 0, 2, 1, 2, 0, 1, 2, 1, 2, 2, 2], dtype=int32)
```

TAB:17 CLUSTER OUTPUT (K-MEANS CLUSTERING)

## Step:6 Appending Clusters to the original dataset

| | spending | advance_payments | probability_of_full_payment | current_balance | credit_limit | min_payment_amt | max_spent_in_single_shopping | agg_clusters | Clus_kmeans3 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 19.94 | 16.92 | 0.875200 | 6.675 | 3.763 | 3.252000 | 6.550 | 1 | 1 |
| 1 | 15.99 | 14.89 | 0.906400 | 5.363 | 3.582 | 3.336000 | 5.144 | 3 | 2 |
| 2 | 18.95 | 16.42 | 0.882900 | 6.248 | 3.755 | 3.368000 | 6.148 | 1 | 1 |
| 3 | 10.83 | 12.96 | 0.810588 | 5.278 | 2.641 | 5.182000 | 5.185 | 2 | 0 |
| 4 | 17.99 | 15.86 | 0.899200 | 5.890 | 3.694 | 2.068000 | 5.837 | 1 | 1 |
| 5 | 12.70 | 13.41 | 0.887400 | 5.183 | 3.091 | 8.079625 | 5.000 | 3 | 0 |
| 6 | 12.02 | 13.33 | 0.850300 | 5.350 | 2.810 | 4.271000 | 5.308 | 2 | 0 |
| 7 | 13.74 | 14.05 | 0.874400 | 5.482 | 3.114 | 2.932000 | 4.825 | 2 | 2 |
| 8 | 18.17 | 16.26 | 0.863700 | 6.271 | 3.512 | 2.853000 | 6.273 | 1 | 1 |
| 9 | 11.23 | 12.88 | 0.851100 | 5.140 | 2.795 | 4.325000 | 5.003 | 2 | 0 |

TAB:18 CLUSTERS INTO THE ORIGINAL DATASET K-MEANS CLSUTERING

## Step:7 Checking the mapping of the datapoints to the clusters is correct or not ?

silhouette_score--------0.4008059221522216

sil_width

```
array([0.5732776 , 0.36556355, 0.63709249, 0.515595  , 0.36097201,
       0.22152508, 0.47529542, 0.36025848, 0.51938329, 0.53443903,
       0.46599399, 0.12839864, 0.39177784, 0.52379458, 0.11202082,
       0.22512083, 0.33760956, 0.5018087 , 0.03635503, 0.23801566,
       0.36177434, 0.3693663 , 0.43153403, 0.26364196, 0.47484293,
       0.06663956, 0.27151643, 0.50414367, 0.55487254, 0.43479958,
       0.37528473, 0.43006502, 0.39151526, 0.3943622 , 0.5362567 ,
       0.55717776, 0.50878421, 0.42617776, 0.50641159, 0.62170114,
       0.55929539, 0.48579454, 0.39864428, 0.61044051, 0.51398993,
       0.37791063, 0.30664315, 0.58154614, 0.48759463, 0.53302467,
       0.31693425, 0.49463828, 0.58531649, 0.59861082, 0.61892471,
       0.23370264, 0.44475373, 0.54060572, 0.57808265, 0.57623567,
       0.55297302, 0.51585343, 0.55579575, 0.27793624, 0.49524145,
```

TAB:19 SIL WIDTH (K-MEANS CLUSTERING).

| | spending | advance_payments | probability_of_full_payment | current_balance | credit_limit | min_payment_amt | max_spent_in_single_shopping | agg_clusters | Clus_kmeans3 | sil_width |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 19.94 | 16.92 | 0.875200 | 6.675 | 3.763 | 3.252 | 6.550 | 1 | 1 | 0.573278 |
| 1 | 15.99 | 14.89 | 0.906400 | 5.363 | 3.582 | 3.336 | 5.144 | 3 | 2 | 0.365564 |
| 2 | 18.95 | 16.42 | 0.882900 | 6.248 | 3.755 | 3.368 | 6.148 | 1 | 1 | 0.637092 |
| 3 | 10.83 | 12.96 | 0.810588 | 5.278 | 2.641 | 5.182 | 5.185 | 2 | 0 | 0.515595 |
| 4 | 17.99 | 15.86 | 0.899200 | 5.890 | 3.694 | 2.068 | 5.837 | 1 | 1 | 0.360972 |

TAB:20 SIL WIDTH INTO THE ORIGINAL DATASET
(KMEANS CLSUTERING)

**Result:**

- From the above perform activities we conclude that
- silhouette_score is postive and good enough for k=3 shows clusters are merged correctly.
- sil_width of the all the obersrvations are positive also shows that the mapping to clusters is performed correctly.
- As our both functions silhouette_score & sil_width tells us that we merges the cluster and maps the obersrvations correctly.

**Step:8 Cluster Frequency**

```
CLUSTER      FREQUENCY
    0            72
    1            67
    2            71
    NAME: CLUS_KMEANS3, DTYPE: INT64
```

TAB:21 CLUSTER FREQUENCY (K-MEANS CLUSTERING)

**Insights**

- Cluster 0 consists 72 customers.
- Cluster 1 consists 67 customers.
- Cluster 2 consists 71 customers.

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

## Cluster Profiles of K-Means Clustering

| Clus_kmeans3 | spending | advance_payments | probability_of_full_payment | current_balance | credit_limit | min_payment_amt | max_spent_in_single_shopping | freq |
|---|---|---|---|---|---|---|---|---|
| 0 | 11.856944 | 13.247778 | 0.848330 | 5.231750 | 2.849542 | 4.733892 | 5.101722 | 72 |
| 1 | 18.495373 | 16.203433 | 0.884210 | 6.175687 | 3.697537 | 3.632373 | 6.041701 | 67 |
| 2 | 14.437887 | 14.337746 | 0.881597 | 5.514577 | 3.259225 | 2.707341 | 5.120803 | 71 |

TAB:22 CLUSTER PROFILES OF K-MEANS CLUSTERING

## Visualization of the Clusters Profiles Across the Variables.

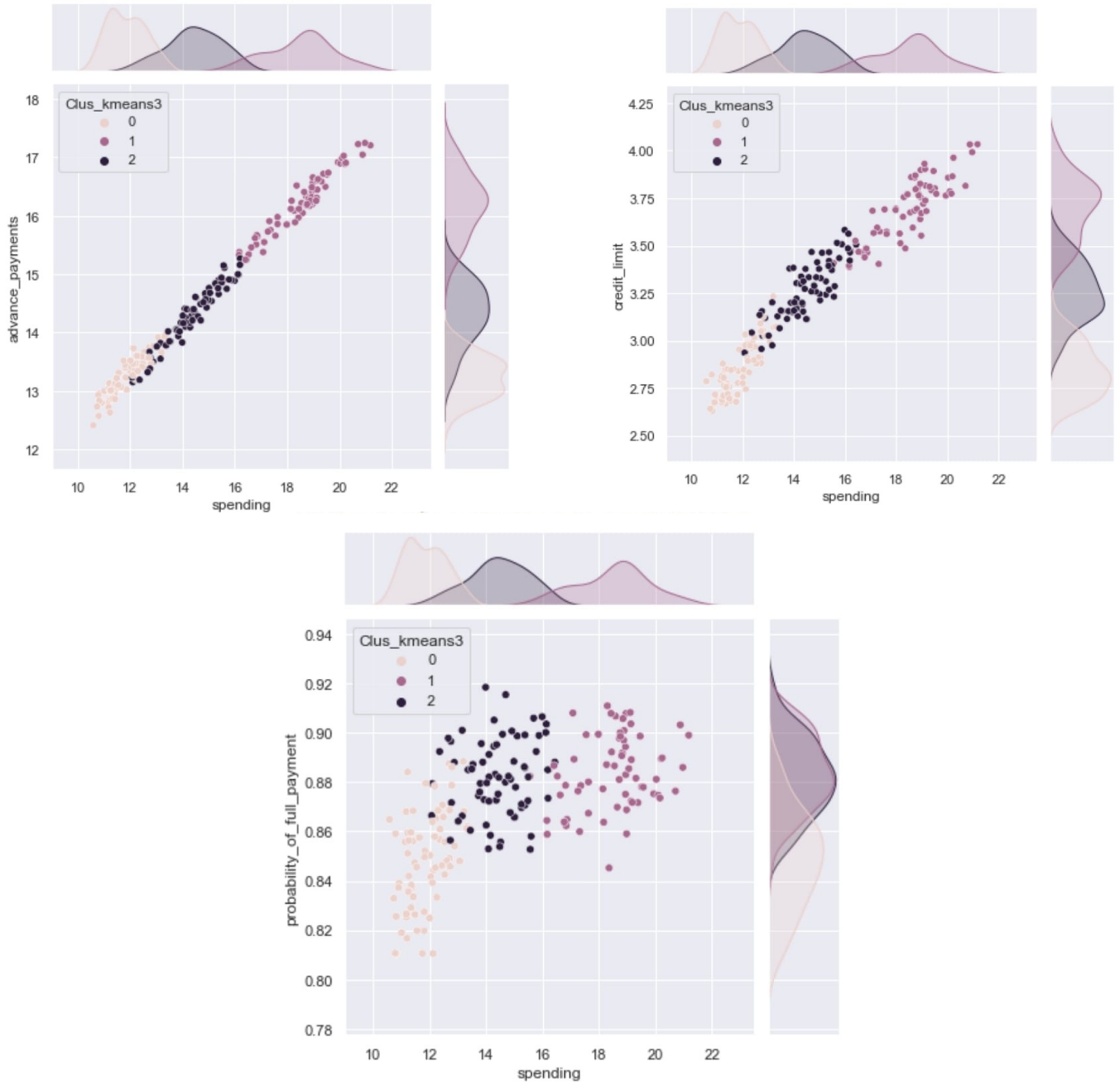



FIG:23 DISTRIBUTION OF CLUSTER PROFILES WITH VARIBALES

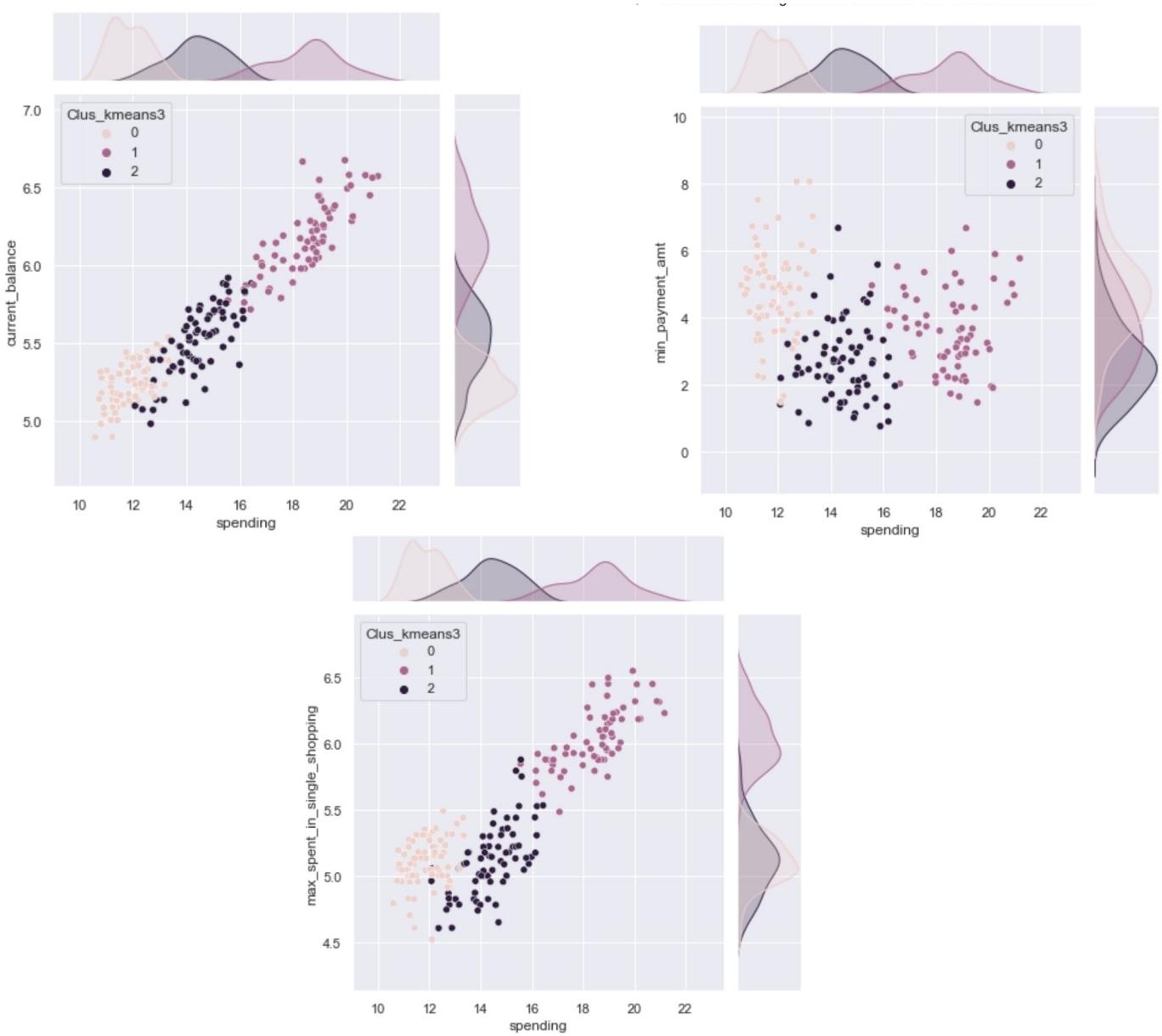



FIG:23 DISTRIBUTION OF CLUSTER PROFILES WITH VARIBALES

## Insights

As we applied two differnt clustering algorithms to do the customer segmentaion.The clusters formed were nearly identical. We can classify the cluster into 3 differnt profiles,namely:

- Cluster 0 - High risk customers with low advance_payments and probability_of_full_payment.
- These customers have low current_balance,credit_limit ,max_spent_in_single_shopping,but high min_payment_amt.
- Cluster 1 - Low risk customers with high advance_payments and probability_of_full_payment.
- These customers have high spending,current_balance,credit_limit, max_spent_in_single_shopping but less min_payment_amt.

- Cluster 2 - Moderate risk customers with high advance_payments and probability_of_full_payment as compared to the cluster 0 customers.
- These customers have high spending, current_balance, credit_limit, max_spent_in_single_shopping than cluster 0 Customers.

**Recommendations**

* The customers in Cluster 1 have high,advance_payments and probability_of_full_payment,credit_limit. So,bank can promote various insta loan schemes to these customers.

* The customers in Cluster 0 ,2 have low, credit_limit So,bank can intoduces various credit_limit enhancement schemes to these customers.

* The customers in Cluster 1 have high,spending , max_spent_in_single_shopping but higher min_payment_amt than cluster 2 customers So,bank can intoduces various min_payment_amt schemes to the cluster 1 customers.

* Maximum max_spent_in_single_shopping is high for cluster1 , so can be offered discount/offer on next transactions upon full payment.Increase their credit limit based on spending habits.Give loan against the credit card, as they are customers with good repayment record.Tie up with brands, which will drive more one_time_maximum spending.

* Cluster 2 customers are the potential target customers who are paying bills, purchasing using credit cards and maintaining comparatively good credit score. So, we can increase credit limit or can lower down interest rate as seasonal offers and study the behavior further.Promote premium/loyalty card to increase transactions.Increase spending habits by giving offers on ecommerce sites, travel portal, travel airlines/hotel, as this will encourage them to spend more.

* Customers of cluster 0 should be given remainders for payments. Offers can be provided on early payments to improve their payment rate.Offer seasonal discounts on festivals.Increase their spending habits by tying up with grocery stores, utilities (electricity, phone, gas, others)

## PROBLEM STATEMENT 2 : CART-RF-ANN

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

### Executive Summary

A Insuarnce firm deals with tour insurance. The dataset consists of various characteristics of the customers. Based on the different attributes/characteristics the customers of the Insurance firm is defined. In this problem statement we will predict the claim status and provide recommendations to management.

### Introduction

The purpose of this whole exercise is to explore the dataset. Do the exploratory data analysis & apply various supervised classification modelling techniques to predict the claim status .Explore the dataset using central tendency and other parameters. The data consists of 3000 different customers with 10 features. Analyse the different attributes of the customers which can help in analysing the claim status of the customers . This assignment should help the Insurance firm in exploring the summary statistics, supervised classification model will help insurance firm to predict the claim status of the customers.

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

EDA - Data Description , Data Preprocessing , Data Visualization , Data preparation.

## Records of the Dataset

| | Age | Agency_Code | Type | Claimed | Commision | Channel | Duration | Sales | Product Name | Destination |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 48 | C2B | Airlines | No | 0.70 | Online | 7 | 2.51 | Customised Plan | ASIA |
| 1 | 36 | EPX | Travel Agency | No | 0.00 | Online | 34 | 20.00 | Customised Plan | ASIA |
| 2 | 39 | CWT | Travel Agency | No | 5.94 | Online | 3 | 9.90 | Customised Plan | Americas |
| 3 | 36 | EPX | Travel Agency | No | 0.00 | Online | 4 | 26.00 | Cancellation Plan | ASIA |
| 4 | 33 | JZI | Airlines | No | 6.30 | Online | 53 | 18.00 | Bronze Plan | ASIA |
| 5 | 45 | JZI | Airlines | Yes | 15.75 | Online | 8 | 45.00 | Bronze Plan | ASIA |
| 6 | 61 | CWT | Travel Agency | No | 35.64 | Online | 30 | 59.40 | Customised Plan | Americas |
| 7 | 36 | EPX | Travel Agency | No | 0.00 | Online | 16 | 80.00 | Cancellation Plan | ASIA |
| 8 | 36 | EPX | Travel Agency | No | 0.00 | Online | 19 | 14.00 | Cancellation Plan | ASIA |
| 9 | 36 | EPX | Travel Agency | No | 0.00 | Online | 42 | 43.00 | Cancellation Plan | ASIA |

TAB:1 RECORDS OF THE DATASET

## Data Dictionary of the Problem Statement 2

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)
7. Destination of the tour (Destination)
8. Amount of sales of tour insurance policies (Sales)
9. The commission received for tour insurance firm (Commission)
10. Age of insured (Age)

TAB:2 DATA DICTIONARY

## Summary of the Dataset of Problem Statement 2

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

TAB:3 SUMMARY OF THE DATASET

**From the above table we can infer the -**

- count,mean, std , 25% , 50% ,75% and min & max values of the all numeric variables present in the dataset.
- Unique , top and frequency for the categorical variables present in the dataset
- There is bad values found in the dataset duration and Commision , i.e. Duration minimum value is -1 which is not possible.As we know commision is usually in the percentage form & we found here max value of commision is 210.21 and even commision have many values which are more than 100% ,which is a bad value as we know percentage can't exceeds more than 100.Thus , we need to treat & clean them.

### Shape of the dataset

| No. of Rows | No. of cols |
|---|---|
| 3000 | 10 |

TAB:4 SHAPE OF THE DATAFRAME

- Shape attribute tells us number of observations and variables we have in the data set. It is used to check the dimension of data. The insurance_data.csv data set has 3000 observations (rows) and 10 variables (columns) in the dataset.

### Appropriateness of Datatypes & Information of the Dataframe.

```
<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
```

TAB:5 APPROPRIATENESS OF DATATYPES
& INFORMATION OF THE DATAFRAME

From the above results we can see that there is no missing value present in the dataset.Their are total 3000 rows & 10 columns in this dataset,indexed from 0 to 2999. Datatypes of 2 variables are int64, datatypes of 6 variables are object and datatypes of 2 variables are float64. Memory used by the dataset: 234.5+ KB.

**Checking for Null Values**

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

TAB:6 NULL VALUES

No Null Values present in the dataset.

**Checking for Anomalies in the Dataset.**

```
array([48, 36, 39, 33, 45, 61, 37, 43, 52, 31, 23, 29, 28, 27, 44, 46, 25,
       60, 30, 40, 32, 26, 38, 42, 51, 24, 59, 41, 56, 35, 58, 73, 47, 50,
       22, 20, 53, 68, 34, 79, 19, 21, 66, 69, 57, 63, 54, 64, 71, 49, 62,
       84, 17, 55, 76, 72, 65, 67, 75, 70, 18, 77, 14, 81, 74,  8, 16, 83,
       15, 11])
```

**Age**

array(['C2B', 'EPX', 'CWT', 'JZI'], dtype=object)

**Agency Code**

array(['Airlines', 'Travel Agency'], dtype=object)

**Type**

array(['No', 'Yes'], dtype=object)

## Claimed

```
array([7.0000e-01, 0.0000e+00, 5.9400e+00, 6.3000e+00, 1.5750e+01,
       3.5640e+01, 4.6960e+01, 1.5880e+01, 5.8800e+00, 2.3760e+01,
       5.4000e+01, 5.9400e+01, 1.8200e+01, 1.7250e+01, 6.2500e+00,
       1.4000e+01, 2.5550e+01, 7.7000e+00, 1.5000e+01, 2.0380e+01,
       9.7500e+00, 1.6250e+01, 7.3800e+00, 3.1500e+00, 4.0250e+01,
       1.1550e+01, 8.3800e+00, 6.3210e+01, 4.1580e+01, 7.6400e+00,
       1.0500e+01, 2.3500e+01, 4.8300e+01, 2.2130e+01, 4.8000e+00,
       6.7500e+00, 1.1750e+01, 5.2330e+01, 5.2500e+00, 9.1000e+00,
       8.8800e+00, 8.3250e+01, 3.8000e-01, 2.6630e+01, 7.7000e-01,
       6.0000e+00, 5.3460e+01, 1.2250e+01, 1.7550e+01, 2.0000e+01,
       1.7820e+01, 7.7220e+01, 1.1880e+01, 1.2950e+01, 1.4937e+02,
       2.9700e+01, 2.7300e+01, 2.8500e+01, 4.5000e+00, 5.8500e+00,
       4.1420e+01, 2.0130e+01, 4.9730e+01, 4.2500e+00, 4.8420e+01,
       5.3800e+00, 1.3630e+01, 2.2000e+01, 1.0692e+02, 8.3160e+01,
       4.7520e+01, 6.9710e+01, 6.4800e+01, 3.6400e+01, 5.0000e+00,
       1.5400e+01, 3.8000e+00, 1.2130e+01, 1.2675e+02, 9.5700e+00,
       2.1021e+02, 3.1200e+01, 2.6600e+00, 4.0000e+00, 1.0380e+01,
       1.8000e+01, 7.2940e+01, 2.9000e-01, 2.4150e+01, 5.6300e+00,
       1.2400e+01, 1.0098e+02, 3.3800e+00, 6.1300e+00, 1.2750e+01,
       5.9800e+00, 6.5340e+01, 2.4380e+01, 1.2630e+01, 2.2040e+01,
```

## Commision

array(['Online', 'Offline'], dtype=object)

## Channel

```
array([  7,   34,    3,    4,   53,    8,   30,   16,   19,   42,  368,
        77,   23,   21,  366,    2,   40,   33,   66,   71,   25,   29,
        31,   36,   70,   11,   27,   32,   20,   90,   17,   14,    5,
        75,   35,   24,   22,  364,   13,   15,    6,   37,  365,   38,
        81,   96,   28,   44,   57,   51,   26,  367,   12,    9,  110,
        58,   65,  100,  186,   39,   48,   46,  266,   72,   61,   43,
        18,  146,   10,   59,   93,   56,   50,   99,  135,  382,   47,
        74,   64,  379,  152,  112,  380,  189,   60,   41,   89,   95,
       383,   62,  374,   55,  401,  145,   80,   83,    1,   52,  114,
        63,   49,   69,   67,  109,  105,   54,   98,  259,  239,  102,
       166,  165,  107,  385,   68,   73,  144,  126,  116,   45,  113,
       111,  378,   82,  187,  394,  147,   88,  175,   87,  402,  393,
       224,  384,  244,  148,  209,  132,  377,  235,   79,  101,  158,
       120,  376,  125,  386,  162,  375,   76,  164,   78,  428,  129,
       396,  392,  121,  203,  185,  369,   91,  119,  398,  138,  372,
       289,  216,  123,  370,  280,  115,   85,  131,  103,  198,   84,
       281,  204,  234,   86,  391,  167,  156,  202,  128,  171,  118,
       160,  180,  390,  173,   92,  191,  229,   97,  159,  431,  208,
       106,  397,   94,  130,   -1,  172,  388,  108,  215,  373,  141,
       395,    0,  232,  124,  179,  177,  122,  195,  371,  334,  273,
       153,  104,  413,  201,  217,  168,  387,  155,  278,  434,  212,
       421,  226,  163,  309,  149,  417,  197,  142,  222,  154,  218,
       389,  276,  139,  381,  190,  303,  134,  137,  419,  127, 4580,
       184,  223,  466,  416])
```

## Duration

```
       2.5130e+01, 1.4440e+01, 2.1000e+01, 5.4000e-01, 8.6300e+00,
       2.1850e+01, 1.8600e+01, 7.6700e+00, 1.9140e+01, 1.4000e-01,
       1.3200e+01, 1.9070e+01, 1.4160e+01, 5.0000e-01, 3.6100e+01,
       2.5000e-01, 4.4690e+01, 2.3750e+01, 2.1000e-01, 1.9600e+00,
       1.2540e+01, 8.1000e-01, 1.0725e+02, 7.3500e+00, 5.7130e+01,
       4.6800e+01, 3.1750e+01, 4.9600e+01, 1.2070e+01, 3.2680e+01,
       2.3400e+01, 1.3299e+02, 1.3250e+01, 1.6038e+02, 4.4500e+01,
       1.0150e+01, 8.8100e+00, 3.4380e+01, 1.8560e+01, 1.4460e+01,
       2.8000e+01, 6.7750e+01, 5.7400e+01, 9.5900e+00, 2.9000e+01,
       2.0060e+01, 3.7500e+00, 9.1300e+00, 6.9400e+00, 3.0000e+01,
       1.2380e+01, 6.8800e+00, 1.0890e+01, 3.2000e+01, 2.7000e+01,
       1.8130e+01, 2.1350e+01, 5.0600e+00, 5.8350e+01, 2.0300e+01,
       8.1300e+00, 1.9130e+01, 1.6450e+01, 2.4000e+01, 2.0000e+00,
       8.4130e+01, 9.7250e+01, 2.5000e+01, 6.3380e+01, 4.3060e+01,
       4.5500e+01, 1.7710e+01, 2.9130e+01, 2.0250e+01, 5.0250e+01,
       1.0250e+01, 2.6400e+00, 1.1860e+01, 5.8450e+01, 1.8300e+00,
       3.4000e+01, 3.9200e+00, 3.0100e+00, 4.1250e+01, 7.5000e+00,
       1.3310e+01, 3.4130e+01, 1.2450e+01, 3.1000e+01, 1.7390e+01,
       3.0450e+01, 1.3490e+01, 2.3000e+01, 9.0000e-02, 4.1270e+01,
       2.7250e+01, 1.1250e+01, 9.0090e+01, 2.8100e+00, 1.4950e+02,
       1.3750e+01, 1.2000e+01, 7.1850e+01, 1.3500e+01])
```

**Sales**

array(['Customised Plan', 'Cancellation Plan', 'Bronze Plan', 'Silver Plan', 'Gold Plan'], dtype=object)

**Product Name**

array(['ASIA', 'Americas', 'EUROPE'], dtype=object)

**Destination**

TAB:7 CHECKING FOR ANOMALIES
FOR VARIABLES IN THE DATASET

- No Anomalies found in the Dataset.

**Checking the Value counts on all the Categorical Column.**

| Agency Code | Counts |
|---|---|
| EPX | 1365 |
| C2B | 924 |
| CWT | 472 |
| JZI | 239 |

Name: Agency_Code, dtype: int64

- There are 4 Agency_Code present in the data set named as 'EPX' , 'C2B' , 'CWT' , 'JZI'.
- 1365 customers have Agency_Code 'EPX' which is the max among all 4 Agency_Code present in the data.
- 239 customers have Agency_Code 'JZI' which is the min among all 4 Agency_Code present in the data.

| Type | Counts |
|---|---|
| Travel Agency | 1837 |
| Airlines | 1163 |

Name: Type, dtype: int64

- Most of the customers prefer Travel Agency as their tour insurance firm.

| Claimed | Counts |
|---|---|
| No | 2076 |
| Yes | 924 |

Name: Claimed, dtype: int64

- 924 customers Claim their insurance.
- 2076 didn't Claim their insurance.

| Channel | Counts |
|---|---|
| Online | 2954 |
| Offline | 46 |

Name: Channel, dtype: int64

- 2954 customers choose online channel.
- Only 46 customers choose offline channel.

| Product Name | Counts |
|---|---|
| CUSTOMISED PLAN | 1136 |
| CANCELLATION PLAN | 678 |
| BRONZE PLAN | 650 |
| SILVER PLAN | 427 |
| GOLD PLAN | 109 |

NAME: PRODUCT NAME, DTYPE: INT64

- Customised Plan is the most purcahsed insurance plan by the customers with a value count of 1136.
- Gold Plan is the least purcahsed insurance plan by the customers with a value count of 109.

| Destination | Counts |
|---|---|
| ASIA | 2465 |
| Americas | 320 |
| EUROPE | 215 |

Name: Destination, dtype: int64

- Asia is most preferred Destination of the tour.
- Europe is least preferred Destination of the tour.

TAB:8 VALUE COUNTS ON ALL THE CATEGORICAL COLUMN

**Observation**

- There is no missing value & bad value present in the above categorical variables.

**Treatment of Bad Values**

As we found that Duration min value found to be inappropirate , so we need cleaned that.We know that min value for duration of the tour can't be -1. But in Duration (Duration of the tour) we found min value of -1 this has to be cleaned.

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

TAB:9 BAD VALUES IN DATASET

From above records we observe that '-1' in Duration has been enterd maybe because the data was not available or by mistake of data entry operator. However, this data has to be imputed. We can either impute it with mean/median value or make some assumption.Here we are impute this with median.

### Corrected Bad Values

| | |
|---|---|
| Age | 25 |
| Agency_Code | JZI |
| Type | Airlines |
| Claimed | No |
| Commision | 6.3 |
| Channel | Online |
| Duration | 26.5 |
| Sales | 18.0 |
| Product Name | Bronze Plan |
| Destination | ASIA |

Name: 1508, dtype: object

TAB:10 TREATMENT OF BAD VALUES IN DATASET

- We have successfully impute the bad value present in Duration with median value.

As we found that Commision have values that are inappropirate , so we need cleaned that.We know that Commision is always is % and % is not more than 100.As we found many values in Commision (The commission received for tour insurance firm) is more than 100% so we need to clean them.

| | Age | Agency_Code | Type | Claimed | Commision | Channel | Duration | Sales | Product Name | Destination |
|---|---|---|---|---|---|---|---|---|---|---|
| **133** | 44 | CWT | Travel Agency | Yes | 149.37 | Online | 366.0 | 229.80 | Silver Plan | Americas |
| **186** | 53 | CWT | Travel Agency | Yes | 106.92 | Online | 64.0 | 178.20 | Customised Plan | Americas |
| **239** | 32 | C2B | Airlines | Yes | 126.75 | Online | 90.0 | 507.00 | Silver Plan | ASIA |
| **246** | 39 | CWT | Travel Agency | Yes | 210.21 | Online | 368.0 | 323.40 | Gold Plan | Americas |
| **323** | 54 | CWT | Travel Agency | No | 100.98 | Online | 18.0 | 0.00 | Customised Plan | Americas |
| **448** | 24 | C2B | Airlines | Yes | 103.00 | Online | 166.0 | 412.00 | Silver Plan | ASIA |
| **511** | 52 | C2B | Airlines | Yes | 108.00 | Online | 367.0 | 432.00 | Silver Plan | ASIA |
| **529** | 33 | CWT | Travel Agency | Yes | 208.16 | Online | 368.0 | 320.25 | Gold Plan | Americas |
| **631** | 31 | CWT | Travel Agency | Yes | 112.86 | Online | 175.0 | 188.10 | Customised Plan | EUROPE |
| **734** | 60 | CWT | Travel Agency | Yes | 166.53 | Online | 365.0 | 256.20 | Gold Plan | Americas |
| **736** | 48 | CWT | Travel Agency | Yes | 210.21 | Online | 365.0 | 323.40 | Gold Plan | Americas |
| **795** | 30 | CWT | Travel Agency | Yes | 166.53 | Online | 375.0 | 256.20 | Gold Plan | Americas |
| **796** | 37 | C2B | Airlines | Yes | 112.31 | Online | 365.0 | 449.25 | Gold Plan | ASIA |
| **887** | 36 | C2B | Airlines | Yes | 112.31 | Online | 386.0 | 449.25 | Gold Plan | ASIA |
| **937** | 26 | C2B | Airlines | No | 122.88 | Online | 185.0 | 491.50 | Gold Plan | ASIA |
| **1079** | 37 | CWT | Travel Agency | No | 210.21 | Online | 364.0 | 323.40 | Gold Plan | Americas |
| **1155** | 44 | CWT | Travel Agency | No | 142.56 | Online | 29.0 | 237.60 | Customised Plan | Americas |
| **1202** | 24 | C2B | Airlines | No | 134.75 | Online | 175.0 | 539.00 | Gold Plan | ASIA |
| **1444** | 55 | CWT | Travel Agency | No | 106.92 | Online | 106.0 | 178.20 | Customised Plan | EUROPE |
| **1450** | 44 | CWT | Travel Agency | Yes | 208.00 | Online | 397.0 | 320.00 | Gold Plan | Americas |
| **1481** | 26 | CWT | Travel Agency | No | 124.74 | Online | 93.0 | 207.90 | Customised Plan | Americas |
| **1525** | 24 | CWT | Travel Agency | No | 124.74 | Online | 80.0 | 0.00 | Customised Plan | Americas |
| **1603** | 34 | CWT | Travel Agency | No | 166.53 | Online | 364.0 | 256.20 | Gold Plan | Americas |
| **1766** | 64 | CWT | Travel Agency | No | 124.74 | Online | 58.0 | 207.90 | Customised Plan | Americas |
| **1799** | 36 | CWT | Travel Agency | Yes | 106.92 | Online | 32.0 | 178.20 | Customised Plan | Americas |
| **1834** | 61 | CWT | Travel Agency | Yes | 166.53 | Online | 365.0 | 256.20 | Gold Plan | Americas |
| **1896** | 28 | CWT | Travel Agency | No | 166.53 | Online | 369.0 | 256.20 | Gold Plan | Americas |
| **1913** | 39 | CWT | Travel Agency | Yes | 107.25 | Online | 368.0 | 165.00 | Silver Plan | ASIA |
| **2032** | 39 | CWT | Travel Agency | No | 132.99 | Online | 369.0 | 204.60 | Gold Plan | ASIA |
| **2050** | 46 | CWT | Travel Agency | No | 160.38 | Online | 84.0 | 267.30 | Customised Plan | EUROPE |

TAB:9 BAD VALUES IN DATASET

From above records we observe that values more than'100' in Commision has been enterd maybe because the data was not available or by mistake of data entry operator. However, this data has to be imputed. We can either impute it with mean/median value or make some assumption.Here we are impute this with median.

**Corrected Bad Values**

| | |
|---|---|
| Age | 44 |
| Agency_Code | CWT |
| Type | Travel Agency |
| Claimed | Yes |
| Commision | 4.63 |
| Channel | Online |
| Duration | 366.0 |
| Sales | 229.8 |
| Product Name | Silver Plan |
| Destination | Americas |

Name: 133, dtype: object

TAB:10 TREATMENT OF BAD VALUES IN DATASET

We have successfully impute the bad value present in Commision with the median value.Now the Range of Commision is from 0 to 99.90%.

**Checking Duplicate Values.**

Number of Duplicated Row in the Dataset = 139

- As duplicated row is not useful for us , we are going to drop them by using drop fund().
- Succesfully dropped all the duplicated values from the dataset.After Removing duplicates there are 2861 Rows and 10 Columns present in the dataset.

[ 0,0,0,0,0,---------0,0,0,0,0,0,0 ]

**TAB:11 DUPLICATE ROWS**

## Univariate Analysis of Numerical Variables.

### *Histogram & Boxplot

- A histogram takes as input a numeric variable only. The variable is cut into several bins, and the number of observation per bin is represented by the height of the bar. It is possible to represent the distribution of several variable on the same axis using this technique.
- A boxplot gives a nice summary of one or several numeric variables. The line that divides the box into 2 parts represents the median of the data. The end of the box shows the upper and lower quartiles. The extreme lines show the highest and lowest value excluding outliers.

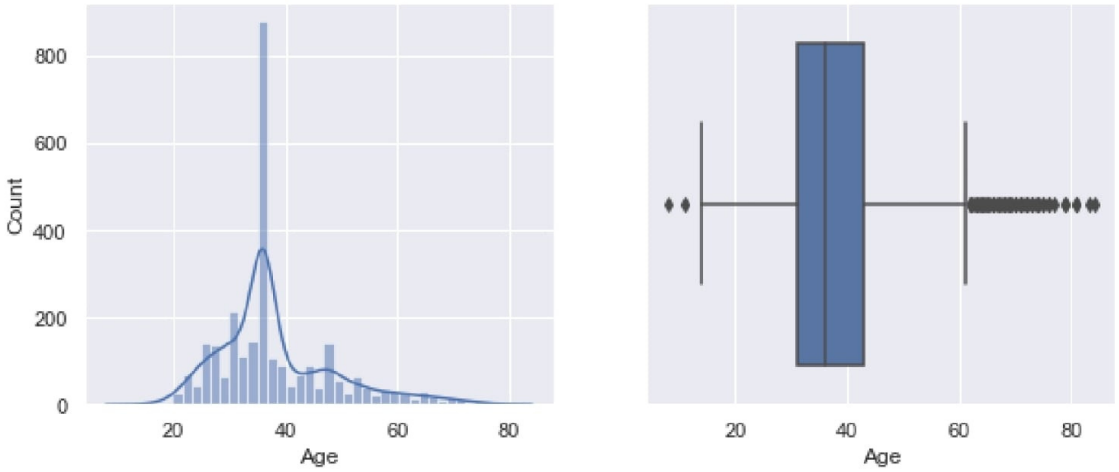



FIG:1 HISTOGRAM & BOXPLOT OF AGE

**Description**

| | |
|---|---|
| count | 2861.000000 |
| mean | 38.204124 |
| std | 10.678106 |
| min | 8.000000 |
| 25% | 31.000000 |
| 50% | 36.000000 |
| 75% | 43.000000 |
| max | 84.000000 |

Name: Age, dtype: float64

**Skewness**

1.1025661500650201

## Insight

- Age: Age of insured ranges from a minimum of 8to maximum of 84.
- The average Age:Age of insured is around 38.20.
- The standard deviation of the Age:Age of insured is 10.67.
- 25% , 50% (median) and 75 % of the Age:Age of insured are 31 , 36 and 43.
- Skewness indicating that the ditribution is slightly right skewed.
- Age:Age of insured have outliers.

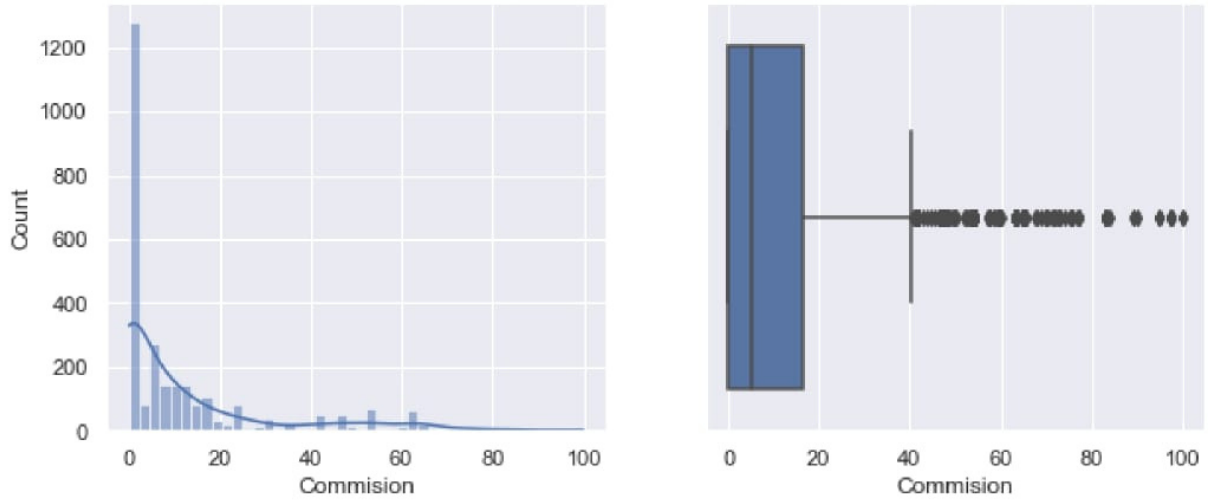



FIG:2 HISTOGRAM & BOXPLOT OF COMMISION

| Description | Skewness |
|---|---|
| COUNT 2861.000000<br>MEAN 12.992723<br>STD 19.775132<br>MIN 0.000000<br>25% 0.000000<br>50% 5.000000<br>75% 16.250000<br>MAX 99.900000<br>NAME: COMMISION, DTYPE: FLOAT64 | 1.9082464095013183 |

## Insights

- Commision: The commission received for tour insurance firm ranges from a minimum of 0 to maximum of 99.90.
- The average Commision:The commission received for tour insurance firm is around 12.99.
- The standard deviation of the Commision:The commission received for tour insurance firm is 19.77.
- 25% , 50% (median) and 75 % of the Commision:The commission received for tour insurance firm are 0 , 5 and 16.25.
- Skewness indicating that the ditribution is slightly right skewed.
- Commision:The commission received for tour insurance firm have outliers.

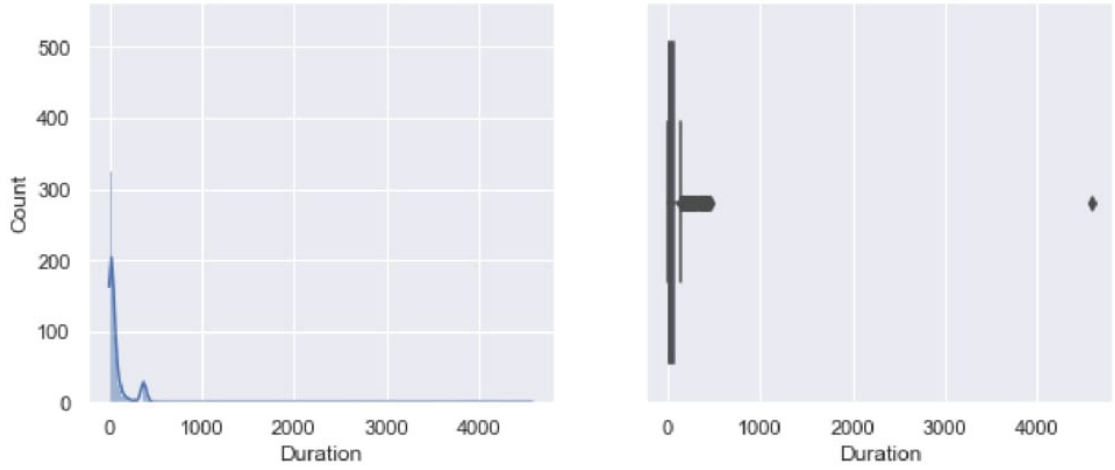



FIG:3 HISTOGRAM & BOXPLOT OF DURATION

| Description | | Skewness |
|---|---|---|
| count | 2861.000000 | 13.779972602601612 |
| mean | 72.129850 | |
| std | 135.973001 | |
| min | 0.000000 | |
| 25% | 12.000000 | |
| 50% | 28.000000 | |
| 75% | 66.000000 | |
| max | 4580.000000 | |

Name: Duration, dtype: float64

## Insights

- Duration:Duration of the tour ranges from a minimum of 0 to maximum of 4580.
- The average Duration:Duration of the tour is around 72.12.
- The standard deviation of the Duration:Duration of the tour is 135.97.
- 25% , 50% (median) and 75 % of the Duration:Duration of the tour are 12 , 28 and 66.
- Skewness indicating that the ditribution is slightly right skewed.
- Duration:Duration of the tour have outliers.

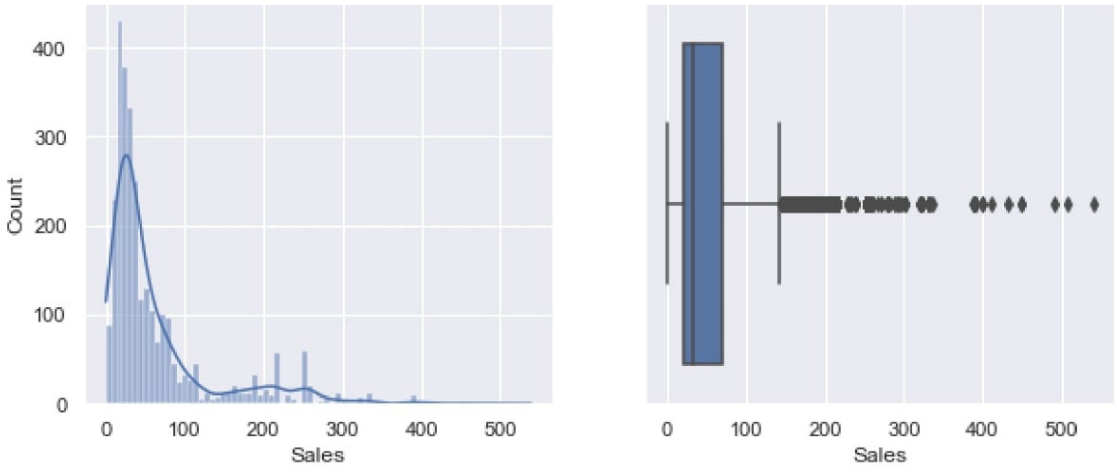



FIG:4 HISTOGRAM & BOXPLOT OF SALES

| Description | Skewness |
|---|---|
| count 2861.000000<br>mean 61.757878<br>std 71.399740<br>min 0.000000<br>25% 20.000000<br>50% 33.500000<br>75% 69.300000<br>max 539.000000<br>Name: Sales, dtype: float64<br>In [212]: | 2.3434132352067008 |

### Insights

- Sales:Amount of sales of tour insurance policies ranges from a minimum of 0 to maximum of 539.
- The average Sales:Amount of sales of tour insurance policies is around 61.75.
- The standard deviation of the Sales:Amount of sales of tour insurance policies is 71.39.
- 25% , 50% (median) and 75 % of the Sales:Amount of sales of tour insurance policies are 20 , 33 and 69.
- Skewness indicating that the ditribution is slightly right skewed.
- Sales:Amount of sales of tour insurance policies have outliers.

### Univariate Analysis of Categorical Variables.

***Countplot**

A countplot is kind of like a histogram or a bar graph for categorical variables.

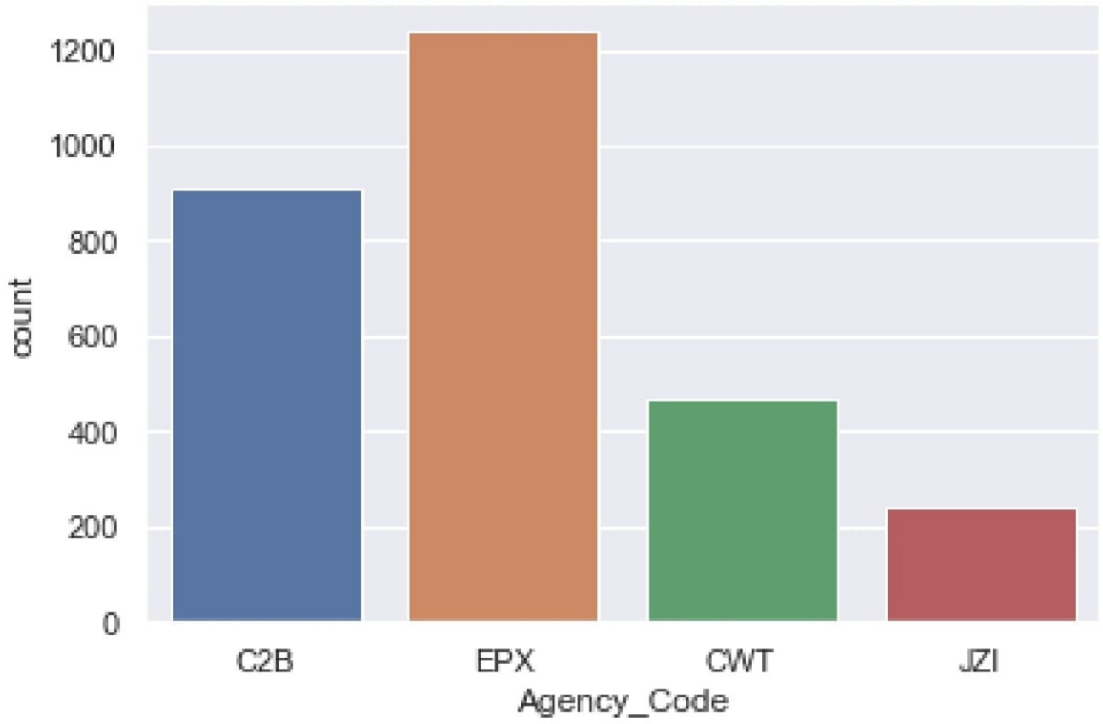



FIG:5 COUNT PLOT OF AGENCY CODE

### Description

EPX 0.432716
C2B 0.319119
CWT 0.164628
JZI 0.083537
Name: Agency_Code, dtype: float64

## Insights

- There are 4 Agency_Code present in the data set named as 'EPX' , 'C2B' , 'CWT' , 'JZI'.
- 43.27% customers have Agency_Code 'EPX' which is the max among all 4 Agency_Code present in the data.
- Only 8.3% customers have Agency_Code 'JZI' which is the min among all 4 Agency_Code present in the data.

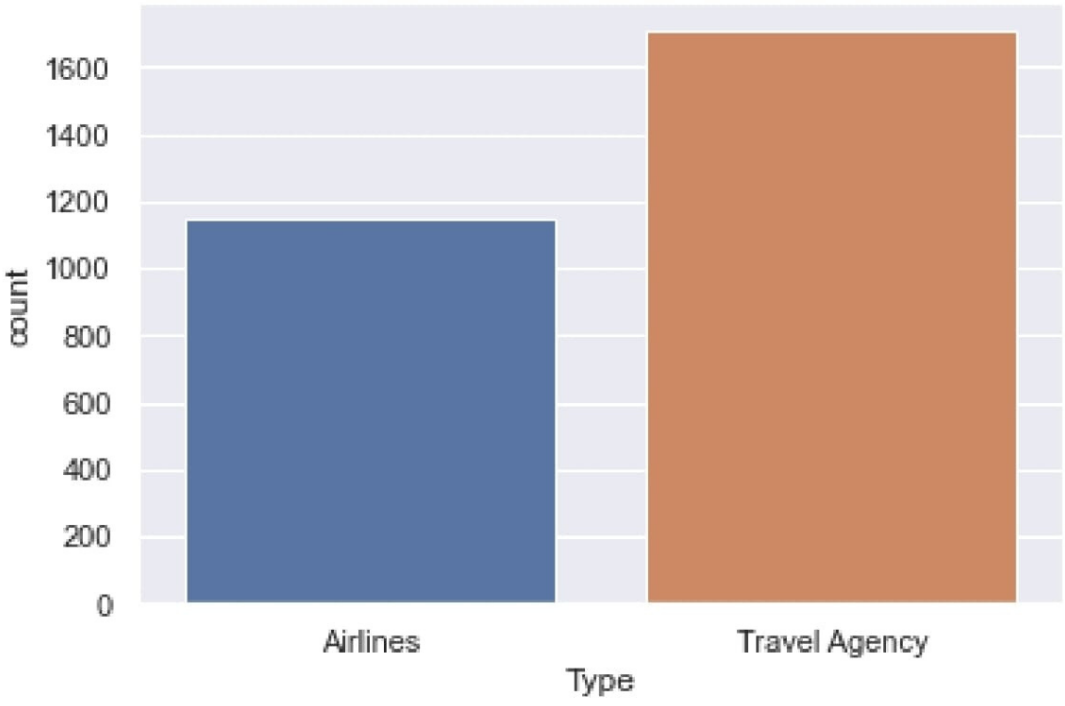



FIG:6 COUNT PLOT OF TYPE

### Description

Travel Agency 0.597344
Airlines 0.402656
Name: Type, dtype: float64

## Insights

- 59.73% customers prefer Travel Agency as their tour insurance firm.
- 40.27% customers prefer Airlines as their tour insurance firm.

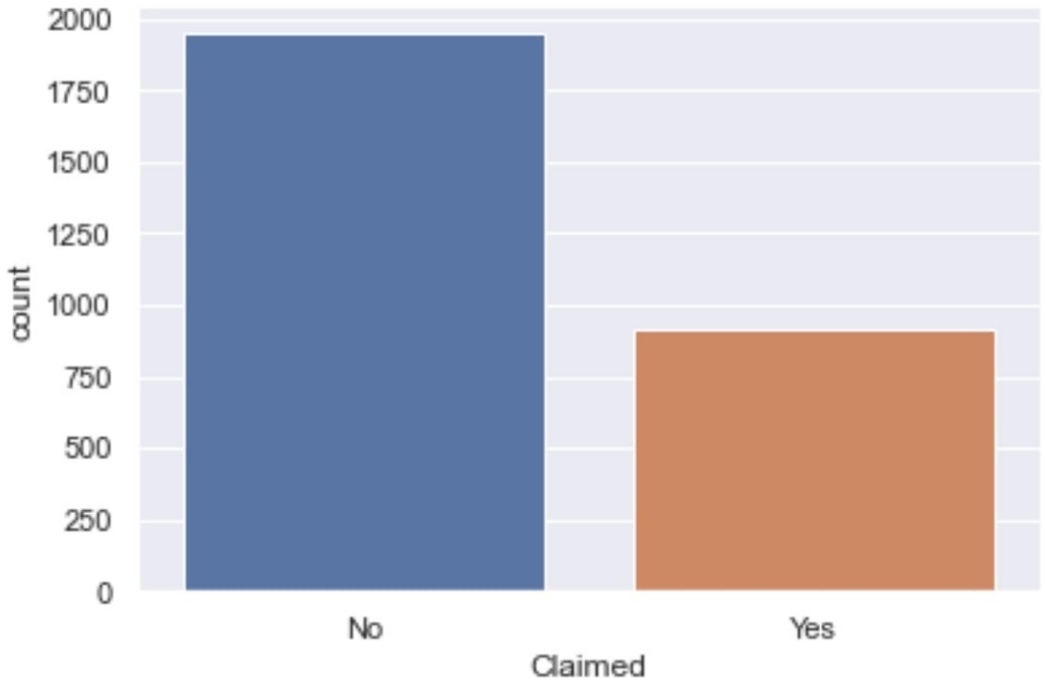



FIG:7 COUNT PLOT OF CLAIMED

## Description

No 0.680531
Yes 0.319469
Name: Claimed, dtype: float64

## Insights

- 68.05 % didn't Claim their insurance.
- 31.94 % Claim their insurance.

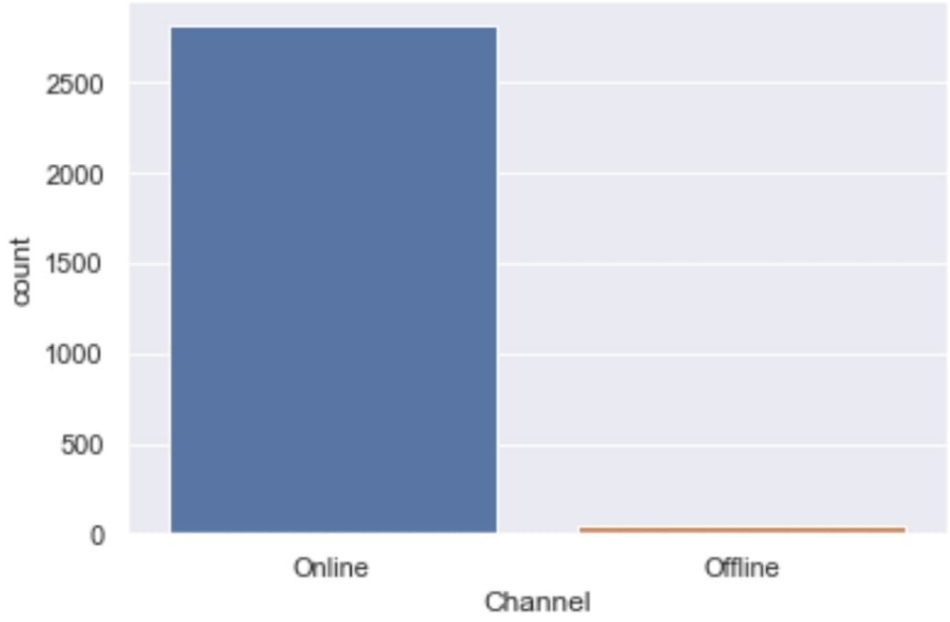



FIG:8 COUNT PLOT OF CHANNEL

## Description

Online 0.983922
Offline 0.016078
Name: Channel, dtype: float64

## Insights

- 98.4% customers choose online channel.
- Only 1.60% customers choose offline channel.

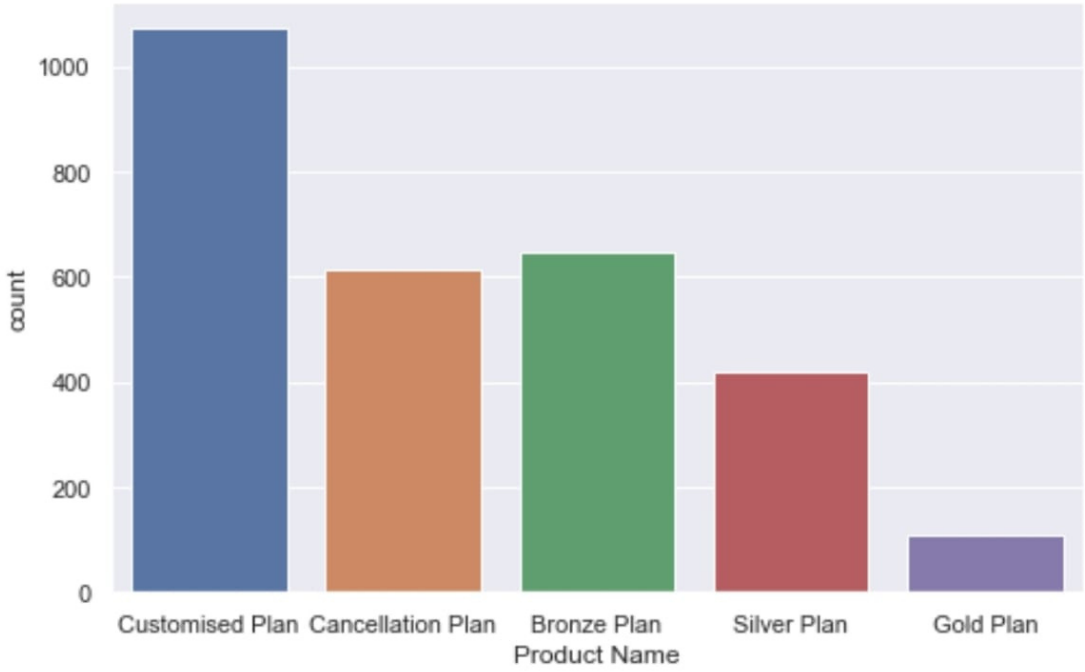



FIG:9 COUNT PLOT OF PRODUCT NAME

**Description**

Customised Plan 0.374345
Bronze Plan 0.225446
Cancellation Plan 0.214960
Silver Plan 0.147151
Gold Plan 0.038099
Name: Product Name, dtype: float64

**Insights**

- 37.43% customers purchased Customised Plan.
- 22.54% customers purchased Bronze Plan.
- 21.49% customers purchased Cancellation Plan.
- 14.71% customers purchased Silver Plan.
- Only 3.8% customers purchased Gold Plan.

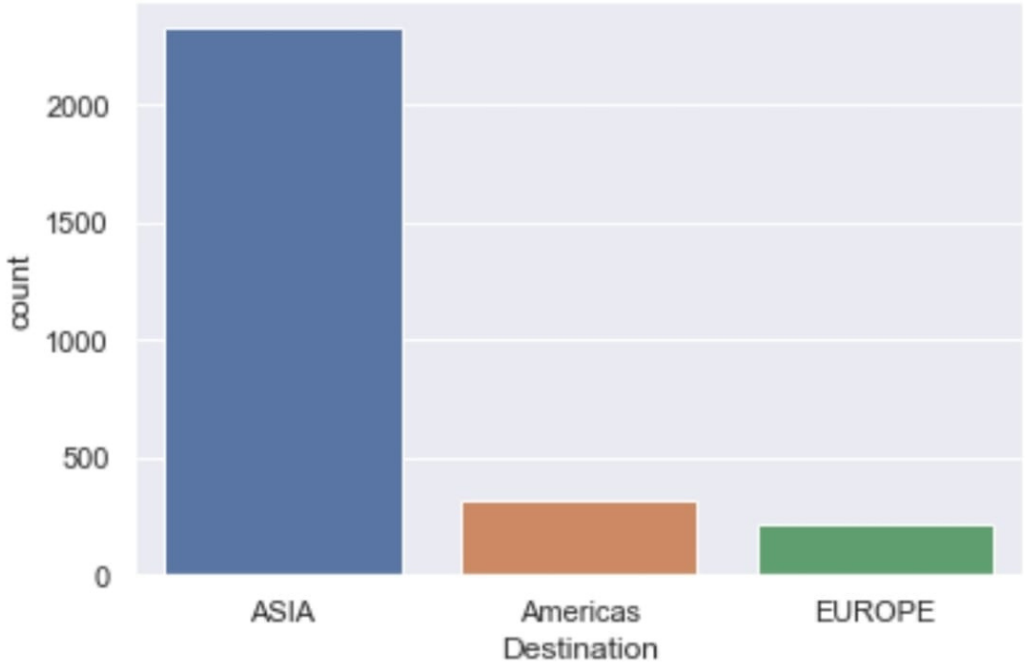



FIG:10 COUNT PLOT OF DESTINATION

**Description**

ASIA 0.813352
Americas 0.111499
EUROPE 0.075149
Name: Destination, dtype: float64

### Insights

- 81.33% customers choosed Asia as Destination of the tour.
- 11.11% customers choosed Americas as Destination of the tour.
- Only 7.5% customers choosed Europe as Destination of the tour.

### Bivariant Analysis

#### *Scatter Plot

A scatter plot (aka scatter chart, scatter graph) uses dots to represent values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables.


FIG:11 SCATTER PLOT OF SALES VS COMMISION

### Insight

- From the above plot we see that as the Sales increases the Commision is also increasing showing a positive relationship.

***Countplot with Hue.**

- A countplot is kind of like a histogram or a bar graph for categorical variables.
- Hue :This parameter take column name for colour encoding

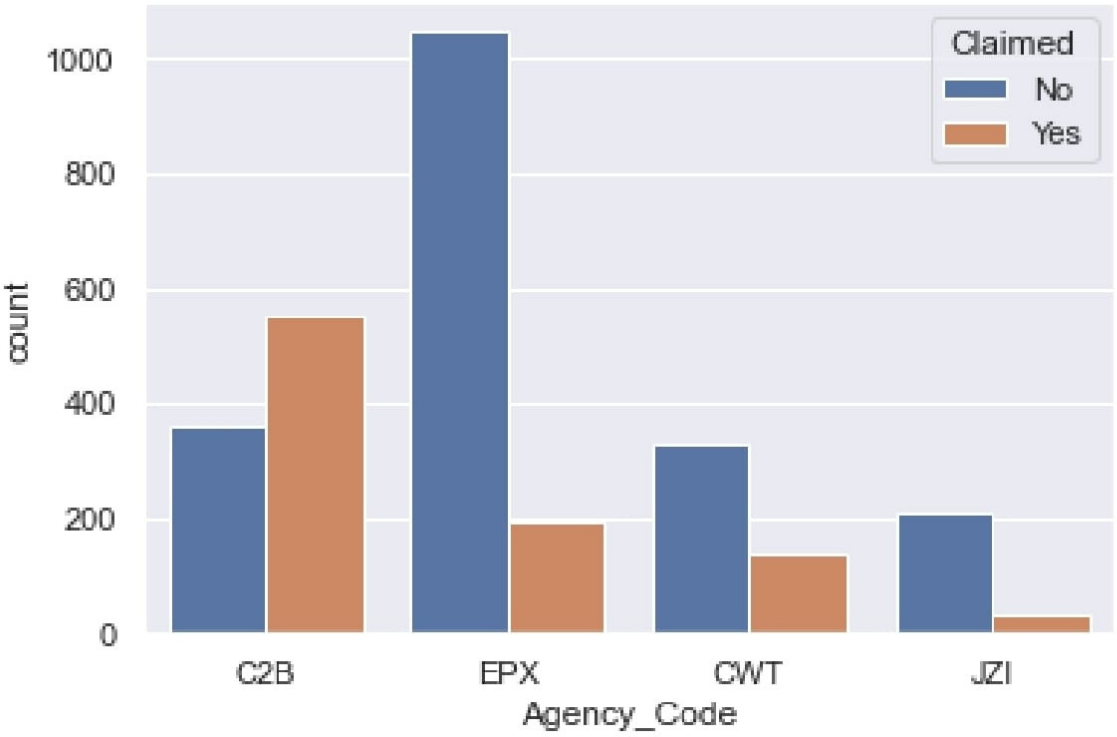



FIG:12 COUNT PLOT WITH HUE AGENCY CODE VS CLAIMED

**Insights**

- Customers with Agency Code C2B claimed more insurance.
- Most of Customers with Agency Code EPX didn't claimed insurance.
- Customers with Agency Code CWT & JZI have no claimed more than claimed.

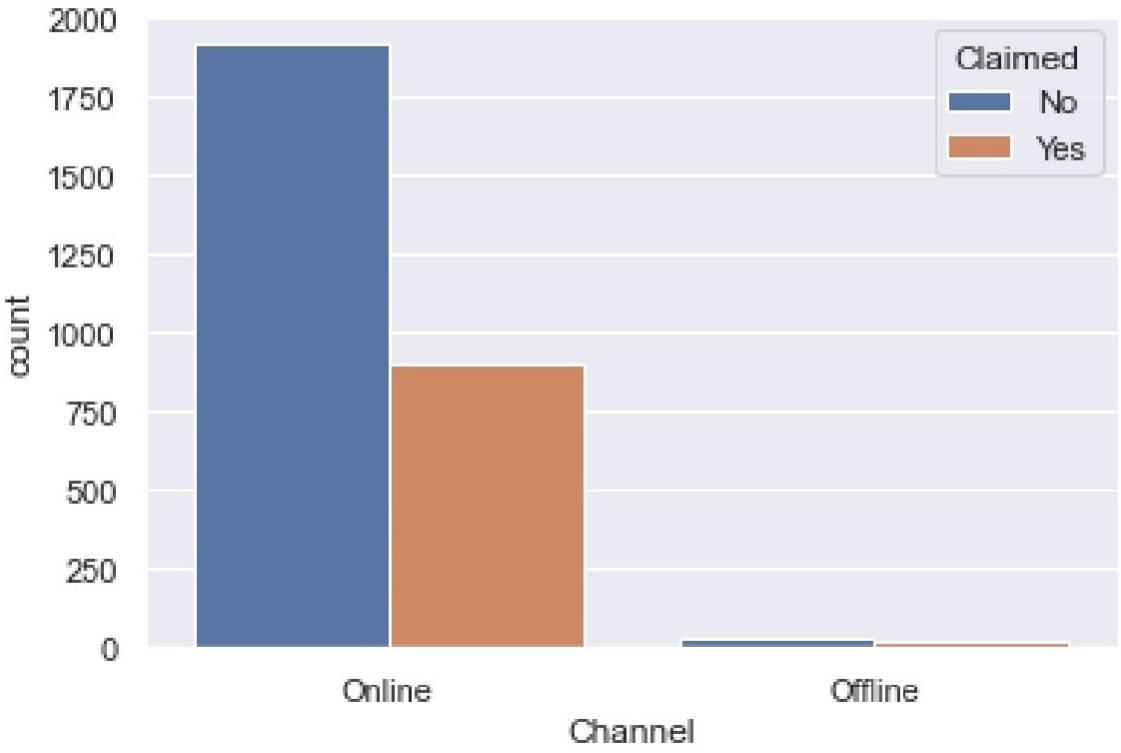



FIG:13 COUNT PLOT WITH HUE CHANNEL VS CLAIMED

## Insights

- In online channel no claimed status is more than claimed.
- In offline channel no claimed status is almost equivalent to claimed.

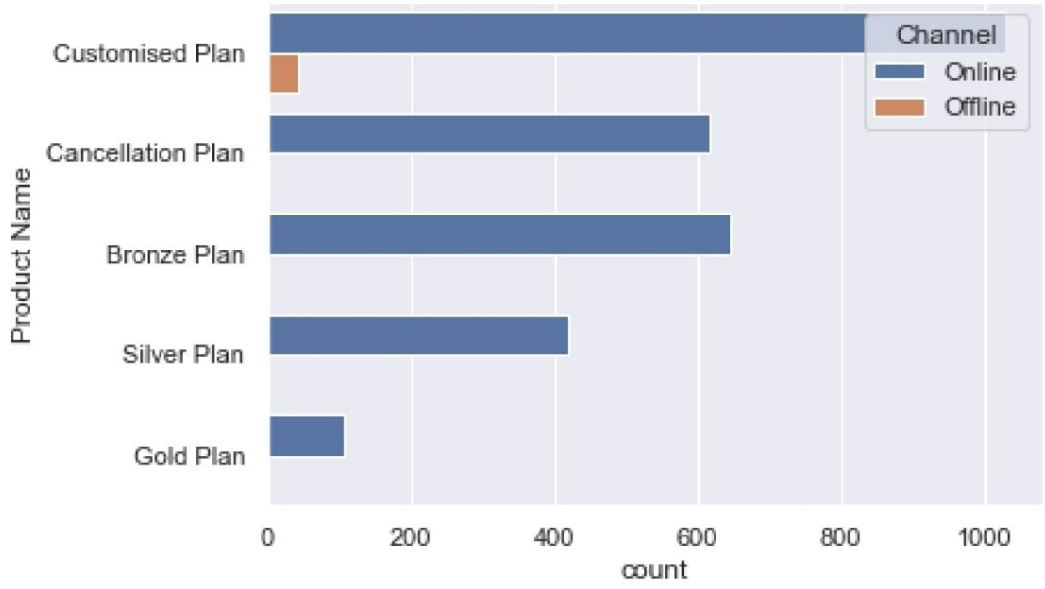



FIG:14 COUNT PLOT WITH HUE PRODUCT NAME VS CHANNEL

## Insights

- Only Customized Plan is available in offline channel for customers.

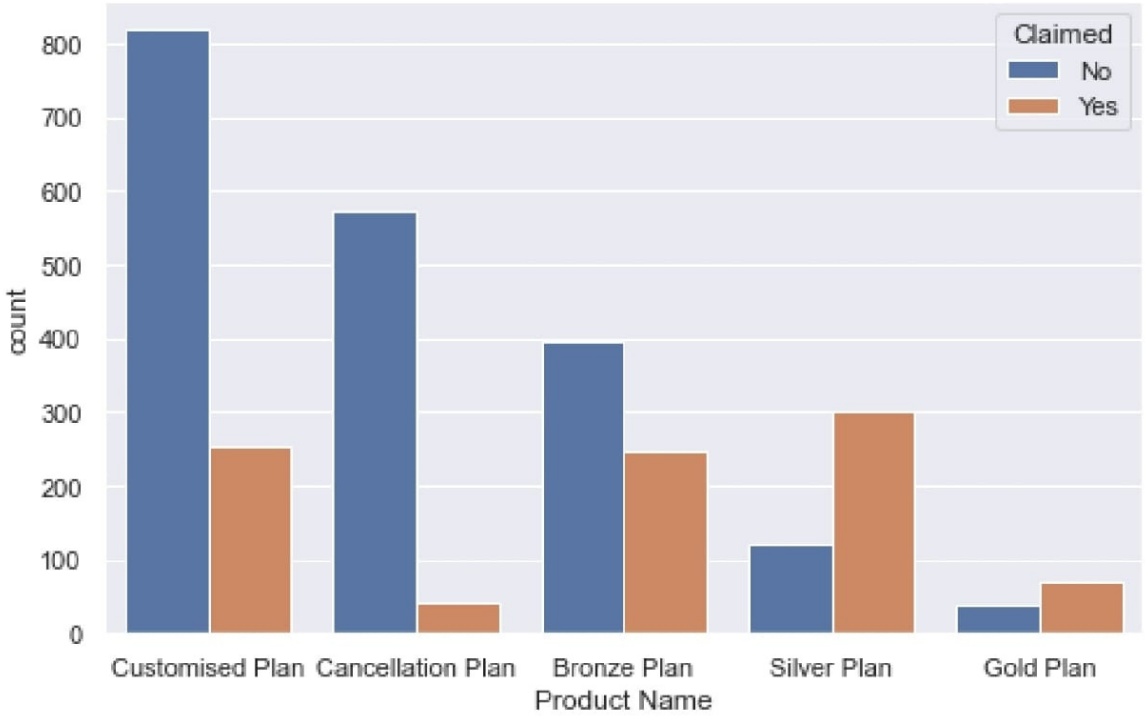



FIG:15 COUNT PLOT WITH HUE PRODUCT NAME VS CLAIMED

## Insights

- Customers with Silver and Gold Plan claimed status is more than no claimed.
- Most of customers with Customized Plan and Cancellation Plan have no claimed status.

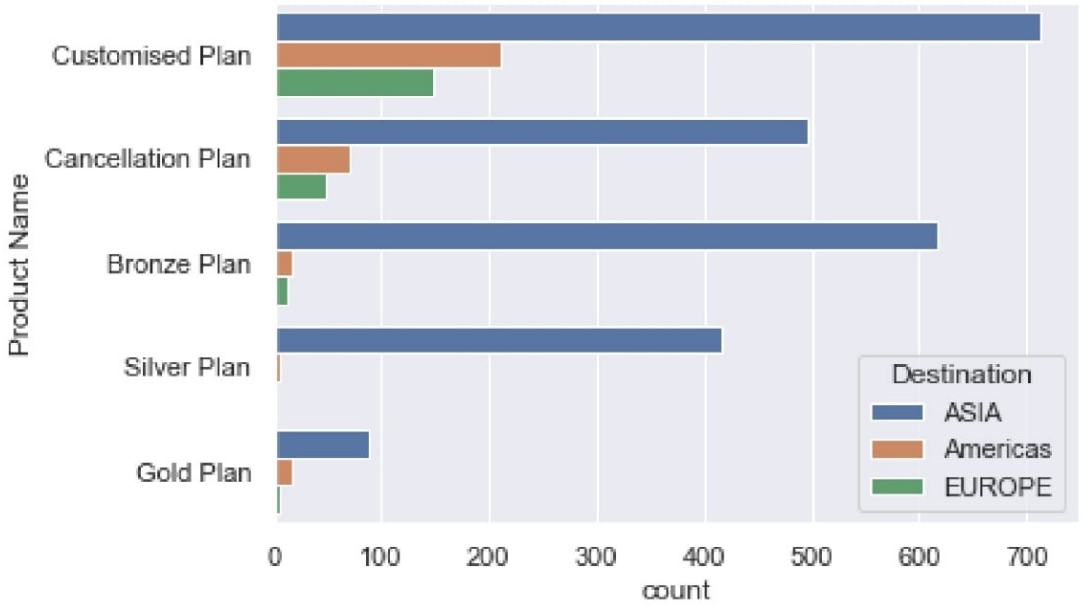



FIG:16 COUNT PLOT WITH HUE PRODUCT NAME VS DESTINATION

## Insights

- Customers whose destination is asia buys customised plan most.
- Customers whose destination is aisa buys gold plan least.
- Customers whose destination is europe buys least insuarnce plans.

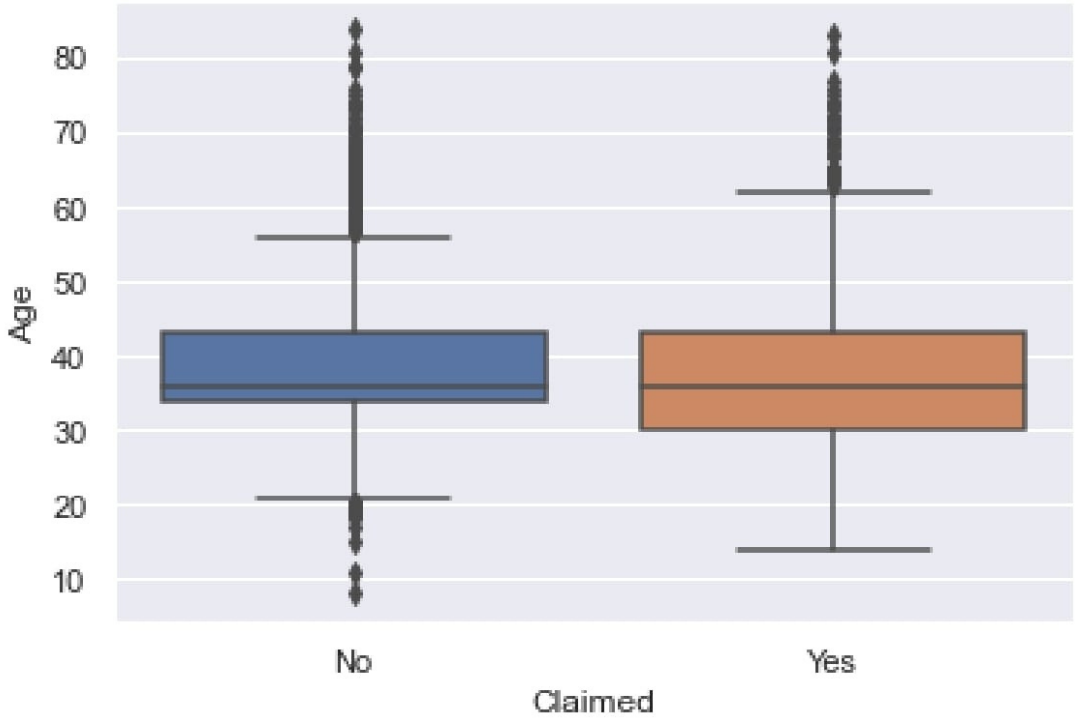



FIG:17 BOX PLOT WITH CLAIMED VS AGE

## Insights

- 50% customers who claimed the insurance are in the age level of 35-40.

## Multivariate Analysis

### * Heatmap

A correlation heatmap uses colored cells, typically in a monochromatic scale, to show a 2D correlation matrix (table) between two discrete dimensions or event types.Correlation heatmaps are ideal for comparing the measurement for each pair of dimension values.Darker Shades have higher Corealtion , while lighter shades have smaller values of correaltion as compared to darker shades values.Correlation values near to 1 or -1 are highly positively correlated and highly negatively correlated respectively. Correlation values near to 0 are not correlated to each other.

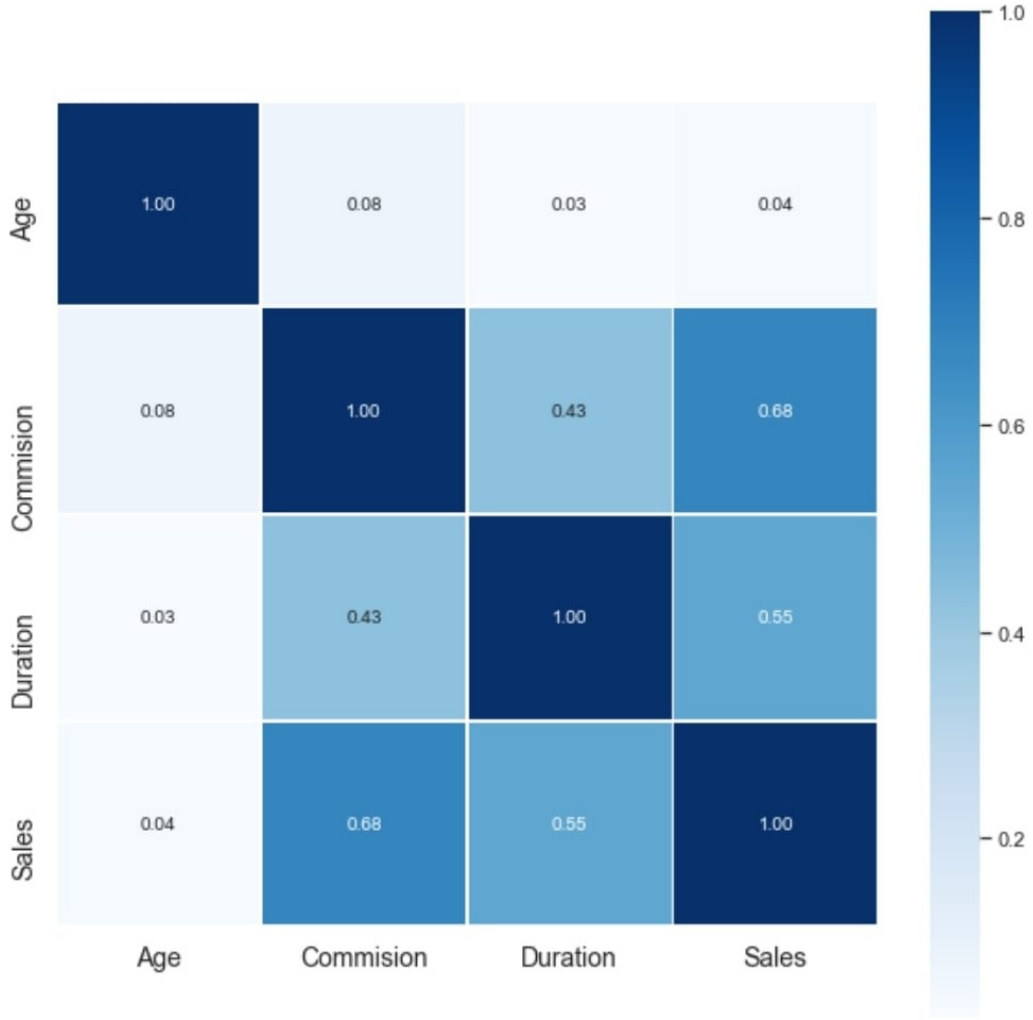



FIG:18 HEATMAP OF PROBLEM 2

**Insights**

- From the above correlation table we conclude that,
- Commision with Sales show strong correlation i.e. 0.68.
- Duration with Sales also show strong correlation i.e. 0.55.
- Duration with Age and Age with Sales show least +ve correlation among all.

***Pairplot**

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


FIG:19 PAIRPLOT OF PROBLEM 2

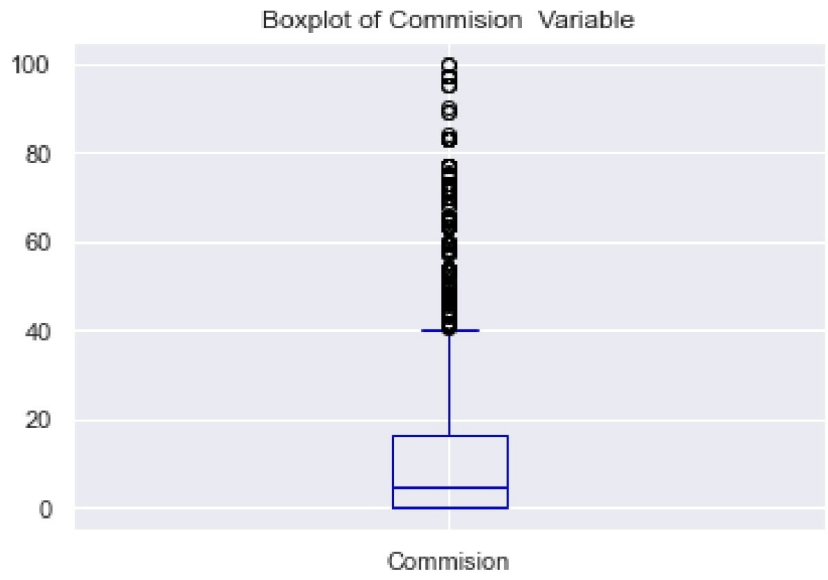
## Insights

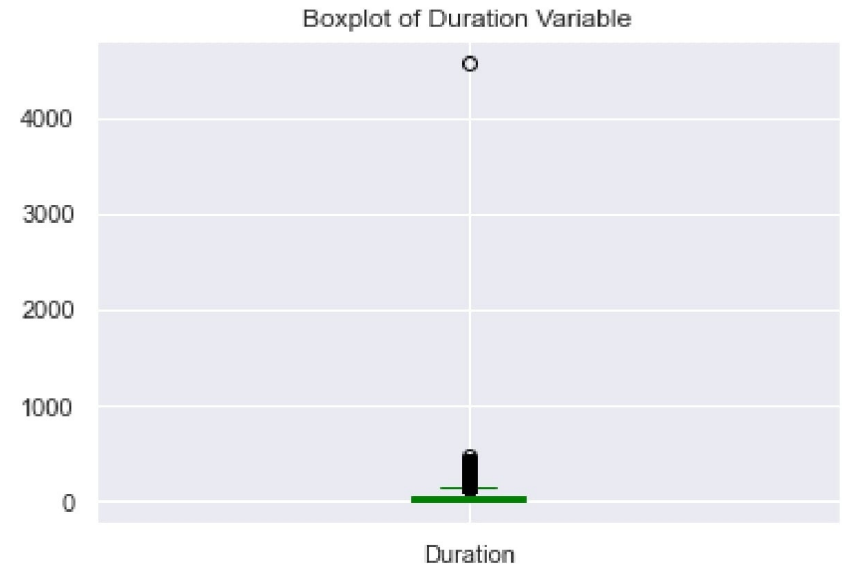
- Sales with Commision show positive relationship , Sales increases the Commision is also increasing.

## Checking for Outliers in the dataset.

To check for outliers, we will be plotting the box plots.


Boxplot of Age Variable
80
70
60
50
40
30
20
10
Age


Boxplot of Commision Variable
100
80
60
40
20
0
Commision


Boxplot of Duration Variable
4000
3000
2000
1000
0
Duration

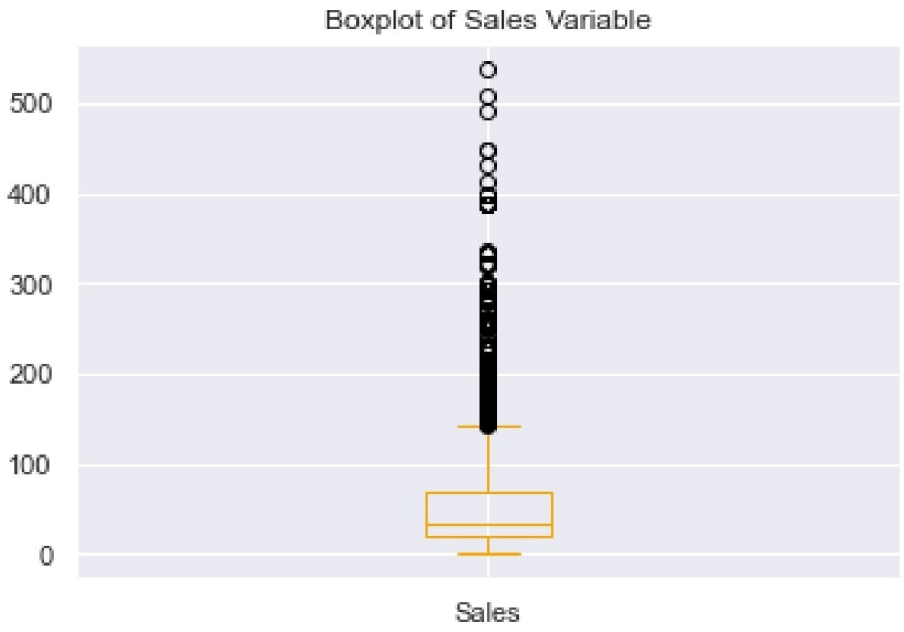



FIG:20 OUTLIER DETECTION PROBLEM 2

**Observations:**

- Looking at the box plot, it seems that the four variables Age, Commision , Duration and Sales have outlier present in the variables.

- As per the instructions - Prefer not to treat outliers here. An observation is considered to be an outlier if that particular has been mistakenly captured in the data set. Treating outliers sometimes results in the models having better performance but the models lose out on the generalization. So, a good way to approach this would be to build models with and without treating outliers and then report the results. So we are only check the outliers but not treat them as per context of the problem given.

## Encoding

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

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

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

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

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

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

**TAB:12 DATA ENCODING.**

## Checking the unique counts

Agency_Code
2 1238
0 913
1 471
3 239
Name: Agency_Code, dtype: int64

Type
1 1709
0 1152
Name: Type, dtype: int64

Claimed
0 1947
1 914
Name: Claimed, dtype: int64

Channel
1 2815
0 46
Name: Channel, dtype: int64

Product Name
2 1071
0 645
1 615
4 421
3 109
Name: Product Name, dtype: int64

Destination
0 2327
1 319
2 215
Name: Destination, dtype: int64

TAB:13 ENCODED UNIQUE COUNTS

## CHECKING ORIGINAL DATASET AFTER ENCODING

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

TAB:14 ORIGINAL DATASET AFTER ENCODING

## Checking Original Dataset Description & Info after Encoding

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2861 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Age            2861 non-null   int64  
 1   Agency_Code    2861 non-null   int8   
 2   Type           2861 non-null   int8   
 3   Claimed        2861 non-null   int8   
 4   Commision      2861 non-null   float64
 5   Channel        2861 non-null   int8   
 6   Duration       2861 non-null   float64
 7   Sales          2861 non-null   float64
 8   Product Name   2861 non-null   int8   
 9   Destination    2861 non-null   int8   
dtypes: float64(3), int64(1), int8(6)
memory usage: 193.1 KB
```

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| Age | 2861.0 | 38.204124 | 10.678106 | 8.0 | 31.0 | 36.0 | 43.00 | 84.0 |
| Agency_Code | 2861.0 | 1.280671 | 1.003773 | 0.0 | 0.0 | 2.0 | 2.00 | 3.0 |
| Type | 2861.0 | 0.597344 | 0.490518 | 0.0 | 0.0 | 1.0 | 1.00 | 1.0 |
| Claimed | 2861.0 | 0.319469 | 0.466352 | 0.0 | 0.0 | 0.0 | 1.00 | 1.0 |
| Commision | 2861.0 | 12.992723 | 19.775132 | 0.0 | 0.0 | 5.0 | 16.25 | 99.9 |
| Channel | 2861.0 | 0.983922 | 0.125799 | 0.0 | 1.0 | 1.0 | 1.00 | 1.0 |
| Duration | 2861.0 | 72.129850 | 135.973001 | 0.0 | 12.0 | 28.0 | 66.00 | 4580.0 |
| Sales | 2861.0 | 61.757878 | 71.399740 | 0.0 | 20.0 | 33.5 | 69.30 | 539.0 |
| Product Name | 2861.0 | 1.666550 | 1.277822 | 0.0 | 1.0 | 2.0 | 2.00 | 4.0 |
| Destination | 2861.0 | 0.261797 | 0.586239 | 0.0 | 0.0 | 0.0 | 0.00 | 2.0 |

TAB:14 DESCRIPTION SUMMNARY &
INFO. DATASET AFTER ENCODING

**Result**

- Label Encoding has been done and all columns are converted to number.
- After performing EDA , various data preprocessing & data preparation steps our dataset is now ready for supervised modelling algorithms like Decision Tree , RandomForest & ANN.

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

**Proportion of 1s and 0s.**

| Claimed | Proportion |
|---|---|
| 0 | 0.680531 |
| 1 | 0.319469 |

Name: Claimed, dtype: float64

TAB:15 PROPORTION OF 1S AND 0S

- There is no issue of class imbalance here as we have reasonable proportions in both the classes.

**Extracting the target column into separate vectors for training set and test set.**

- Here we separating the independent and target column for performing the modelling.

**Splitting data into training and test set.¶**

- Here we are split the data into train and test part , like x_train ,x_test ,train_labels & test_labels ,by using train_test_split funn() from sk-learn library here , we are taking 70 % data for training and 30 % data for testing.

**Checking the dimensions of the training and test data.**

| Train Data | TEST DATA |
|---|---|
| X_train (2002, 9) | X_test (859, 9) |
| train_labels (2002,) | test_labels (859,) |

TAB:16 DIMENSIONS OF THE TRAINING AND TEST DATA

## BUILDING A DECISION TREE CLASSIFIER

Grid Search for finding out the optimal values for the hyper parameters

As per the industries standards we are taking various hyper parameters to build our decision tree ,hyperparametrs are listed below.

```
'criterion': ['gini'],
'max_depth': [10,12,14,15],
'min_samples_leaf': [50,100,150],
'min_samples_split': [150,300,350],
```

TAB:17 CART MODEL GRID SEARCH HYPERPARAMETERS

### As per industry standatd

- Max_depth = 10 to 15
- min_samples_leaf' = 2-3 % of the dataset observation
- min_samples_split = 3 times of min_sample_ leaf

The best estimator for building our decision tree are obtained by using the grid search cv function() are tabluate below:

```
DecisionTreeClassifier
(max_depth=10,
min_samples_leaf=50,
min_samples_split=300,
 random_state=1)
```

TAB:18 CART MODEL GRID SEARCH
BEST ESTIMATOR

### Generating Tree

By using the best parameters will are going to build our cart model for better output.

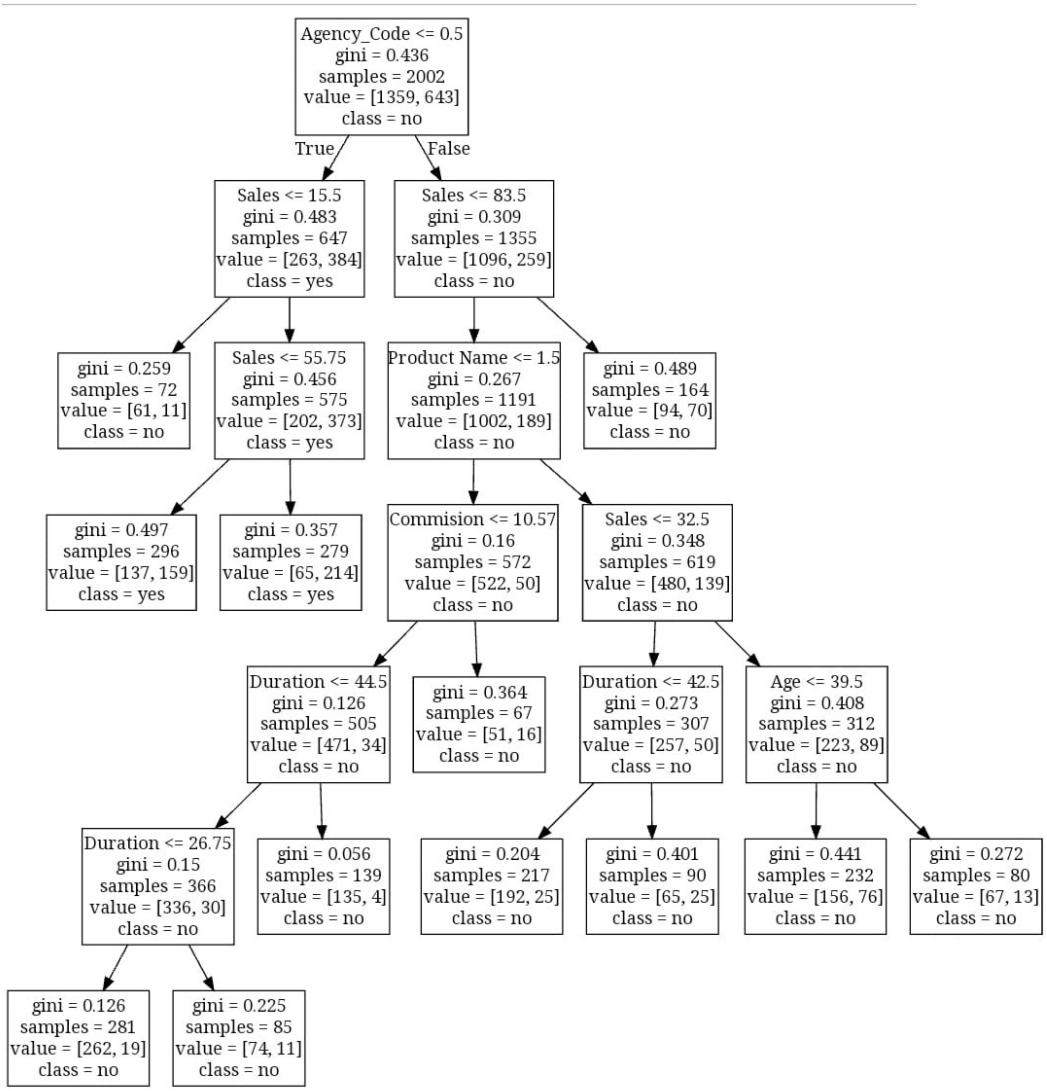



FIG: 21 DECISION TREE REGULARIZED

## VARIABLE IMPORTANCE

| variable | Imp |
|---|---|
| Agency_Code | 0.600450 |
| Sales | 0.304966 |
| Product Name | 0.047357 |
| Duration | 0.018764 |
| Commision | 0.014732 |
| Age | 0.013731 |
| Type | 0.000000 |
| Channel | 0.000000 |
| Destination | 0.000000 |

TAB:20 CART MODEL VARIABLE IMPORTANCE

## Building a Random Forest Classifier

Grid Search for finding out the optimal values for the hyper parameters

As per the industries standards we are taking various hyper parameters to build our decision tree ,hyperparametrs are listed below.

max_depth': [10,12,14,15],
'max_features': [6,7,8,9],
'min_samples_leaf': [50,100,150],
'min_samples_split': [150,300,450],
'n_estimators': [100,200]

TAB:26 RF MODEL GRID SEARCH HYPERPARAMETERS

### As per industry standatd

- max_depth = 10 to 15
- max_features'= 5 to10 to reduce correlation
- min_samples_leaf' = 2-3 % of the dataset observation
- min_samples_split = 3 times of min_sample_ leaf
- n_estimators'=100 - 500

The best estimator for building our random forest model are obtained by using the grid search cv function() are tabluate below:

max_depth': 10,
'max_features': 8,
'min_samples_leaf': 50,
'min_samples_split': 150,
'n_estimators': 100

TAB:27 RF MODEL GRID SEARCH BEST ESTIMATOR

### Variable Importance

| variable | Imp |
|---|---|
| Agency_Code | 0.511924 |
| Sales | 0.247836 |
| Product Name | 0.153181 |
| Duration | 0.038917 |
| Commision | 0.023702 |
| Age | 0.022406 |
| Destination | 0.002012 |
| Type | 0.000021 |
| Channel | 0.000000 |

TAB:28 RF MODEL VARIABLE IMPORTANCE

## Building a Artificial Neural Network Classifier

Grid Search for finding out the optimal values for the hyper parameters

As per the industries standards we are taking various hyper parameters to build our decision tree ,hyperparametrs are listed below.

```
'hidden_layer_sizes': [50,100],
    'max_iter': [2500,],
    'solver': ['adam','sgd'],
    'tol': [0.1,0.01],
```

TAB:35 ANN MODEL GRID SEARCH HYPERPARAMETERS

**As per industry standatd**

- hidden layer = 20 to 100
- max_iter'= 1000 to 5000
- solver = sgd , Adam
- tol'=0.01 ,0.1

The best estimator for building our ANN model are obtained by using the grid search cv function() are tabluate below:

```
hidden_layer_sizes=100,
max_iter=2500,
random_state=1,
tol=0.01
```

TAB:36 ANN MODEL GRID SEARCH BEST ESTIMATOR

**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.**

## CART MODEL

### Predicting on Training and Test dataset

### CART_YTRAIN_PREDICT

array([0, 0, 0, ..., 0, 0, 0], dtype=int8)

### CART_YTEST_PREDICT

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

TAB:20 CART MODEL PREDICTING ON TRAINING AND TESTING DATASET

### Getting the Predicted Probability

### CART_YTRAIN_PREDICT PROBA

| | 0 | 1 |
|---|---|---|
| 0 | 0.847222 | 0.152778 |
| 1 | 0.672414 | 0.327586 |
| 2 | 0.837500 | 0.162500 |
| 3 | 0.462838 | 0.537162 |
| 4 | 0.870588 | 0.129412 |

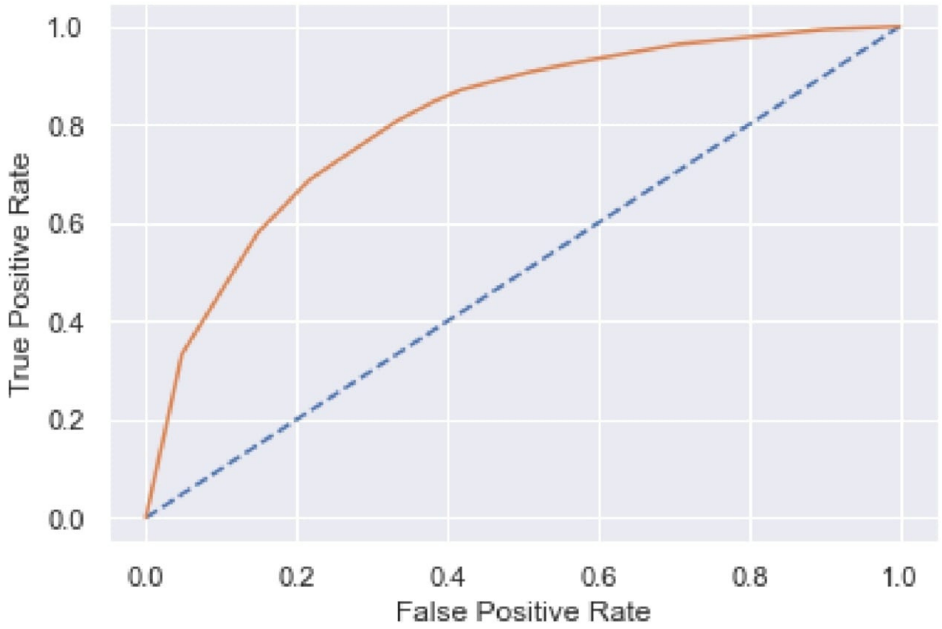
### CART_YTEST_PREDICT PROBA

| | 0 | 1 |
|---|---|---|
| 0 | 0.573171 | 0.426829 |
| 1 | 0.971223 | 0.028777 |
| 2 | 0.232975 | 0.767025 |
| 3 | 0.837500 | 0.162500 |
| 4 | 0.837500 | 0.162500 |

TAB:21 CART MODEL PREDICTED PROBABILITY

## CART MODEL EVALUATION

### CART AUC AND ROC FOR THE TRAINING DATA




**AUC: 0.810**

FIG: 22 CART MODEL - AUC AND ROC FOR THE TRAINING DATA

### Confusion Matrix for the training data

**TN [1157, 202] FP**

**,**

**FN [ 270, TP**
**373]**

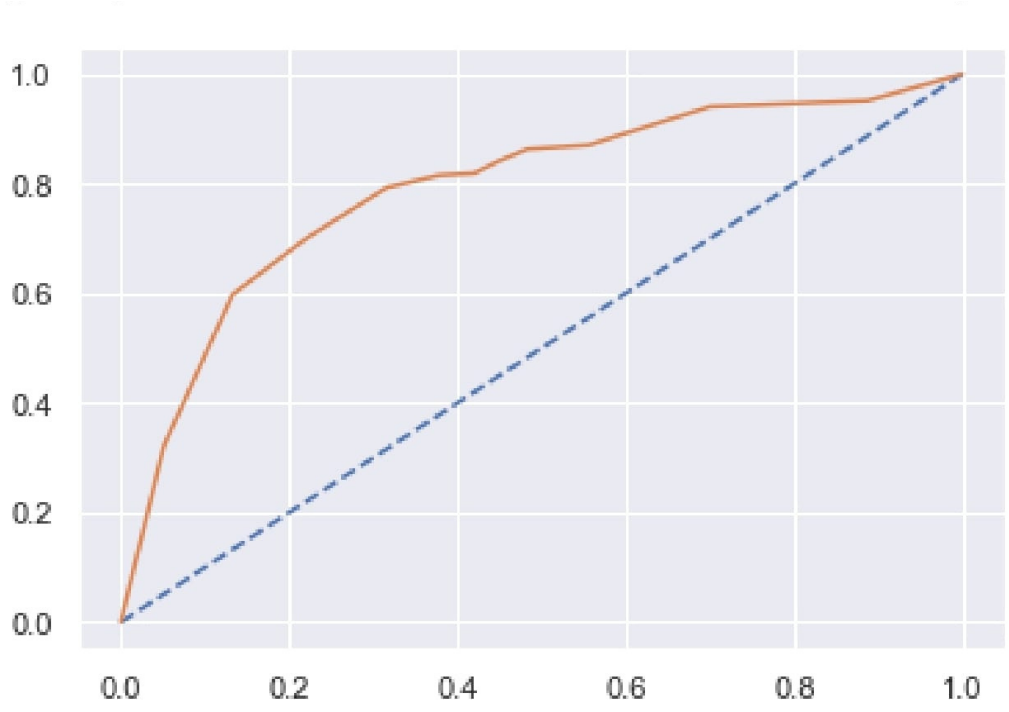
TAB:22 CART MODEL CONFUSION MATRIX FOR THE TRAINING DATA

### Classification Report of the training data

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.81 | 0.85 | 0.83 | 1359 |
| 1 | 0.65 | 0.58 | 0.61 | 643 |
| accuracy | | | 0.76 | 2002 |
| macro avg | 0.73 | 0.72 | 0.72 | 2002 |
| weighted avg | 0.76 | 0.76 | 0.76 | 2002 |

TAB:23 CART MODEL CLASSIFICATION REPORT FOR TRAINING DATA

## AUC and ROC for the test data




**AUC: 0.792**

FIG: 23 CART MODEL - AUC AND ROC FOR THE TESTING DATA

## Confusion Matrix for test data

**TN [510, 78] FP**

**,**

**FN [ 109, 162] TP**

TAB:24 CART MODEL CONFUSION MATRIX FOR THE TESTING DATA

## Classifiaction Report for the test data

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.82 | 0.87 | 0.85 | 588 |
| 1 | 0.68 | 0.60 | 0.63 | 271 |
| accuracy | | | 0.78 | 859 |
| macro avg | 0.75 | 0.73 | 0.74 | 859 |
| weighted avg | 0.78 | 0.78 | 0.78 | 859 |

TAB:25 CART MODEL CLASSIFICATION REPORT FOR TESTING DATA

### Variable Importance

| variable | Imp |
|---|---|
| Agency_Code | 0.600450 |
| Sales | 0.304966 |
| Product Name | 0.047357 |
| Duration | 0.018764 |
| Commision | 0.014732 |
| Age | 0.013731 |
| Type | 0.000000 |
| Channel | 0.000000 |
| Destination | 0.000000 |

### CART Conclusion:

Train Data:
AUC: 81.0%
Accuracy: 76%
Precision: 65%
Recall:58%
f1-Score: 61%

Test Data:
AUC: 79.2%
Accuracy: 78%
Precision: 68%
Recall:60%
f1-Score: 63%

- Training and Test set results are almost similar, and with the overall measures, the model is a good model.
- Agency_Code is the most important variable for predicting claim status.

## Random Forest Classifier Model

#### Predicting on Training and Test dataset

**RF_YTRAIN_PREDICT**

array([0, 0, 0, ..., 0, 0, 0], dtype=int8)

**RF_YTEST_PREDICT**

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

TAB:29 RF MODEL PREDICTING ON TRAINING AND TESTING DATASET

## Getting the Predicted Probability

### RF_YTRAIN_PREDICT PROBA

| | 0 | 1 |
|---|---|---|
| 0 | 0.734154 | 0.265846 |
| 1 | 0.755767 | 0.244233 |
| 2 | 0.778022 | 0.221978 |
| 3 | 0.448887 | 0.551113 |
| 4 | 0.880853 | 0.119147 |

### RF_YTEST_PREDICT PROBA

| | 0. | 1 |
|---|---|---|
| 0 | 0.554259 | 0.445741 |
| 1 | 0.922222 | 0.077778 |
| 2 | 0.289767 | 0.710233 |
| 3 | 0.750097 | 0.249903 |
| 4 | 0.695166 | 0.304834 |

TAB:30 RF MODEL PREDICTED PROBABILITY

## RF Model Performance Evaluation¶

RF AUC AND ROC FOR THE TRAINING DATA

AREA UNDER CURVE IS 0.819904055332974

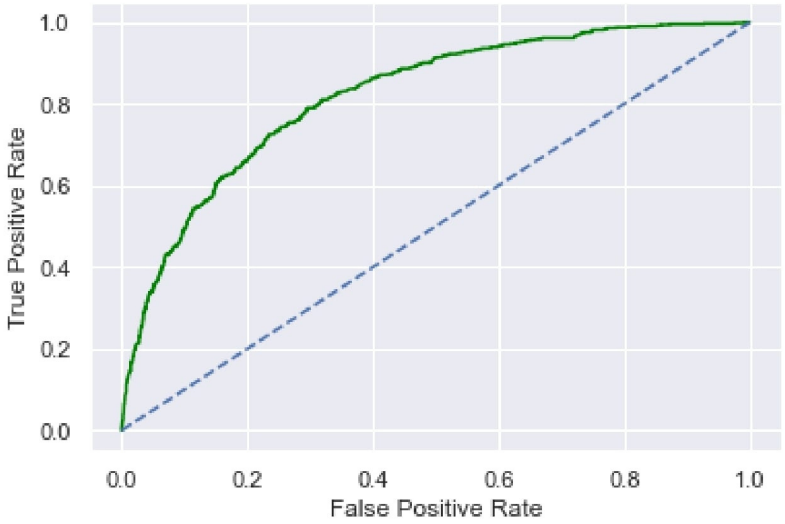



FIG: 24 RANDOM FOREST MODEL - AUC AND ROC FOR THE TRAINING DATA

## Confusion Matrix for Train data

**TN [1200, 159] FP**

**FN [ 292, 351] TP**

TAB:31 RF MODEL CONFUSION MATRIX FOR THE TRAINING DATA

## Classifiaction Report for the train data

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.80 | 0.88 | 0.84 | 1359 |
| 1 | 0.69 | 0.55 | 0.61 | 643 |
| accuracy | | | 0.77 | 2002 |
| macro avg | 0.75 | 0.71 | 0.73 | 2002 |
| weighted avg | 0.77 | 0.77 | 0.77 | 2002 |

TAB:32 RF MODEL CLASSIFICATION REPORT
FOR TRAINING DATA

## AUC and ROC for the test data

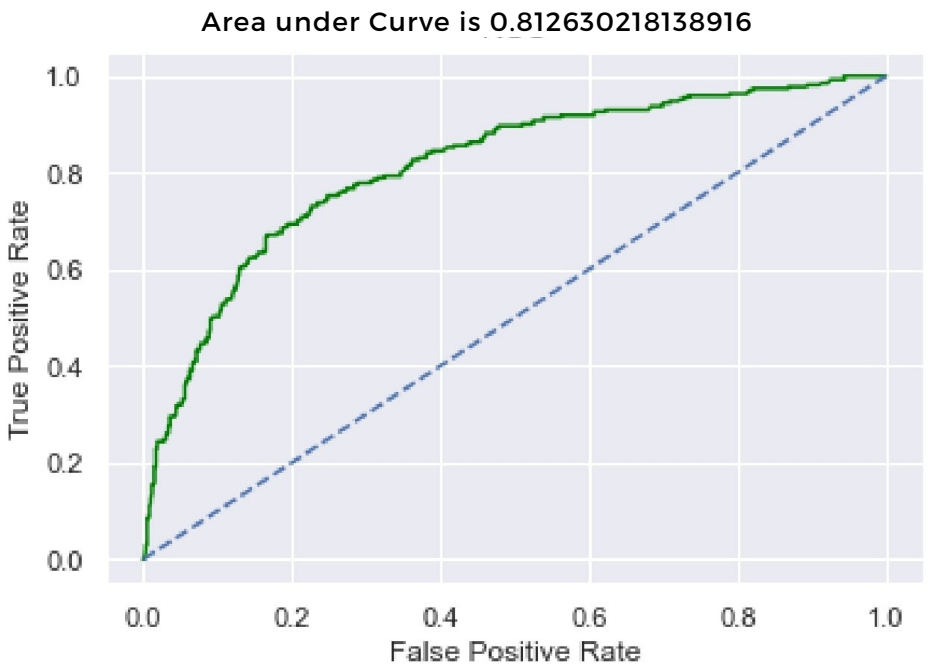



FIG: 25 RANDOM FOREST MODEL - AUC AND ROC
FOR THE TESTING DATA

## Confusion Matrix for Test data

**TN [520, 68] FP**

**FN [ 125, 146] TP**

TAB:33 RF MODEL CONFUSION
MATRIX FOR THE TESTING DATA

## Classifiaction Report for the test data

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.81 | 0.88 | 0.84 | 588 |
| 1 | 0.68 | 0.54 | 0.60 | 271 |
| accuracy | | | 0.78 | 859 |
| macro avg | 0.74 | 0.71 | 0.72 | 859 |
| weighted avg | 0.77 | 0.78 | 0.77 | 859 |

TAB:34 RF MODEL CLASSIFICATION REPORT
FOR TESTING DATA

## Variable Importance

| variable | Imp |
|---|---|
| Agency_Code | 0.511924 |
| Sales | 0.247836 |
| Product Name | 0.153181 |
| Duration | 0.038917 |
| Commision | 0.023702 |
| Age | 0.022406 |
| Destination | 0.002012 |
| Type | 0.000021 |
| Channel | 0.000000 |

## Random Forest Conclusion

Train Data:
AUC: 82%
Accuracy: 77%
Precision: 69%
Recall:55%
f1-Score: 61%

Test Data:
AUC: 81%
Accuracy: 78%
Precision: 68%
Recall:54%
f1-Score: 60%

- Training and Test set results are almost similar, and with the overall measures, the model is a good model.
- Agency_Code is the most important variable for predicting claim status.

## Artificial Neural Network Model

## Predicting the Training and Testing data

**ANN_YTRAIN_PREDICT**

array([0, 0, 0, ..., 0, 0, 0], dtype=int8)

**ANN_YTEST_PREDICT**

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

TAB:37 ANN MODEL PREDICTING ON TRAINING AND TESTING DATASET

## Getting the Predicted Probability

**ANN_YTRAIN_PREDICT PROBA**

| | 0 | 1 |
|---|---|---|
| 0 | 0.820807 | 0.179193 |
| 1 | 0.725507 | 0.274493 |
| 2 | 0.527579 | 0.472421 |
| 3 | 0.250810 | 0.749190 |
| 4 | 0.508285 | 0.491715 |

**ANN_YTEST_PREDICT PROBA**

| | 0 | 1 |
|---|---|---|
| 0 | 0.029578 | 0.970422 |
| 1 | 0.889201 | 0.110799 |
| 2 | 0.251386 | 0.748614 |
| 3 | 0.728909 | 0.271091 |
| 4 | 0.653790 | 0.346210 |

TAB:38 ANN MODEL PREDICTED PROBABILITY

## AUC and ROC for the train data

Area under Curve is 0.7914639686806578

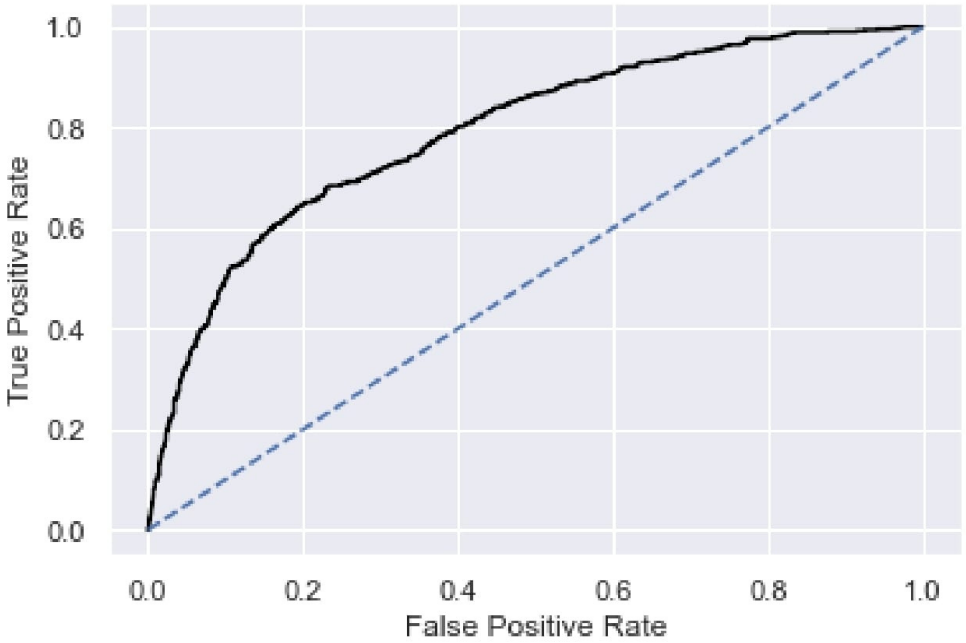



FIG: 26 ARTIFICIAL NEURAL NETWORK MODEL - AUC AND ROC FOR THE TRAINING DATA

## Confusion Matrix for Train Data

TN [1099, 260] FP

,

FN [ 234, 409] TP

TAB:39 ANN MODEL CONFUSION MATRIX FOR THE TRAINING DATA.

## Classifiaction Report for the train data

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.82 | 0.81 | 0.82 | 1359 |
| 1 | 0.61 | 0.64 | 0.62 | 643 |
| accuracy | | | 0.75 | 2002 |
| macro avg | 0.72 | 0.72 | 0.72 | 2002 |
| weighted avg | 0.76 | 0.75 | 0.75 | 2002 |

TAB:40 ANN MODEL CLASSIFICATION REPORT FOR TRAINING DATA

## AUC and ROC for the test data

Area under Curve is 0.7925734869593593

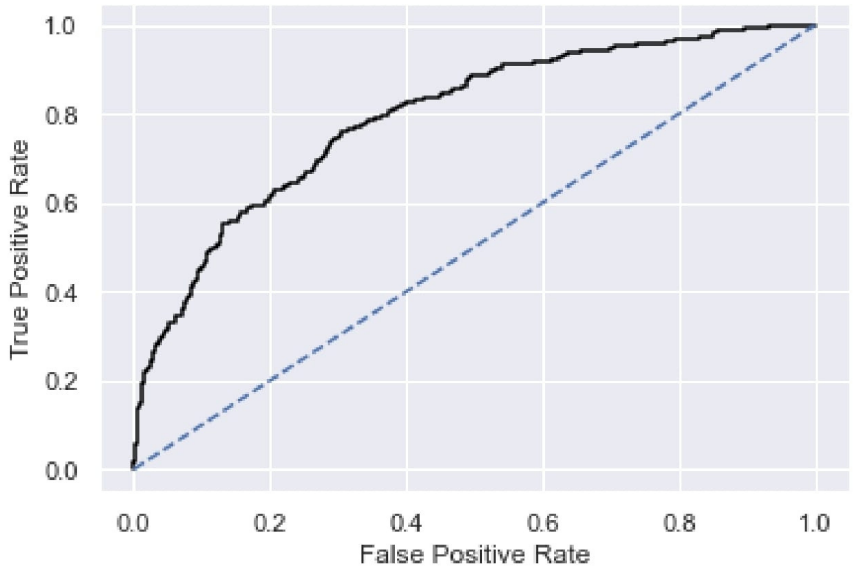



FIG: 27 ARTIFICIAL NEURAL NETWORK MODEL - AUC AND ROC FOR THE TESTING DATA

## Confusion Matrix for Test Data

TN [475, 113] FP

,

FN [ 109, 162] TP

TAB:41 ANN MODEL CONFUSION MATRIX FOR THE TESTING DATA

## Classifiaction Report for the test data

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.81 | 0.81 | 0.81 | 588 |
| 1 | 0.59 | 0.60 | 0.59 | 271 |
| accuracy | | | 0.74 | 859 |
| macro avg | 0.70 | 0.70 | 0.70 | 859 |
| weighted avg | 0.74 | 0.74 | 0.74 | 859 |

TAB:42 ANN MODEL CLASSIFICATION REPORT FOR TESTING DATA

## Artificial Neural Network Model Conclusion

Train Data:
AUC: 79.14%
Accuracy: 75%
Precision: 61%
Recall:64%
f1-Score: 62%

Test Data:
AUC: 79.25%
Accuracy: 74%
Precision: 59%
Recall:60%
f1-Score: 59%

- Training and Test set results are almost similar, and with the overall measures, the model is a good model.

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

Comparison of the performance metrics from the 3 models

| | CART Train | CART Test | Random Forest Train | Random Forest Test | Neural Network Train | Neural Network Test |
|---|---|---|---|---|---|---|
| **Accuracy** | 0.76 | 0.78 | 0.77 | 0.78 | 0.75 | 0.74 |
| **AUC** | 0.81 | 0.79 | 0.82 | 0.81 | 0.79 | 0.79 |
| **Recall** | 0.58 | 0.60 | 0.55 | 0.54 | 0.64 | 0.60 |
| **Precision** | 0.65 | 0.68 | 0.69 | 0.68 | 0.61 | 0.59 |
| **F1 Score** | 0.61 | 0.63 | 0.61 | 0.60 | 0.62 | 0.59 |

TAB:43 COMPARISON OF THE PERFORMANCE METRICS FROM THE 3 MODELS

## ROC Curve for the 3 models on the Training data

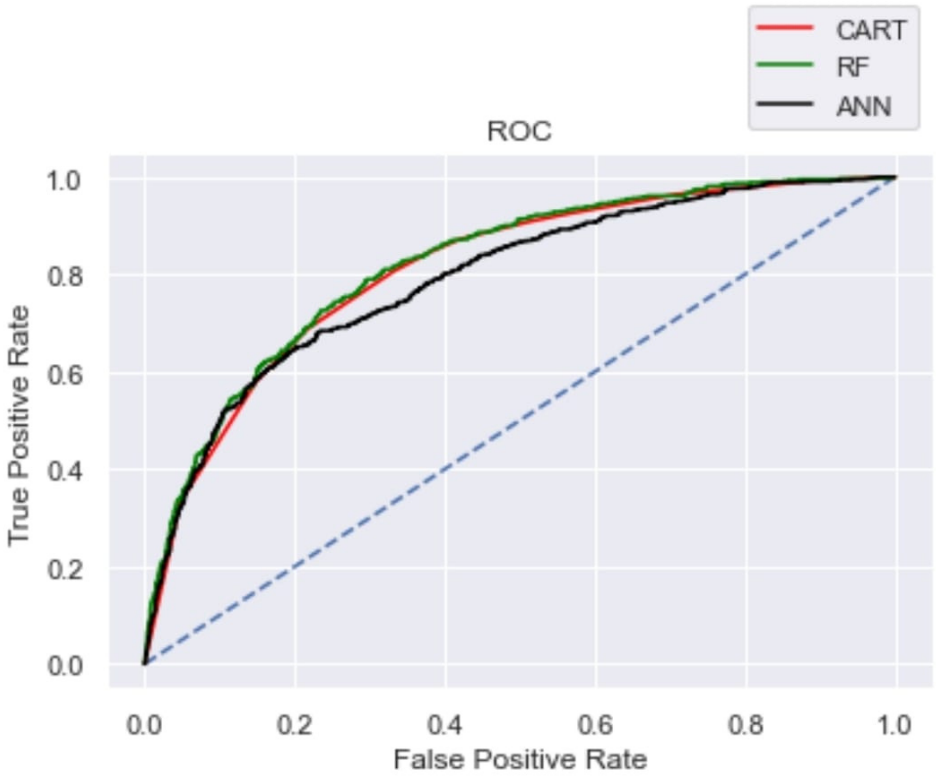



FIG: 28 ROC CURVE FOR THE 3 MODELS ON THE TRAINING DATA

**ROC Curve for the 3 models on the Test data**

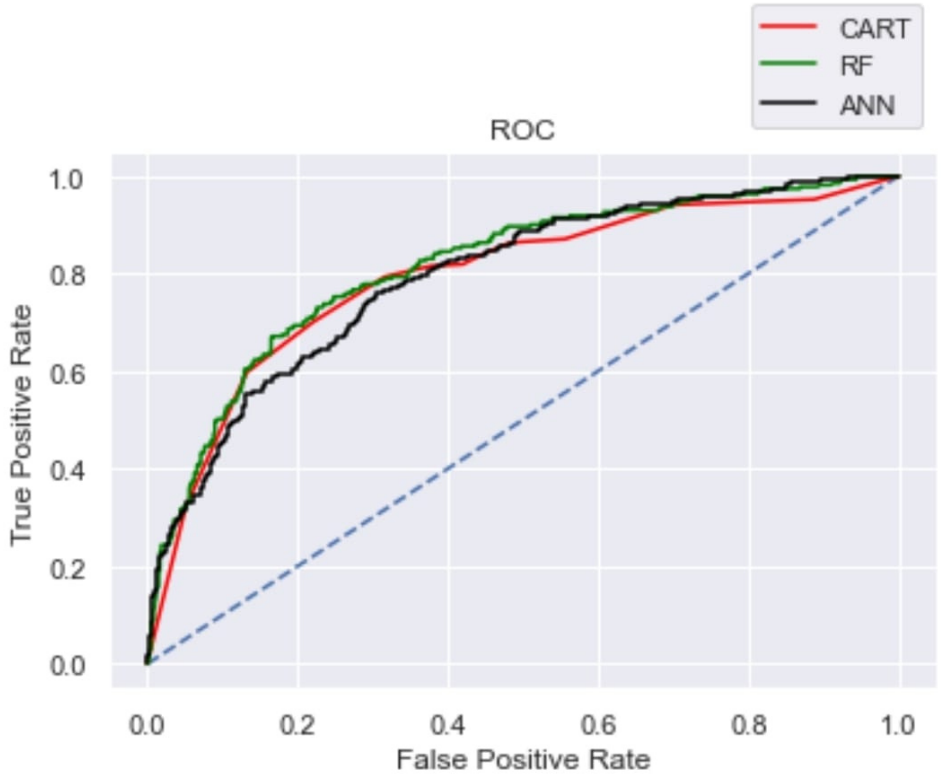



FIG: 29 ROC CURVE FOR THE 3 MODELS ON THE TESTING DATA

**Conclusion:**

- Out of the 3 models, Random Forest has slightly better performance than the Cart and Artificial Neural network model.
- Overall all the 3 models are reasonaly stable enough to be used for making any future predictions. From Cart and Random Forest Model, the variable Agency_Code is found to be the most useful feature amongst all other features for predicting if a customer has claim insurance or not.

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

**Insights**

- Agency_Code is found to be the most useful feature amongst all other features for predicting if a customer has claim insurance or not.
- 43.27% customers have Agency_Code 'EPX' which is the max among all 4 Agency_Code present in the data.
- Only 8.3% customers have Agency_Code 'JZI' which is the min among all 4 Agency_Code present in the data.
- 59.73% customers prefer Travel Agency as their tour insurance firm.
- 40.27% customers prefer Airlines as their tour insurance firm.
- 98.4% customers choose online channel for doing their insurance.i.e. online medium has made most of the sale.
- Only 1.60% customers choose offline channel for doing their insurance
- 81.33% customers choosed Asia as Destination of the tour.
- we see that as the Sales increases the Commision is also increasing showing a positive relationship.

**RECOMMENDATIONS**

- Need to train the JZI agency resources to pick up sales as they are in bottom, need to run promotional marketing campaign or evaluate if we need to tie up with alternate agency in order to attract more customers.
- Also based on the model we are getting 75-78% accuracy, so we need customer books airline tickets or plans, cross sell the insurance based on the claim data pattern.
- Other interesting fact is more sales happen via Agency than Airlines and the trend shows the claim are processed more at Airline. So we may need to deep dive into the process to understand the workflow and why?
- Key performance indicators (KPI) The KPI's of insurance claims are:
- Increase customer satisfaction which in fact will give more revenue.
- Combat fraud transactions, deploy measures to avoid fraudulent transactions at earliest.
- Optimize claims recovery method.
- Reduce claim handling costs.

**THE END !**