DATA MINING

Market Segmentation Analysis - Clustering

Insurance Analysis - CART-RF-ANN

Market Analytics – Clustering

Table of Contents

Market Segmentation Analysis :

Problem Statement 1:

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.

Data Dictionary for Market Segmentation:

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

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

The objectives of EDA can be summarized as follow:

- Maximize insight into the data/understand the data structure.
- EDA is an approach to analyse date using non-visual and visual techniques.
- EDA involves through analyse of data to understand the current business situation.
- EDA objective is to extract "'Gold' from the "'Data mine'' based on domain understanding.

As a first step, importing all the necessary libraries, we think that will be requiring to perform the EDA.

Loading the data set **–** Loading the ' **bank_marketing_part1_Data.csv '** file using pandas. For this we will be using read excel file.

EDA Exploration: Following is the output from Jupyter.

Head of the dataset: After reading the CSV file, the head command with Transpose option gives the bellow output.

	0	1	2	3	4
spending	19.9400	15.9900	18.9500	10.8300	17.9900
advance_payments	16.9200	14.8900	16.4200	12.9600	15.8600
probability_of_full_payment	0.8752	0.9064	0.8829	0.8099	0.8992
current_balance	6.6750	5.3630	6.2480	5.2780	5.8900
credit_limit	3.7630	3.5820	3.7550	2.6410	3.6940
min_payment_amt	3.2520	3.3360	3.3680	5.1820	2.0680
max_spent_in_single_shopping	6.5500	5.1440	6.1480	5.1850	5.8370

Table 1 : Sample Data

Shape of the dataset : Output from shape command is –

```
The dataset has 210 rows and 7 columns
```

info() is used to check the Information about the data and the datatypes of each respective attributes: Output from Info command is –

```
<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
```

Null value check : Output from isnull with sum command is –

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

Duplication check : Output from duplicated with sum command is –

```
The dataset has 0 duplication.
```

Descriptive Analytics : Describe method will help us see how data is spread for the numerical values, also we can see the minimum value, mean values, different percentile values and maximum values.

Output from Describe with Transpose option is –

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

Table 2 : Descriptive Statistics

Following insights are observed from the output we got from Jupyter for EDA :

- Dataset contains of 210 rows and 7 Columns.
- There are no missing values present in the dataset and all datatypes of variables are float64.
- There are no null values.
- There are no duplicated values present in the dataset.
- There are no bad values present in the dataset.
- The indexes of columns are spending, advance_payments, probability_of_full_payment, current balance, credit_limit, min_payment_amt, max_spent_in_single_shopping.
- An average **spending** amount by the customer per month (in 1000s) is around 14.84

- Per month (in 1000s) customer made an **advance payment** by case in an average of 14.55 and it ranges from minimum of 12.41 and maximum of 17.25.
- **Probability of payment** done in full by the customer to the bank in an average of 0.87.
- Average **current balance** left in the account to make purchases (in 1000s) by the customer in an average of 5.62 and customer current balance ranges from minimum of 4.90 and maximum of 6.67.
- **Minimum amount paid** by the customer while making payments for purchases made monthly (in 100s) ranges from minimum of 0.77 and maximum of 8.46 and its average is 3.70.
- In an average, the **credit limit** (in 10000s) of the customer is 3.26. Customer credit limit ranges from minimum of 2.56 and maximum of 8.46.
- **Maximum amount** spent in one purchase (in 1000s) by a customer in an average of 5.40 and customer maximum spent amount ranges from minimum of 5.22 and maximum of 6.55.
- There are considerable number of variables that are highly correlated to each other.
- There is no drastic difference of values seems in Upper and Lower range of most of the variables, this indicates that the presence of outliers will be minimal.
- 'Spending' and 'Advance_payments' are highly correlated with each other.

Univariant Analysis :- Histogram & Boxplot

The objective of univariant analysis is to derive the data, define, analyze and summarize the pattern present in it. In a dataset, it explores each variable separately such as Numerical variable and Categorical variable. Some of the patterns that can be easily identified with univariant analysis are Central Tendency (mean, mode and median), Dispersion (range, variance), Quartiles (interquartile range), and Standard deviation. Univariant analysis can be described and visualize with the help of most used plots of Histogram/Distplot and Barplot.

Column : Spending (Amount spent by the customer per month (in 1000s)

Skewness of spending: 0.40

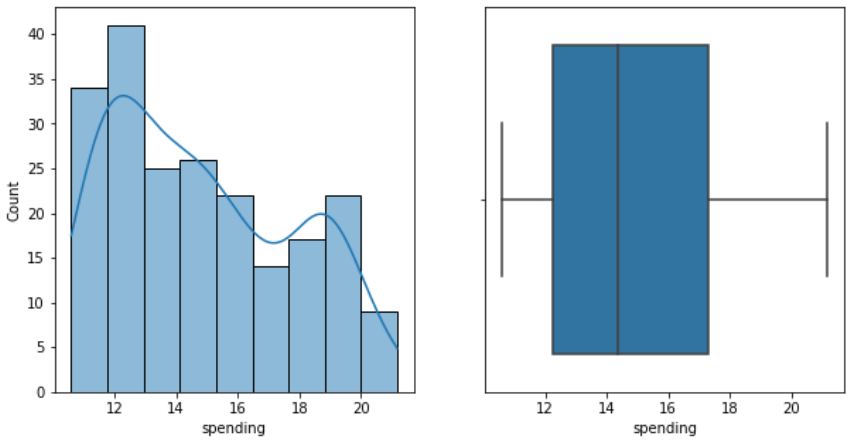


Fig.no 1: hist plot & box plot (column:spending)

Observation :

- From the above graphs, we can infer that distribution of 'spending' is right skewed.
- The histplot shows the distribution of data ranges from 10 to 22 (1000's)
- The boxplot of the 'spending' variable shows no outliers.

Column : Advance_payments (Amount paid by the customer in advance by cash (in 100s)

Skewness of advance_payments: 0.38

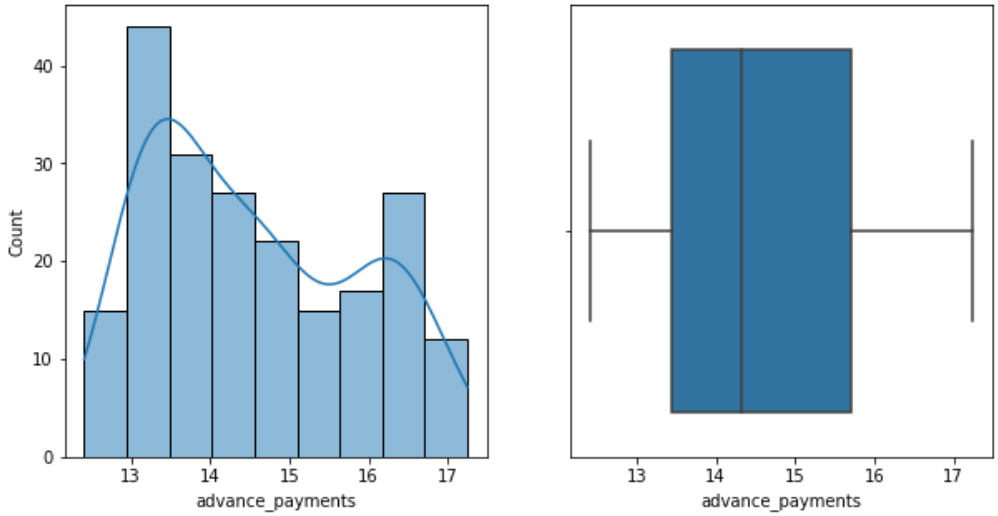


Fig.no 2: hist plot & box plot (column:Advance payments)

Observation :

- From the above graphs, we can infer that distribution of ‘ advance_payments ’ is slightly right skewed.
- The histplot shows the distribution of data ranges from 12 to 17 (100’s)
- The boxplot of the ‘ advance_payments ’ variable shows no outliers.

Column : Probability_of_full_payment (Probability of payment done in full by the customer to the bank)

Skewness of probability_of_full_payment: 0.38

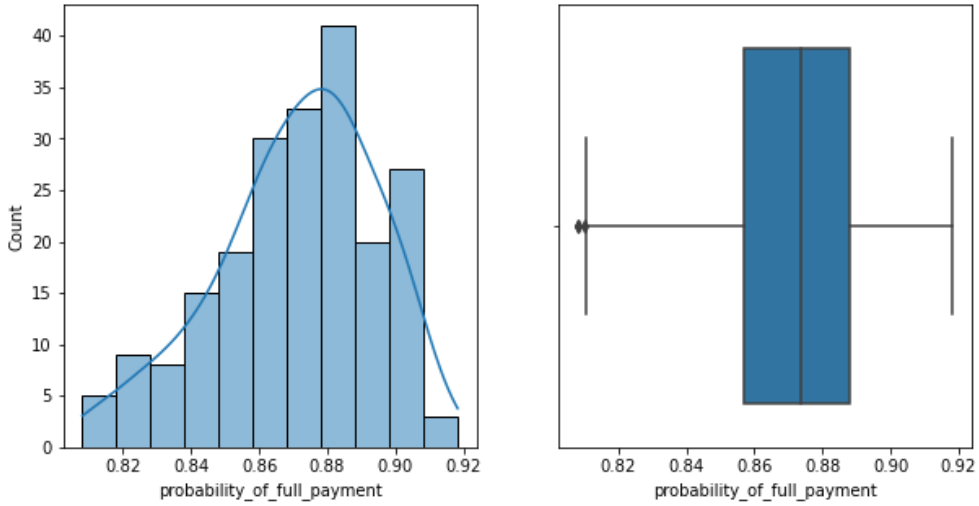


Fig.no 3: hist plot & box plot (column:probability of full payments)

Observation :

- From the above graphs, we can infer that distribution of ‘ probability of full payment ’ is left skewed.
- The histplot shows the distribution of data ranges from 0.82 to 0.92
- The boxplot of the ‘ probability of full payment ’variable shows few outliers.

Column : Current_balance (Balance amount left in the account to make purchases (in 1000s)

Skewness of current_balance: 0.38

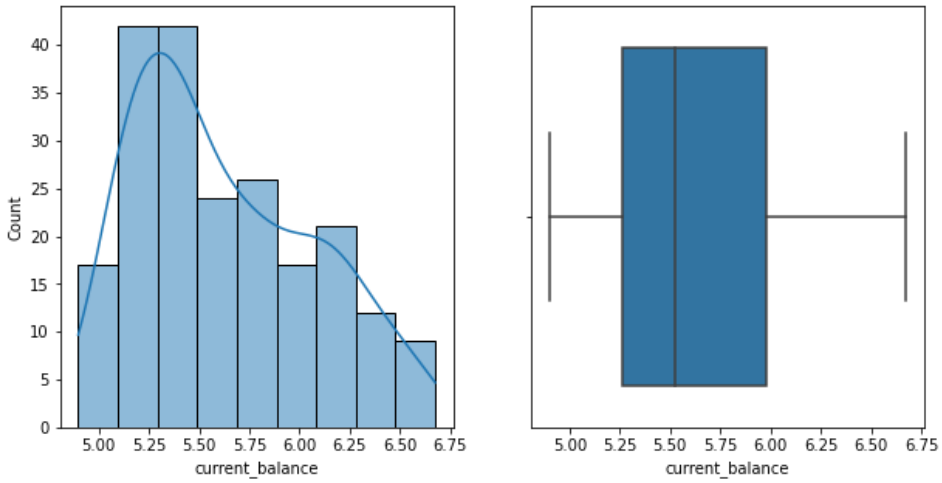


Fig.no 4: hist plot & box plot (column:current balance)

Observation :

- From the above graphs, we can infer that distribution of ‘ current_balance ’ is right skewed.
- The histplot shows the distribution of data ranges from 5.0 to 6.5(1000’s).
- The boxplot of the ‘current_balance’ variable shows no outliers.

Column: Credit_limit (Limit of the amount in credit card (10000s)

Skewness of credit_limit: 0.38

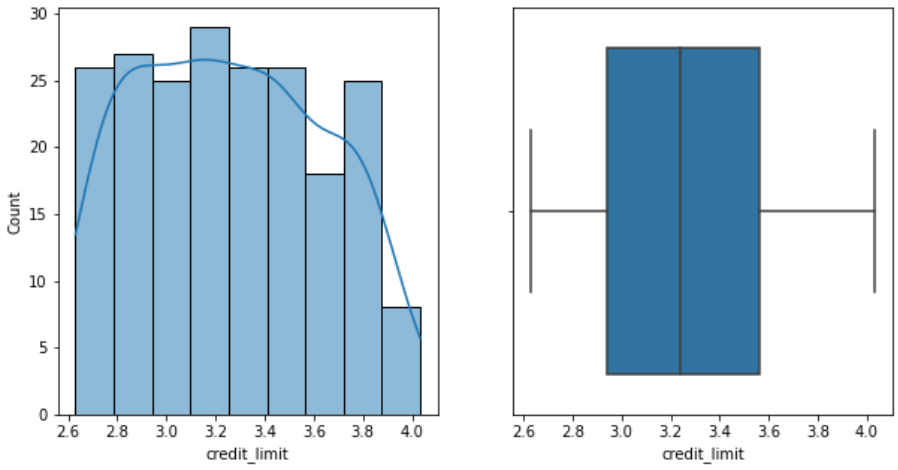


Fig.no 5: hist plot & box plot (column:credit limit)

Observation :

- From the above graphs, we can infer that distribution of ‘ credit_limit ’ is slightly right skewed and almost seems to be normal.
- The histplot shows the distribution of data ranges from 2.6 to 4.0(10,000’s).
- The boxplot of the ‘ credit_limit ’ variable shows no outliers.

Column: min_payment_amt (minimum paid by the customer while making payments for purchases made monthly (in 100s)

Skewness of min_payment_amt: 0.40

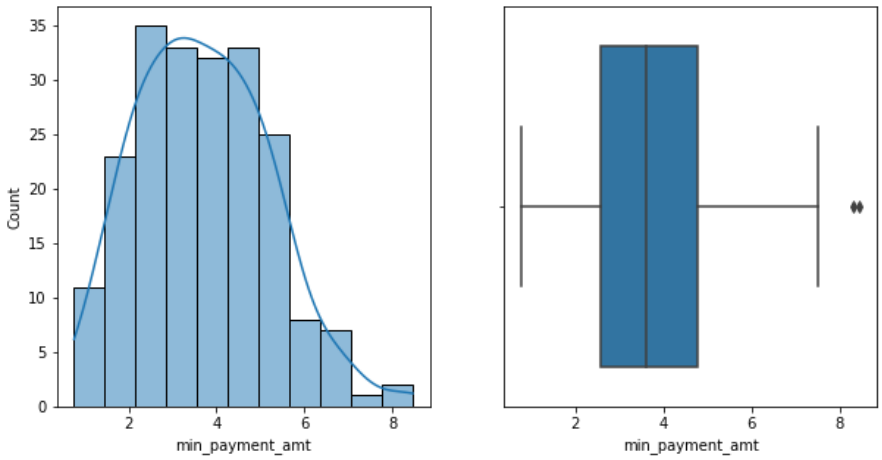


Fig.no 6: hist plot & box plot (column:min payment amt)

Observation :

- From the above graphs, we can infer that distribution of ‘ min_payment_amt ’ is almost seems to be normal.
- The histplot shows the distribution of data ranges from 2 to 8(100’s).
- The boxplot of the ‘ min_payment_amt ’ variable shows few outliers.

Column: Max_spent_in_single_shopping (minimum paid by the customer while making payments for purchases made monthly (in 100s)

Skewness of max_spent_in_single_shopping: 0.40

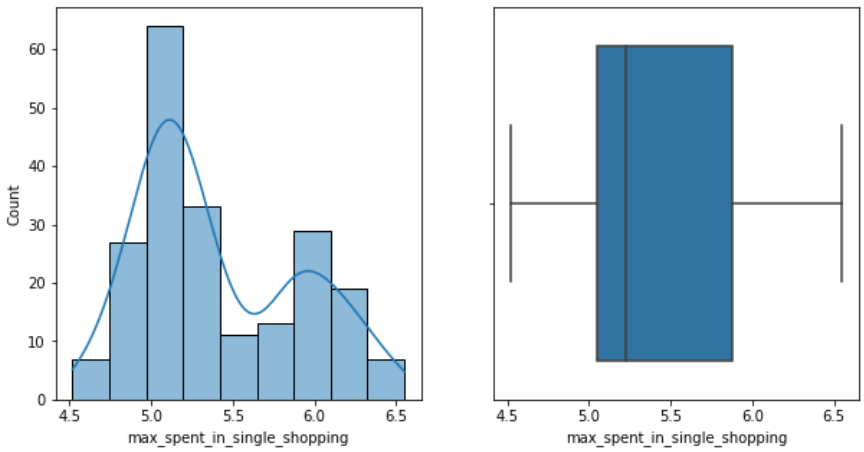


Fig.no 7: hist plot & box plot (column:max spent in single shopping

Observation :

- From the above graphs, we can infer that distribution of ' credit_limit ' is right skewed.
- The histplot shows the distribution of data ranges from 4.5 to 6.5(1000's).
- The boxplot of the ' credit_limit ' variable shows no outliers.

Bivariant Analysis:- Heatmap

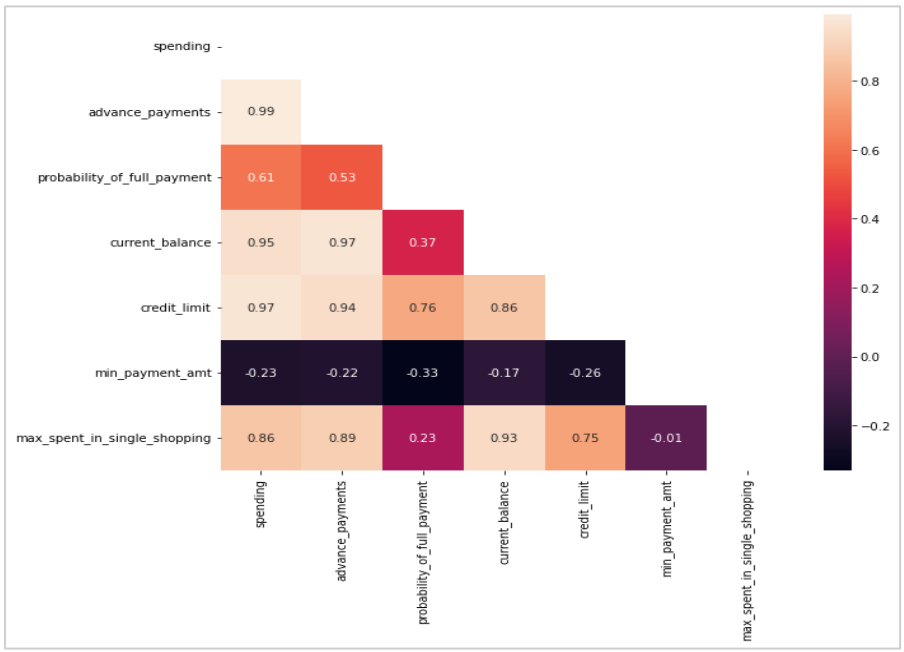


Fig.no 8: Heat Map – Bivariant Analysis

Observation :-

Correlation is a statistical measure that expresses the extent to which two variables are linearly related.

- In reference to above heatmap and correlation matrix table we can infer the **high positive correlation** among following variables :-

Spending and advance_payment	0.99
Advance_payment and current_balance	0.97
Spending and credit_limit	0.97
Spending and current_balance	0.94
Advance_payment and credit_limit	0.94
Current_balance and max_spent_in_single_shopping	0.93

Table 3 : Highest Correlation Variables

- In reference to above heatmap and correlation matrix table we can infer the **Moderate positive correlation** among following variables :-

Advance_payments and max_spent_in_single_shopping	0.89
Spending and max_spent_in_single_shopping	0.86
Current_balance and credit_limit	0.86
Probability_of_full_payments and credit_limit	0.76
Credit_limit and max_spent_in_single_shopping	0.75

Table 4 : Moderate Correlation Variables

- In reference to above heatmap and correlation matrix table we can infer the **poor and negative correlation** among following variables :-

Spending and Probability_of_full_payments	0.61
Advance_payments and Probability_of_full_payments	0.53
Probability_of_full_payments and current balance	0.37
Probability_of_full_payments and max_spent_in_single_shopping	0.23
Spending, advance_payments, probability_of_full_payments, current_balance, credit_balance and min_payment_amount has negative correlation	-017 to -0.33
Min_payment_amount and max_spent_in_single_shopping	-0.01

Table 5 : Poor/Negative Correlation Variables

Inferences:-

- From the above bivariant plot we can say that customers who spends high on purchase have a high current balance and credit limit. In an average, the Advance payments and maximum amount spent in single shopping are done by the customer who have high current balance in their bank accounts.
- Probability of full payments are high for the customers who has higher credit limit
- Minimum payment amount is not correlated to any of the variable, hence the chance of affecting a customer in terms of Spending, advance_payments, probability_of_full_payments, current_balance, credit_balance is impossible.

Multivariant Analysis :- Pair Plot

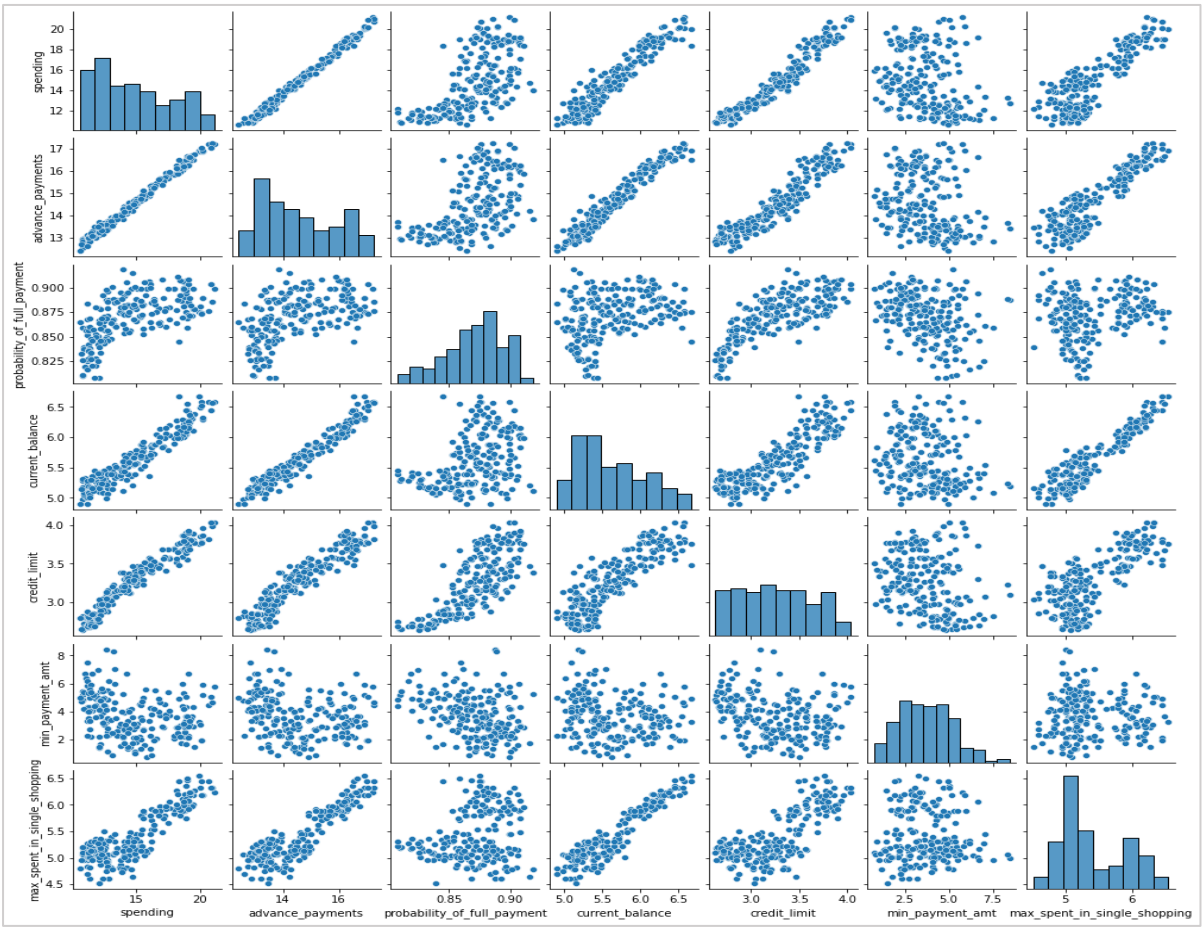


Fig.no 9: Pair plot – Multivariant Analysis

Observation:-

- **spending vs advance_payments**

From the pair plot we see that as the variable spending increases, an advance_payment is also increasing. This shows high positive relationship between variable spending and advance_payment.

- **spending vs credit_limit**

From the pair plot we see that as the variable spending increases, the credit_limit is also increasing. This shows a positive relationship between variable spending and credit_limit.

- **advance_payments vs current_balance**

From the pair plot we see that as the variable advance_payment increases, the current_ balance is also increasing. This shows a positive relationship between variable advance_payment and current_ balance.

- **spending vs current_balance**

From the pair plot we see that as the variable spending increases, the current_balance is also increasing. This shows a positive relationship between variable spending and current_balance.

- **advance_payments vs credit_limit**

From the pair plot we see that as the variable advance_payments increase, the credit_limit is also increasing. This shows a positive relationship between variable advance_payments and credit_limit.

- **min_payment_amt vs spending**

From the above pair plot we see that variable min_payment_amt decreased spending also decreased this shows a negative relationship.

- **min_payment_amt vs advance_payments**

From the above pair plot we see that variable min_payment_amt decreased advance_payments also decreased this show a negative relationship.

- **min_payment_amt vs probability_of_full_payment**

From the above pair plot we see that variable min_payment_amt decreased and probability_of_full payment also decreases this shows a negative relationship.

Outlier Treatment –

The present of outliers in the dataset may affect the output during Clustering. That is because each centroid is a mean, that is measure of central tendency whose value is affected by extreme values.

- As per univariant analysis (Boxplot) we deducted the presence of few outliers in variable Prob_of_full_payment and Min_Payment_Amt.
- Using IQR Capping method, Imputing the Outlier values by replacing the observations outside the lower limit with the value of 25th percentile; and the observations that lie above the upper limit, with the value of 75th percentile of the same dataset.

The below boxplot shows that, after IQR imputation no outliers got deducted in the variable Prob_of_full_payment and Min_Payment_Amt.

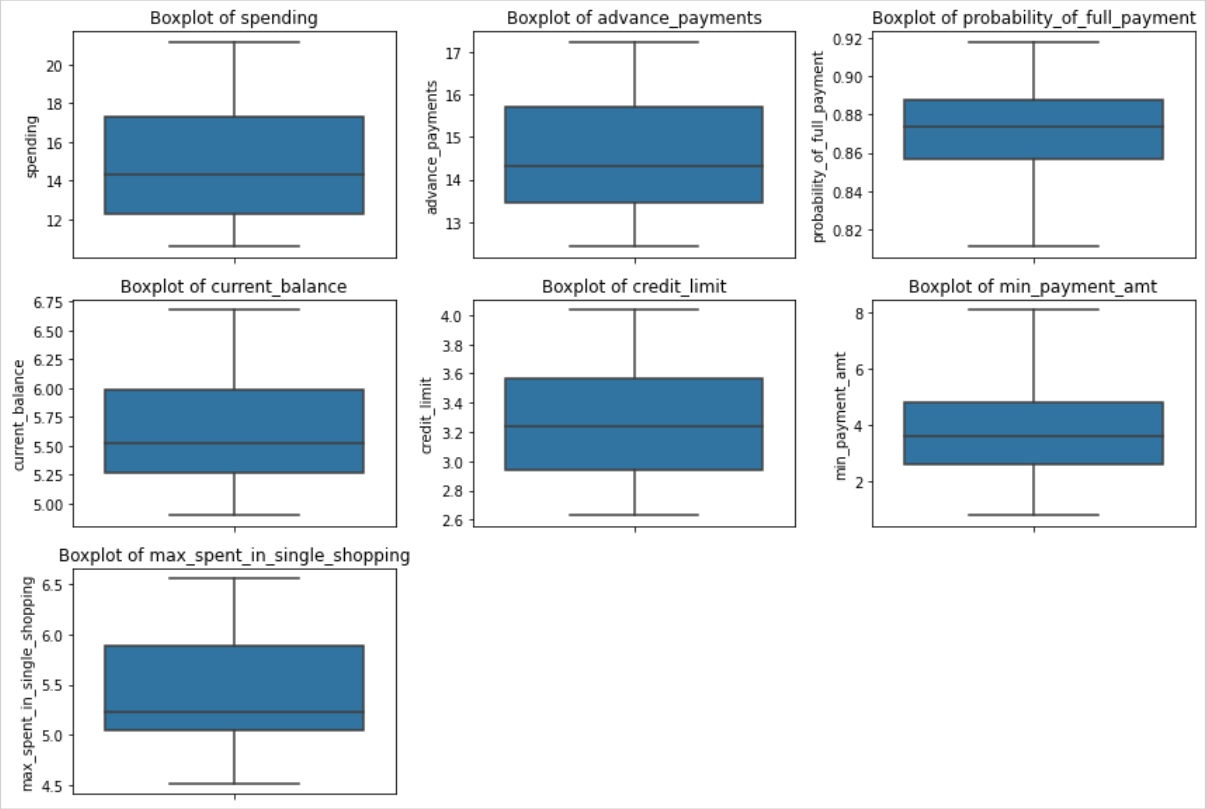


Fig.no 10: Box plot for outlier deduction

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

Yes, it is necessary to normalize data before performing clustering. The clustering model works on the distance-based calculations. Mostly Euclidean distance. So, variable with a high standard deviation will have a higher weight for the calculation of clustering than a variable with a low standard deviation. If you normalize your data, all variables have the same standard deviation, thus all variables have the same weight, and your algorithm calculates relevant clustering.

By visualizing the data description summary in fig.no: , we can infer that:-

- **Mean Value** - Mean value of spending is 14.84, mean value of advance payments is 14.55, but the mean value differs across probability of full payments, current balance, credit limit, min payment amount and max spent in single shopping, their mean value ranges from (0.87 - 5.6).
- **Standard deviation** - There is no drastic std difference notable across 7 feature and they range from (0.02 - 2.90).
- **IQR & Range** (min, 25%, 50%, 75%, max) - IQR & Range value of spending and advance payments ranges from (10.60 -21.18), but the IQR & Range value differs across probability of full payments, current balance, credit limit, min payment amount and max spent in single shopping, their IQR & Range value ranges from (0.76 - 8.45).

In order to normalize the data, we can either go for Z-Score scaling or Min-Max normalization technique , but in this case we go with Z-Score as per the nature of the given dataset indicates the Mean Value, Standard deviation, IQR & Range value differs across all the 7 features, mainly against

with the feature spending and advance payment, also all the variables are expressed in different magnitudes/units such as spending in 1000's, advance payments in 100's and credit limit in 10000's, whereas probability is expressed as fraction or decimal values. So, the other values expressed in higher units will outweigh probabilities and can give varied results.

So, Standardizing the features using Standard Scalar will assumes your data is normally distributed within each feature and will scale them such that the distribution is centred around 0, with a standard deviation of 1.

Z scaling technique can be done using the Z Score formula or Standard Scalar formula in SKLearn package.

$$z = (x - \mu) / S$$

where,

- z is scaled data.
- x is to be scaled data.
- u is the mean of the training samples
- s is the standard deviation of the training samples

Below is the output from Jupyter using Standard Scalar formula in SKLearn package:

	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

Table 6 :Scaled date (Standard Scaler)

Visualizing the Plots prior for before scaling and after scaling dataset:

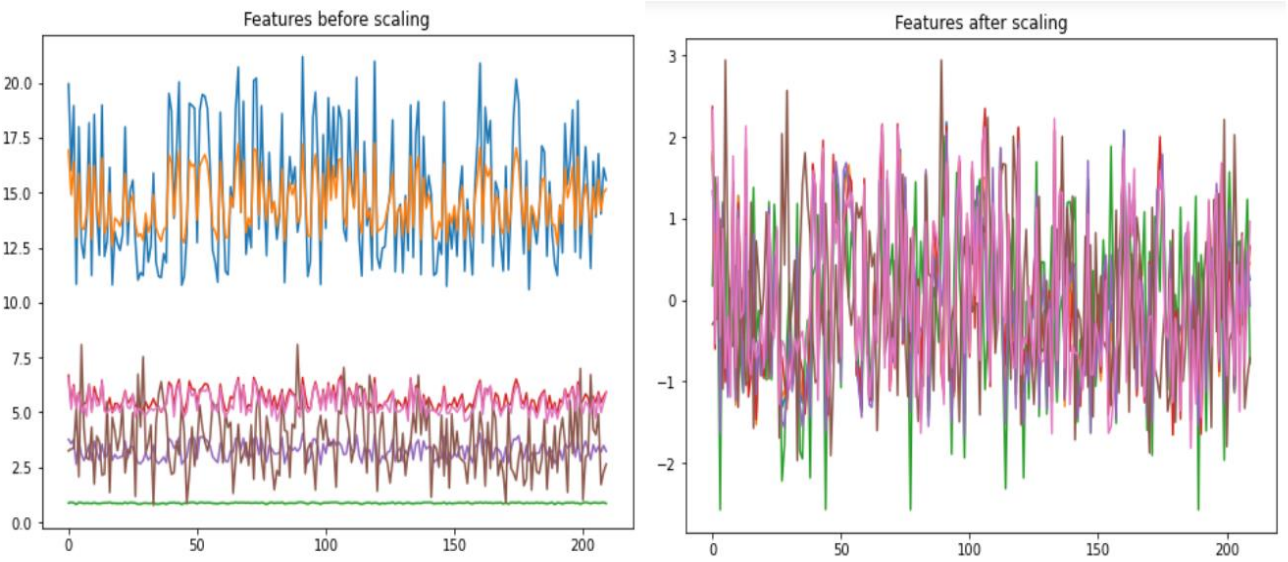


Fig.no 11: Plots prior before scaling and after scaling

Before Scaling:

Before scaling, the dataset is distributed as shown in the above figure. Some variables are plotted higher as they must be having higher absolute values while some are near the zero line as their absolute values are lower than the other variables. Even though the values for these variables are lower, they play significant role in the dataset, we cannot ignore them. Therefore, scaling is necessary, and we performed a Standard Scaler function to scale our dataset.

After Scaling:

From above figure we can now see that all variables are scaled, and the values are close to each other, and all variables are scaled to have a mean tending to 0 and standard deviation to 1. Therefore, scaling is important for our dataset before doing a clustering algorithm.

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

Hierarchical clustering is supervised learning method, and it is based on hierarchy representation of clusters where parent cluster node is connected to further to child cluster node. Anode represents collection of data points to one cluster. It is further divided into two types:

- Agglomerative Clustering
- Divisive Clustering

1. On the given dataset we are selecting an **Agglomerative Clustering** type to do Hierarchical clustering.

Agglomerative Clustering - The agglomerative also known as Agglomerative Nesting (AGNES). The methods start by considering each data point as a single cluster. In the next step the singleton clusters are merged into a bog cluster based on the similarity between them. The procedure is repeated until all the datapoints are merged into one big cluster. The procedure can be represented as hierarchy/tree of clusters.

2. Importing the packages of linkage and dendrogram in Jupiter using the library scipy.cluster.hierarchy.
3. We now perform Agglomerative clustering on the scaled data using Ward's linkage method.

Ward's Linkage - The linkage function specifying the distance between two clusters is computed as the increase in the "error sum of squares" (ESS) after fusing two clusters into a single cluster. Ward´s Method seeks to choose the successive clustering steps so as to minimize the increase in ESS at each step.

4. After performing the Ward's linkage to the scaled data, we now create a dendrogram.

Dendrogram – Dendrogram is a tree like diagram that summarize the process of clustering. On the X-axis are the records. Similar records are joined by lines whose vertical length reflects the distance between the records. The greater the distance in height, the more dissimilarity. By choosing a cut-off distance on the Y-axis, a set of clusters are created.

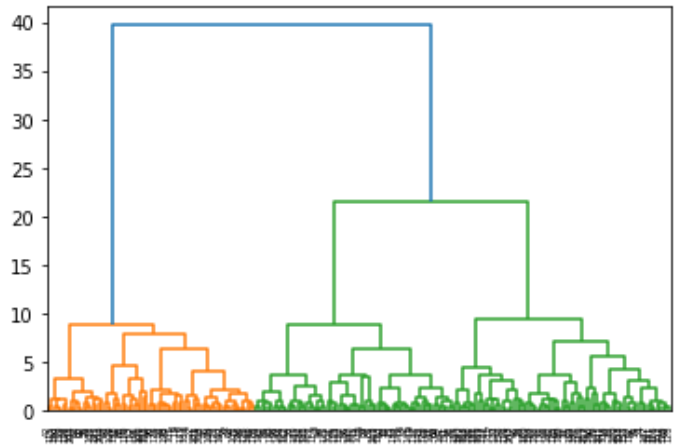


Fig.no 12: Hierarchical Dendrogram

The above dendrogram colour combination denotes the optimal number of cluster as 2. However, the dendrogram above is bit difficult to read and two clusters do not make much business impact as it is kind of implicit.

5. To understand the optimal number of cluster in more readable, cutting the dendrogram using truncate_mode and visualizing the last 10 linkage.

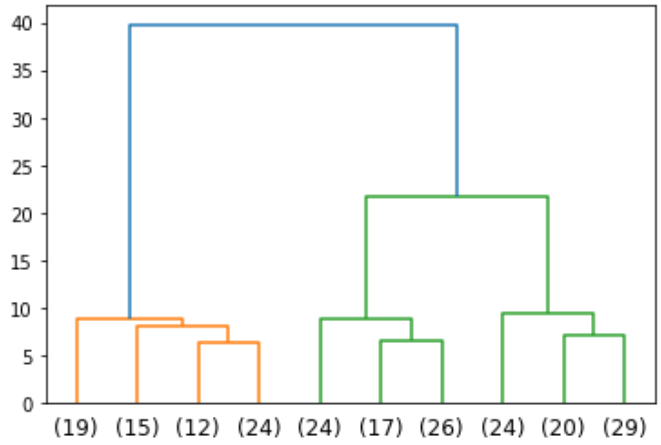


Fig.no 13: Truncated Dendrogram

From the above truncated dendrogram we say that there are more no of customers in cluster 2 compared to cluster 1.

The modified dendrogram using truncated also denotes the optimum number of cluster as 2, but 2 clusters really do not make much business impact as it is kind of implicit. For example, for the dataset, it is imperative that there will be some high spenders and some low spenders, then what is the need of doing clustering?. In this case, by considering the business point of view, we'll make 3 clusters to understand customer segmentation and their behaviour in a much better fashion.

6. Next Mapping the Clusters to Dataset using the fclusters function with criterion as maxclust or distance.
    - ❖ In maxclust, we decide the maximum number of clusters we wish to create for the dataset.

- ❖ In distance, we decide the distance on the y-axis in dendrogram and mention at which point we wish to cut for cluster formation.

7. We can pass the linkage in fcluster either by using the criterion as 'maxclust' along with minimum threshold value = 3 or by using the criterion as 'distance' along with Y-Axis distance =20. Both gives us the same result.

Below array output is cluster for each record of dataset which attired passing the linkage in fcluster, **criterion = maxclust & minimum threshold value = 3**.

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

Below array output is cluster for each record of dataset which attired passing the linkage in fcluster, **criterion =distance** & **Y-Axis distance =20**.

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

Both the criterion maxclust & distance gives the cluster value as 3, i.e.cluster 1 , cluster 2 & cluster 3.

8. Finally, appended the array to our original dataset and clusters got assigned to each record.

Below is the final dataset output from Jupyter:-

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

Table 7 : Clustered table (Agglomerative Clustering – Fcluster)

9. Cluster Frequency and Profiling:

Cluster Profiling is the most important part of the clustering exercise as it helps us in understanding what the clusters actually are in terms of customer behaviour and define them in a business-usable fashion.

- ❖ For cluster profiling, we are grouping the column ‘clusters’ in our dataset and calculating the mean of each variable.
- ❖ We can also calculate the frequency of each cluster and see how many records are grouped under each cluster.

clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371	70
2	11.872388	13.257015	0.848155	5.238940	2.848537	4.940302	5.122209	67
3	14.199041	14.233562	0.879190	5.478233	3.226452	2.612181	5.086178	73

Table 8 : Cluster Frequency and Profiling

As per above table, we get 3 clusters with almost equal number of records and all the 3 cluster gives a more importance to spending and advance_payment columns as per customer segmentation.

- ❖ Mean value of the spending in cluster 1 is 18.37 and mean value of the advance payment is 16.14.
- ❖ Mean value of the spending in cluster 2 is 11.87 and mean value of the advance payment is 13.25.
- ❖ Mean value of the spending in cluster 3 is 14.19 and mean value of the advance payment is 14.23.

Cluster 1 : High performing customers and holds 70 records

Cluster 2 : Low performing/new customers and holds 67 records.

Cluster 3 : Medium performing customers and holds 73 records.

Visualizing the mean customer segmentaion of each clusters with help of bar chart:-

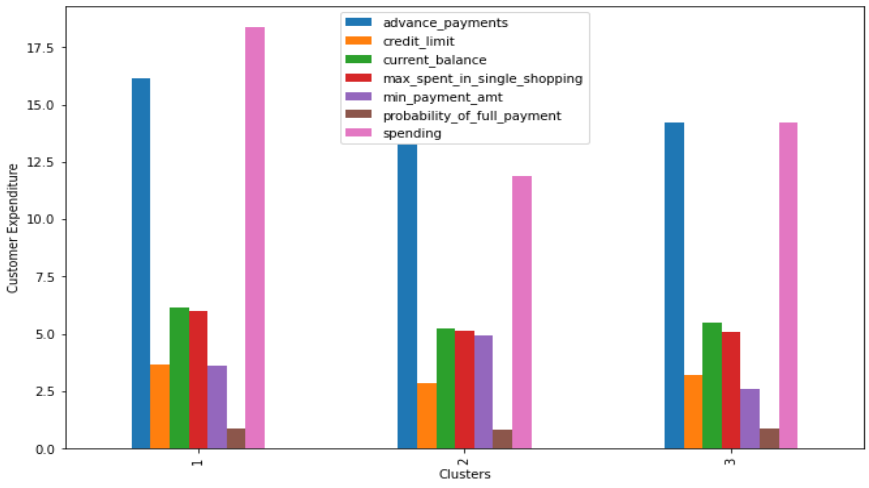


Fig.no 14: Bar Plot – Cluster mean value visualization

From the below bar plot, we can say that Cluster 1 has higher values for all variable combinations followed by cluster 3 followed by cluster 2.

- **In cluster 1** majority of the customer got segmented under 'Spending' category followed by 'advance payments' and the least customer got segmented under 'probability of full payment' followed by 'credit limit' & 'min_payment_amt' and moderate customers got segmented under 'current_balance' & 'max_spent_in_single_shopping'.
- **In cluster 2** majority of the customer got segmented under 'advance payments' category followed by 'spending' and very least customer got segmented under 'probability of full payment' followed by 'credit balance' and moderate customers got segmented under 'current_balance', 'max_spent_in_single_shopping' followed by 'min payment amount'.
- **In cluster 3** majority of the customer got segmented under both 'advance payments' and 'spending' category and very least customer got segmented under 'probability of full payment' followed by 'min payments amount' and moderate customers got segmented under 'current_balance', 'max_spent_in_single_shopping' followed by 'credit limit'.

Pair plot for customer segmentation:-

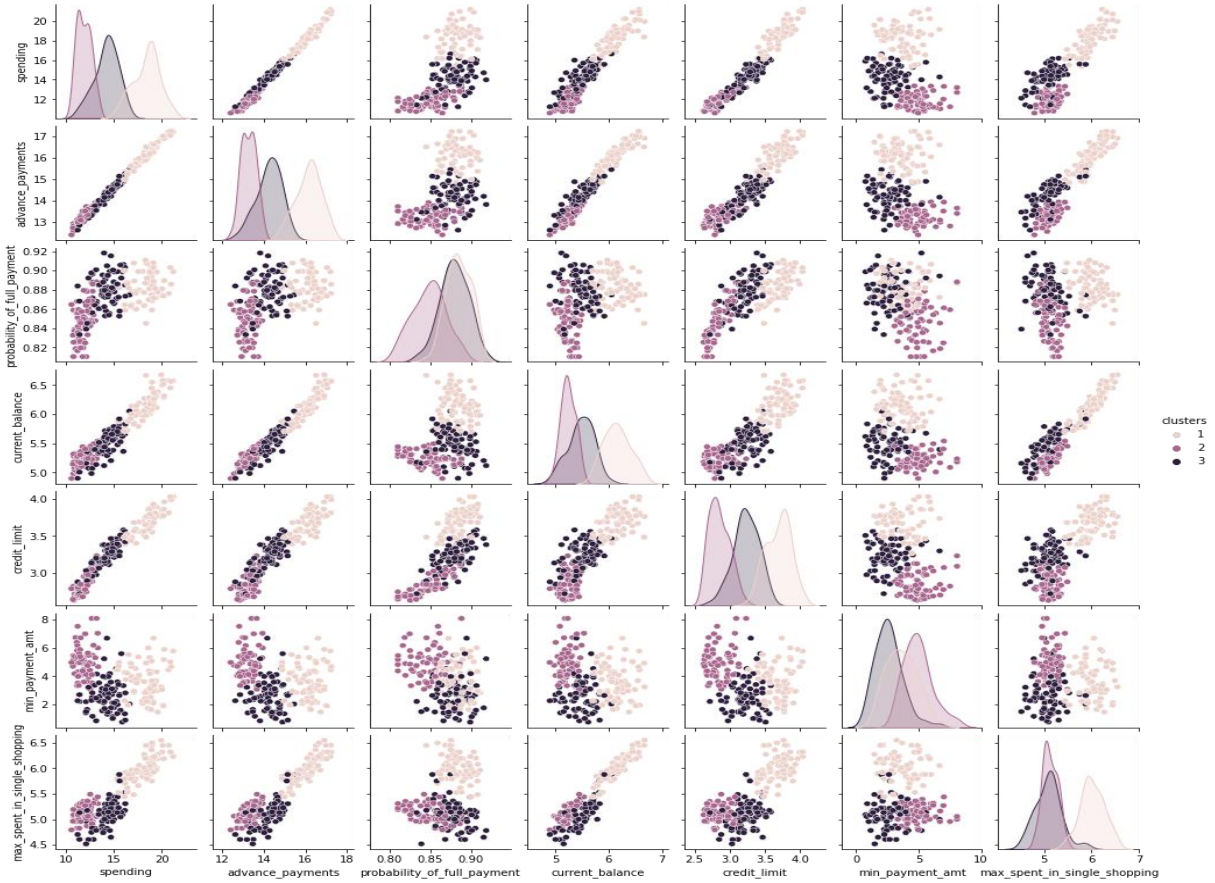


Fig.no 15: Pair Plot – Hierarchical Clustered Values

The customer segmentation can be visualized using pair plot and here the scatter plot clearly explains cluster 1 holds the high performing customers, cluster 2 holds the poor/new performing customers and cluster 3 holds a medium performing customers.

Across the clusters 1, 2 & 3 of customer segmentation, feature 'spending' and 'advance payments' plays an important role in terms of high expenditure.

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.

K-Means Clustering:-

K-means clustering is an unsupervised learning algorithm whose goal is to find groups or assign the data points to clusters on the basis of their similarity. Which means the points in same cluster are similar to each other and in different clusters are dissimilar with each other.

In this technique, we randomly take centroids in our dataset and calculate the within-cluster distance. The datapoints are clustered in such a way that distance between the assigned centroid is least as compared to distance with other centroids. The centroids are then changed accordingly to minimize the measure of dispersion within the clusters. This process is repeated till clusters with minimum dispersion within them are created.

The 'mean' in the K-Means refers in averaging of the data, that is finding the centroid by default K means clustering uses Euclidean method to find the distance.

1. Importing KMeans libraries from sklearn.cluster package in Jupyter for performing K-Means clustering.
2. We start with taking number of clusters ranges (1,11) and calculating within sum of squares (WSS) values for each using intertia. Below output is obtained from Jupyter.

Interia_ is the number of times of running the kmeans with different centroid's initialization and the result of the best one will be reported.

```
[1469.9999999999995,
 659.1474009548498,
 430.29848175122294,
 371.0356644664014,
 325.97412847298756,
 289.45524862464816,
 263.859944426353,
 239.9444663501791,
 220.5935394610811,
 205.76334196787008]
```

Fig.no 16: WSS K-mean cluster value

Above WSS result shows that, values start from cluster 1 till cluster 10. The optimum number of clusters can be obtained when the change in WSS score is not very significant, so from the WSS values, we observe till 3rd cluster the WSS values is significant, later no significant changes seen from cluster 4 to cluster 10. This is one the indication that, K=3 might be an optimum number for our dataset.

3. Plotting the calculated within sum of squares (WSS) ranges (1,11) using Elbow Method to find the optimal K-Values.

Elbow Method :

Elbow method is used to find the optimal no. of clusters or the value of k in the process of clustering. This method is based of plotting the value of cost function against different values of k. As the number of clusters (k) increases, the lesser number of points fall within clusters or around the centroids. Hence the average distortion decreases with the increase of number of clusters. The point

where the distortion declines most is said to be the elbow point and define the optimal number of clusters for data set .

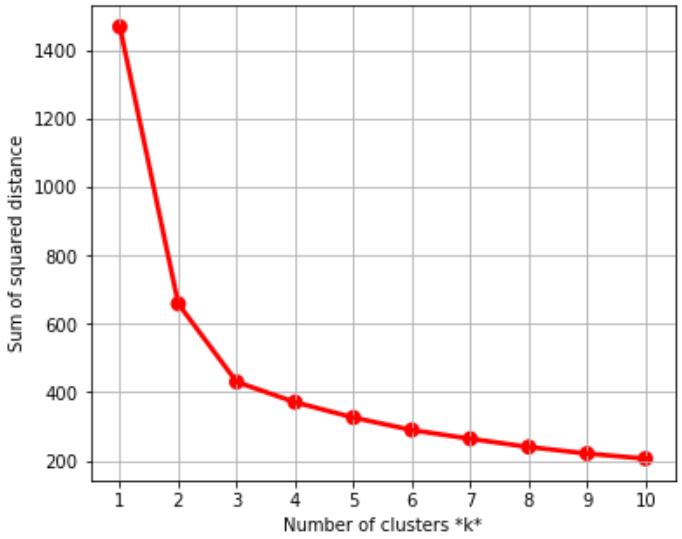


Fig.no 17: Elbow curve for K_means clustering

An above elbow plot says the distortion declines most at 3. Hence the optimal value of k will be 3 for performing the clustering. In other word the plot looks like an arm with an elbow at k = 3.

Silhouette Score:-

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

- Value +1 indicate that an observation is far from its neighbouring cluster and close to its own.
- Value -1 denotes that an observation is close to neighbouring cluster than its own cluster.
- Value 0 indicate the presence of observation on boundary of two clusters.

Silhouette coefficient is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Where a(i) is the distance of observation within its own cluster and b(i) is the distance of observation with its neighbouring cluster.

4. Here we will do cluster evaluation for K=2,K=3 & K=4 using Silhouette Score to cross validate, whether selected optimum value of K=3 is valid or not.
5. Before performing silhouette score we first calculate sil-width for each observation, and when we take the average of sil-widths that is known as silhouette score for a dataset.

In order to check silhouette width and silhouette score in Jupyter we will import silhouette_samples, silhouette_score libraries from sklearn.metrics package.

Below outputs are obtained in jupyter using silhouette_samples with min() command for silhouette widh :-

- Silhouette widh for K=2 is -0.005677379727717533
- Silhouette widh for K=3 is 0.0027685411286160638
- Silhouette widh for K=4 is -0.020681695700988604

Note : If Sil-width is a positive value, then we say that, the mapping of the observation is correct to its current centroid. We get negative value if distance of b(i) is less than distance of a(i).

Below outputs are obtained in jupyter using silhouette_score command:-

- Silhouette score for K=2 is -0.46560100442748986
- Silhouette score for K=3 is 0.4008059221522216
- Silhouette score for K=4 is 0.3373662527862716

6. Here we conclude that, from the above retrieved Silhouette scores for k=2 , k=3 & K=4. The optimum number of clusters or k value -should be 3 since silhouette score is close to +1 and its silhouette samples of minimum value is positive.

As we mentioned earlier if silhouette width value is positive mapping of the observation is correct to its current centroid and if the silhouette score is close to +1 the clusters are well separated by each other on an average.

7. Created k-means clusters using a command of n_clusters=3 in Jupyter.
8. Imported the clusters and silhouette width values to the original dataset.

Below is the Data frame where Clusters and Silhouette width are appended to the original dataset:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	sil_width	clusters
0	19.94	16.92	0.875200	6.675	3.763	3.252000	6.550	0.573278	1
1	15.99	14.89	0.906400	5.363	3.582	3.336000	5.144	0.365564	2
2	18.95	16.42	0.882900	6.248	3.755	3.368000	6.148	0.637092	1
3	10.83	12.96	0.810588	5.278	2.641	5.182000	5.185	0.515595	0
4	17.99	15.86	0.899200	5.890	3.694	2.068000	5.837	0.360972	1
5	12.70	13.41	0.887400	5.183	3.091	8.079625	5.000	0.221525	0
6	12.02	13.33	0.850300	5.350	2.810	4.271000	5.308	0.475295	0
7	13.74	14.05	0.874400	5.482	3.114	2.932000	4.825	0.360258	2
8	18.17	16.26	0.863700	6.271	3.512	2.853000	6.273	0.519383	1
9	11.23	12.88	0.851100	5.140	2.795	4.325000	5.003	0.534439	0

Table 9 : Clusters and Silhouette with appending on original data

9. Cluster Frequency and Profiling:

Cluster Profiling is the most important part of the clustering exercise as it helps us in understanding what the clusters actually are in terms of customer behaviour and define them in a business-usable fashion.

- For cluster profiling, we are grouping the column ‘clusters’ in our dataset and calculating the mean of each variable.
- We can also calculate the frequency of each cluster and see how many records are grouped under each cluster

Below is the output obtained for mean using transpose command :-

clusters	0	1	2
spending	11.856944	18.495373	14.437887
advance_payments	13.247778	16.203433	14.337746
probability_of_full_payment	0.848330	0.884210	0.881597
current_balance	5.231750	6.175687	5.514577
credit_limit	2.849542	3.697537	3.259225
min_payment_amt	4.733892	3.632373	2.707341
max_spent_in_single_shopping	5.101722	6.041701	5.120803
sil_width	0.399556	0.468077	0.338593
Freq	72.000000	67.000000	71.000000

Table 10 : Clustered table (K_maens)

As per above table, we get 3 clusters with almost equal number of records and all the 3 cluster gives a more importance to spending and advance_payment columns as per customer segmentation.

- Mean value of the spending in cluster 0 is 11.86 and mean value of the advance payment is 13.25.
- Mean value of the spending in cluster 1 is 18.50 and mean value of the advance payment is 16.20.
- Mean value of the spending in cluster 2 is 14.44 and mean value of the advance payment is 14.34.

Cluster 0 : Low performing/new customers and holds 72 records

Cluster 1 : High performing customers and holds 67 records.

Cluster 2 : Medium performing/new customers and holds 71 records.

Visualizing the mean customer segmentaion of each clusters with help of bar chart:-

From the below bar plot, we can say that Cluster 1 has higher values for all variable combinations followed by cluster 2 followed by cluster 0.

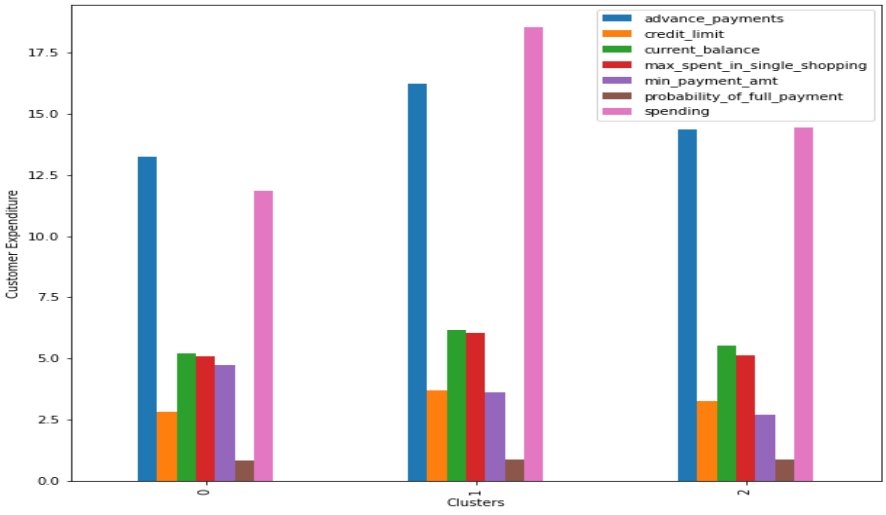


Fig.no 18: Bar Plot – K- mean cluster mean value visualization

Observation:-

- **In cluster 1** majority of the customer got segmented under 'Spending' category followed by 'advance payments' and the least customer got segmented under 'probability of full payment' followed by 'credit limit' & 'min_payment_amt' and moderate customers got segmented under 'current_balance' & 'max_spent_in_single_shopping'.
- **In cluster 2** majority of the customer got segmented under 'advance payments' category followed by 'spending' and very least customer got segmented under 'probability of full payment' followed by 'credit balance' and moderate customers got segmented under 'current_balance', 'max_spent_in_single_shopping' followed by 'min payment amount'.
- **In cluster 0** majority of the customer got segmented under both 'advance payments' and 'spending' category and very least customer got segmented under 'probability of full payment' followed by 'min payments amount' and moderate customers got segmented under 'current_balance', 'max_spent_in_single_shopping' followed by 'credit limit'.

Pair plot for customer segmentation:-

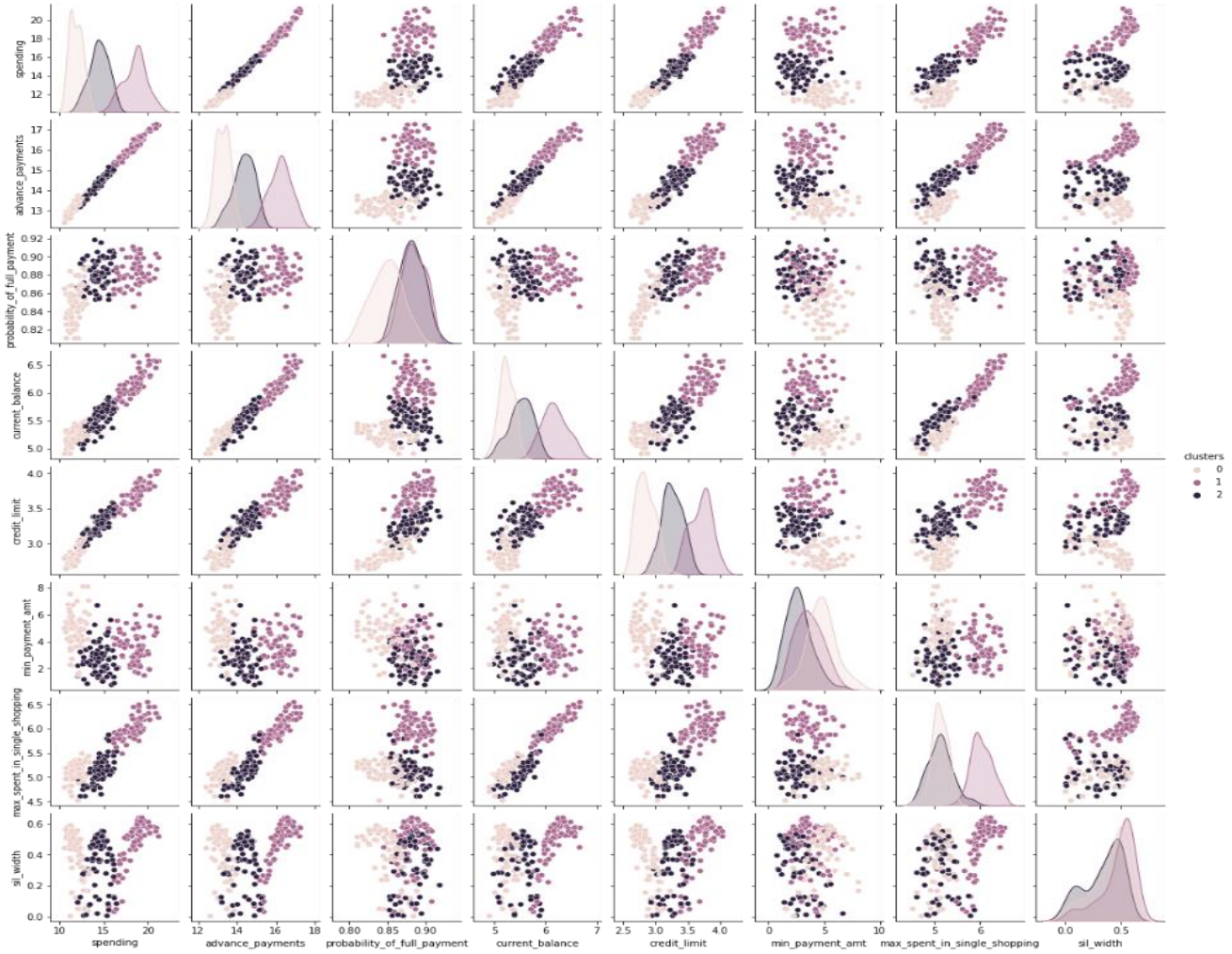


Fig.no 19: Pair plot – K- mean cluster

Observation:-

The customer segmentation can be visualized using pair plot and here the scatter plot clearly explains cluster 1 holds the high performing customers, cluster 2 holds the poor/new performing customers and cluster 3 holds a medium performing customers.

Across the clusters 0, 1 & 2 of customer segmentation, feature 'spending' and 'advance payments' plays an important role in terms of high expenditure.

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

Cluster Profiles - We do cluster profiling by calculating means of every variable for all the observations for every cluster in order to understand its behaviour across all variables.

Below cluster profiles is made using Hierarchical Clustering in terms of mean value:-

clusters	1	2	3
spending	18.371429	11.872388	14.199041
advance_payments	16.145429	13.257015	14.233562
probability_of_full_payment	0.884400	0.848155	0.879190
current_balance	6.158171	5.238940	5.478233
credit_limit	3.684629	2.848537	3.226452
min_payment_amt	3.639157	4.940302	2.612181
max_spent_in_single_shopping	6.017371	5.122209	5.086178
Freq	70.000000	67.000000	73.000000

Table 7 : Clustered table (Agglomerative Clustering – Fcluster)

From the above customer segmentation table in terms of mean value, we infer that,

Cluster 1: Tier 1 customers (High performing customers)

Cluster 2: Tier 3 customers (Low performing customers/new customers)

Cluster 3: Tier 2 customers (Medium performing customers)

Observation :-

Cluster 1:-

- An average amount spent by high performing customer per month (in 1000s) is 18.37 which is higher than low performing customer and medium performing customer.
- High performing customer make an advance payment by cash (in 100s) is 16.14 which shows very good indication. Mean probability of full payment is 0.88 which is pretty good.
- They have highest mean current balance (in 1000s) comparing to other two clusters which is 6.16.
- High performing customer maintain good credit balance (in 10000s) in an average mean of 3.68 and minimum amount paid by the customer while making payments for purchases made monthly (in 100s) is quite less which is 3.68, but they spent maximum amount in one purchase (in 1000s) is 6.01 which is high comparing to other clusters.

Cluster 2:-

- An average amount spent by low performing customer per month (in 1000s) is 11.87 which is very low than medium performing customer.
- Low performing customer make an advance payment by cash (in 100s) is 13.26 which is low comparing to high performing customer but doesn't make much difference with medium level performing customer.
- Mean probability of full payment is 0.84 which is pretty good.
- They have low mean current balance (in 1000s) of 5.24 comparing to high performing customer but doesn't make much difference with medium level performing customer which is 5.48.
- Low performing customer maintain poor credit balance (in 10000s) in an average mean of 2.86 which is very less than other two clusters.
- Low performing customer make a minimum amount for monthly purchases (in 100s) is quite high which is 4.94, also they are spending maximum amount in one purchase (in 1000s) is 5.12 which is not bad and close to other two clusters.

Cluster 3:-

- An average amount spent by moderate performing customer per month (in 1000s) is 14.20 which is higher than low performing customer, but less than high performing customer.
- Medium level performing customer make an advance payment by cash (in 100s) is 14.23 which is low comparing to high performing customer but doesn't make much difference with low level performing customer.
- Mean probability of full payment is 0.88 which is pretty good.
- They have low mean current balance (in 1000s) of 5.48 comparing to high performing customer but doesn't make much difference with medium level performing customer.
- Medium performing customer maintain good credit balance (in 10000s) in an average mean of 3.23.
- Medium performing customer make a minimum amount for monthly purchases (in 100s) is less which is 2.61, but those customer spent maximum amount in one purchase (in 1000s) is 5.09 and the mean value is close to high performing customers.

Below cluster profiles is made using K-Means Clustering in terms of mean value:-

clusters	0	1	2
spending	11.856944	18.495373	14.437887
advance_payments	13.247778	16.203433	14.337746
probability_of_full_payment	0.848330	0.884210	0.881597
current_balance	5.231750	6.175687	5.514577
credit_limit	2.849542	3.697537	3.259225
min_payment_amt	4.733892	3.632373	2.707341
max_spent_in_single_shopping	5.101722	6.041701	5.120803
sil_width	0.399556	0.468077	0.338593
Freq	72.000000	67.000000	71.000000

Table 10 : Clustered table (K_maens)

From the above customer segmentation table in terms of mean value, we infer that,

Cluster 0: Tier 1 customers (Low performing customers/ new customers)

Cluster 1: Tier 3 customers (High performing customers)

Cluster 2: Tier 2 customers (Medium performing customers)

Observation :-

Cluster 0:-

- An average amount spent by low performing customer per month (in 1000s) is 11.86 which is very low than medium & high performing customer.
- Though low performing customer doesn't make high spending, they make an advance payment by cash (in 100s) is 13.25 which is good but bit low comparing to high performing customer but doesn't make much difference with medium level performing customer.
- Mean probability of full payment is 0.84 which is pretty good.
- They have low mean current balance (in 1000s) of 5.24 comparing to high performing customer but doesn't make much difference with medium level performing customer which is 5.51.
- Low performing customer maintain poor credit balance (in 10000s) in an average mean of 2.85 which is very less than other two clusters.
- Low performing customer make a minimum amount for monthly purchases (in 100s) is quite high which is 4.73, also they are spending maximum amount in one purchase (in 1000s) is 5.10 which is not bad and close to other two clusters.

Cluster 1:-

- An average amount spent by high performing customer per month (in 1000s) is 18.50 which is higher than low performing customer and medium performing customer.
- High performing customer make an advance payment by cash (in 100s) is 16.20 which shows very good indication. Mean probability of full payment is 088. which is pretty good.
- They have highest mean current balance (in 1000s) comparing to other two clusters which is 6.18.
- High performing customer maintain good credit balance (in 10000s) in an average mean of 3.69 and minimum amount paid by the customer while making payments for purchases made monthly (in 100s) is quite less which is 3.63, but they spent maximum amount in one purchase (in 1000s) is 6.04 which is high comparing to other clusters.

Cluster 3:-

- An average amount spent by moderate performing customer per month (in 1000s) is 14.38 which is higher than low performing customer, but less than high performing customer.
- Medium level performing customer make an advance payment by cash (in 100s) is 14.33 which is low comparing to high performing customer but doesn't make much difference with low level performing customer.
- Mean probability of full payment is 0.88 which is pretty good.
- They have low mean current balance (in 1000s) of 5.51 comparing to high performing customer but doesn't make much difference with medium level performing customer.

- Medium performing customer maintain good credit balance (in 10000s) in an average mean of 3.25.
- Medium performing customer make a minimum amount for monthly purchases (in 100s) is less which is 2.71, but those customer spent maximum amount in one purchase (in 1000s) is 5.12 and the mean value is close to high performing customers.

From the above hierarchical and K-Means clusters observation we conclude that, both results provide similar information in terms of mean values for customer segmentation. This comparison can also be visualized using bar plots.

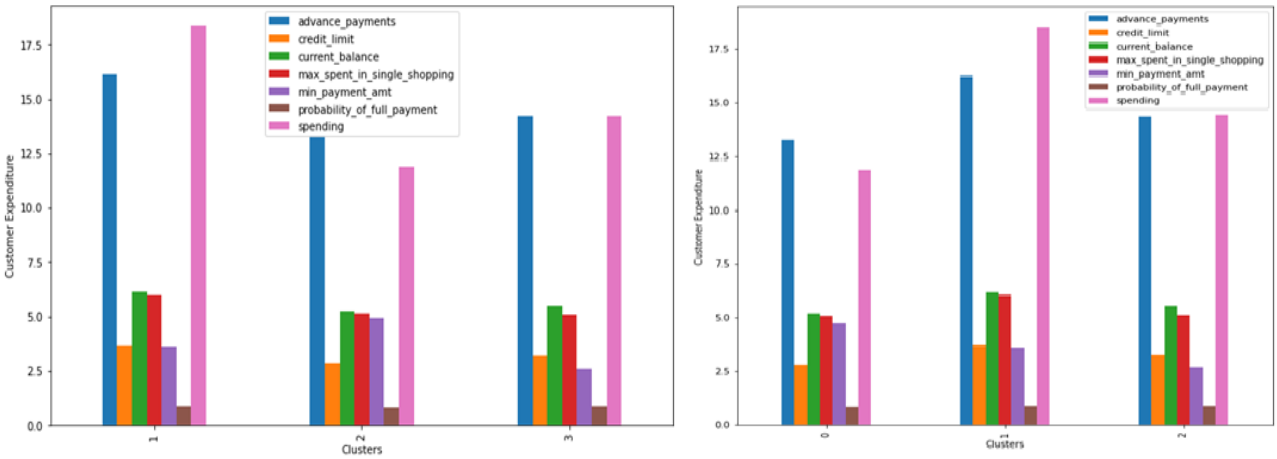


Fig.no 19: Bar plot-comparison of hierarchal and K_mean clusters

Recommendations for different promotional strategies across different clusters based on Kmeans Clustering:

Cluster 0 - Low spending

- This segment has lowest spending per month, they maintain lowest current balance and has low credit limit. This might be financially stressed class of low income in an average.
- Based on the observations this segment can be targeted by providing zero cost annual charges or can provide some benefits such as free coupons, discounts , cashbacks & low interest credit card. In turn would increase the overall credit consumption by the customers and leads in increasing overall business of the bank
- To boost their advance payment rate, early payment offers/Rewards can be implemented.
- Since minimum amount paid by the customer in making the payments for purchases is higher for low performing, we can try to lower their minimum payment amount by credit increasing their limit. This offer can be provided for the customers whose payment rate are fare enough and for the very low payment rate customer, offers can be given for early payments this will lead the low spending customer to increase their payment rate and this strategy might lead the low performing customer to become medium or high performing customers.
- A virtual point system could be used in which customers get certain amount of points for certain amount by using credit card and points gained by the customers can be redeemed during next purchase or certain coupons might prove beneficial to them.
- Increase their spending habits by making partnerships with grocery stores, utilities, and other businesses (electricity, phone, gas, others) and can provide cashbacks while using credit card.

Cluster 1 - High Performing Customers

- This segment has higher spending per month and has high current balance with high credit limit. This might be Prosperous or Upper class were majorly higher income holders.
- High performing customer maximum spent amount in one purchase is very high, so we can provide offers or discounts on their next huge transaction and giving some reward points. This might increase their level of spending and purchases.
- Tie up with luxury brands, can drive the high performing customers to spend maximum amount in one purchase.
- This segment can be targeted using various offers such as providing luxurious benefit card, Secure benefit cards, rewards, and loyalty points for every spent.

Cluster 2 - Medium Performing Customers

- This segment comprises of Customers making decent purchases, pay their bills on time, and maintaining comparatively good credit score, So we can increase their credit limit and can minimize the interest rate.
- For these customers, upselling can be done by giving some lucrative discounts so that, they try to use premium accounts, and this will lead to increase in transactions.
- To increase their spending rate, make a partner with premium ecommerce sites.
- Implement a customer loyalty program and encouraging customer to make a purchase in bulk to be rewarded with maximum discount.

Insurance Analytics – CART-RF-ANN

Table of Contents

Tabel List

List of Figure

Problem Statement 2:

Insurance Analysis:

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

Attribute Information:

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

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

The objectives of EDA can be summarized as follow:

- Maximize insight into the data/understand the data structure.
- EDA is an approach to analyse date using non-visual and visual techniques.
- EDA involves through analyse of data to understand the current business situation.
- EDA objective is to extract "'Gold' from the "'Data mine" based on domain understanding.

As a first step, importing all the necessary libraries, we think that will be requiring to perform the EDA.

Loading the data set – Loading the 'insurance_part2_data-1.csv' file using pandas. For this we will be using read excel file.

EDA Exploration: Following is the output from Jupyter.

Head of the dataset: After reading the CSV file, the head command option gives the bellow output.

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

Table 1 : Sample Data

From the above head table, we can say that,

- ❖ Dataset contains of 10 variables such as 'Age', 'Agency_Code', 'Type', 'Claimed', 'Commision', 'Channel', 'Duration', 'Sales', 'Product Name', 'Destination'.
- ❖ There are 9 independent variables and one target/dependent variable as 'Claimed'.

Shape of the dataset : Output from shape command is –

```
The dataset has 3000 rows and 10 columns
```

info() is used to check the Information about the data and the datatypes of each respective attributes: Output from Info command is –

```
<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
```

The dataset consists of 2 int data type, 2 float type and 6 object type.

Null value check : Output from isnull with sum command is –

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

Dataset doesn't contain null values.

Descriptive Analytics : Describe method will help us see how data is spread for the numerical values, also we can see the minimum value, mean values, different percentile values and maximum values.

Output from Describe with Transpose option is –

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

Table 2 : Descriptive Statistics

From the above descriptive table, we see that,

- There is a drastic change in terms of min and max values across 4 features, this is indication in presents of outliers.
- Feature 'Duration' has the min value as -1 which is a bad value and needs to be treated.
- Tour agencies commission charge has the max as 210.21% which exceeds 100% this categorized as bad value and need to be treated.

Checking value counts on all categorical column:-

```
Agency_Code
EPX    1365
C2B     924
CWT     472
JZI     239
Name: Agency_Code, dtype: int64
```

- There are 4 types of agency code present in the dataset namely EPX, C2B, CWT and JZT.
- Majority of the customers got segregated under agency code with the value count of 1365 and the minimum customer got segregated under agency code with the value count of 239.

```
Type
Travel Agency    1837
Airlines         1163
Name: Type, dtype: int64
```

- Most of the customers prefers Travel agency as their tour insurance firms that Airlines.

```
Claimed
No     2076
Yes     924
Name: Claimed, dtype: int64
```

- 924 customers claimed their insurance.
- 2076 customers didn't claim their insurance.

```
Channel
Online     2954
Offline      46
Name: Channel, dtype: int64
```

- They are 2954 customers who choose online tour insurance agencies.
- They are only 46 customers who choose offline tour insurance agencies.

```
Product Name
Customised Plan      1136
Cancellation Plan     678
Bronze Plan           650
Silver Plan           427
Gold Plan             109
Name: Product Name, dtype: int64
```

- ❖ Across 5 products customer who choose customised plan is high and their value count is 1136.
- ❖ Gold plan is the least purchased insurance plan by the customers with the value of 109.

```
Destination
ASIA          2465
Americas       320
EUROPE         215
Name: Destination, dtype: int64
```

- ❖ Asia is the most preferred tourist destination by customers with the value count of 2465.
- ❖ Europe is a least preferred tourist destination by customers with the value count of 215.

Treatment of bad values:-

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
1508	25	JZI	Airlines	No	6.30	Online	-1	18.00	Bronze Plan	ASIA
1746	48	C2B	Airlines	No	0.14	Online	0	0.51	Customised Plan	ASIA
2628	37	C2B	Airlines	No	49.60	Online	0	124.00	Bronze Plan	ASIA

Table 3 : Sample data after bad value treatment

- ❖ From the above table we infer that feature 'Duration' has the bad value lesser than 0 and their value count is 3, we are treating this bad value using median value.
- ❖ From the descriptive summary we identified that max value updated in feature 'Commission' as 210.12%. In Jupyter we can infer that feature 'Commission' has the 42 records, where commission percent mentioned more than 100%, here we are treating this bad value using median value.

	count	mean	std	min	25%	50%	75%	max
Age	3000.0	38.091000	10.463518	8.0	32.0	36.00	42.0	84.0
Commision	3000.0	12.537687	19.598981	0.0	0.0	4.63	15.6	99.9
Duration	3000.0	70.071668	134.034835	1.0	11.0	27.00	63.0	4580.0
Sales	3000.0	60.249913	70.733954	0.0	20.0	33.00	69.0	539.0

Table 4 : Descriptive Statistic after bad value treatment

- ❖ From above descriptive table we can see that both the missing values got treated using mean value now feature 'Duration' doesn't contain value -1 in min field and feature 'Commission' doesn't conation the value 210.10% in max filed.

Duplication check : Output from duplicated with sum command is –

```
The dataset has 130 duplication.
```

Though there is no customer ID or any unique identifier to determine whether it is true duplication or not, here we are dropping the duplicate value since it is less in number compared to size of the data and this will further avoid bias in analysis.

Duplication check after dropping: Output from duplicated with sum command is –

```
The dataset has 0 duplication.
```

Shape of the dataset after dropping duplicate values : Output from shape command is –

```
The dataset has 2861 rows and 10 columns
```

Univariant Analysis :- Histogram & Boxplot (Numeric Columns)

The objective of univariant analysis is to derive the data, define, analyze and summarize the pattern present in it. In a dataset, it explores each variable separately such as Numerical variable and Categorical variable. Some of the patterns that can be easily identified with univariant analysis are Central Tendency (mean, mode and median), Dispersion (range, variance), Quartiles (interquartile range), and Standard deviation. Univariant analysis can be described and visualize with the help of most used plots of Histogram/Distplot and Barplot.

Column :- Age

```
Skewness of Age: 1.10
```

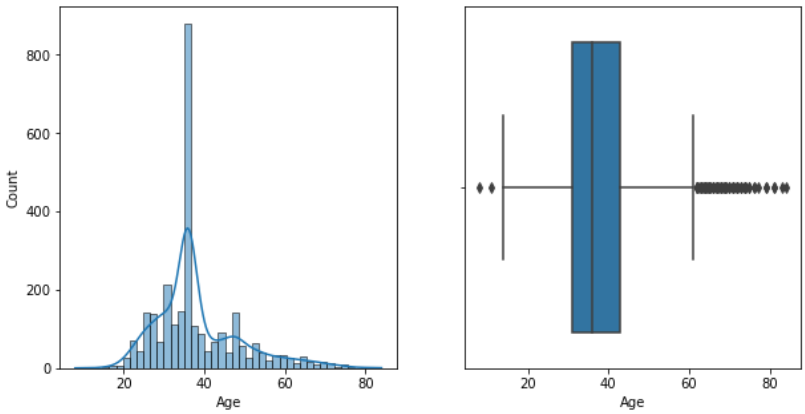


Fig.no 1: hist plot & box plot (column:Age)

- The hist plot shows the distribution of data from 20-80 years.
- Feature 'Age' got slightly right skewed.
- The box plot of 'Age' show lot of outliers.

Column :- Commision

```
Skewness of Commision: 1.91
```

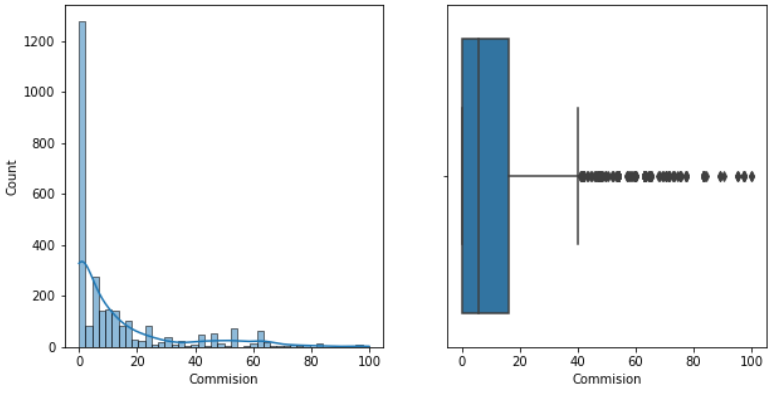


Fig.no 2: hist plot & box plot (column:Commision)

- The hist plot shows the Commision of data from 20-100 years.
- Skewness indicate that feature 'Commision' got right skewed.
- The box plot of 'Commision' shows lot of outliers.

Column :- Duration

```
Skewness of Duration: 13.78
```

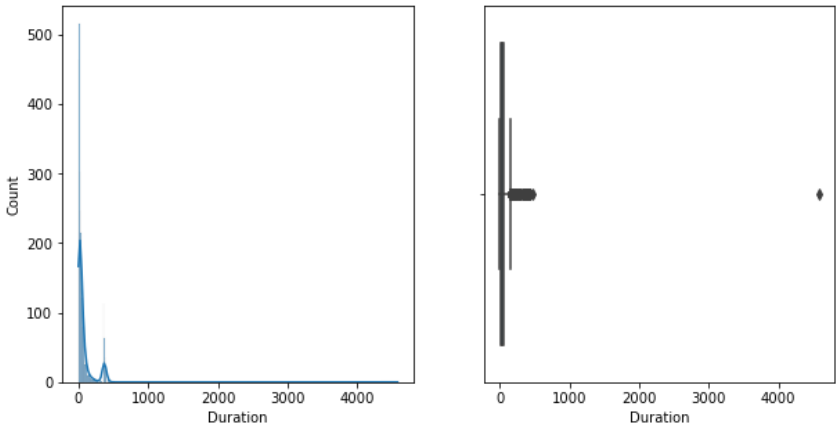


Fig.no 3: hist plot & box plot (column:Duration)

- The hist plot shows the Duration of data ranges from 1000-4000 years.
- Skewness indicate that feature 'Duration' got right skewed.
- The box plot of 'Duration' shows few outliers.

Column :- Sales

```
Skewness of Sales: 2.34
```

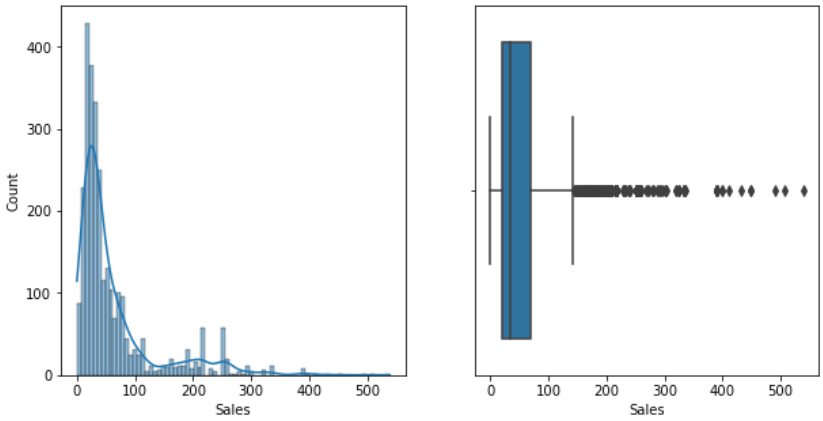


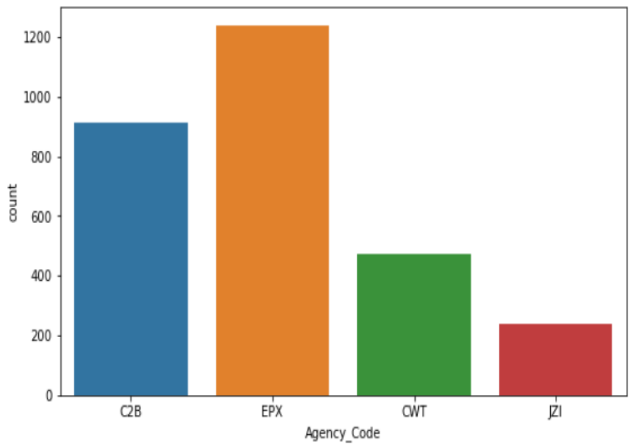
Fig.no 4: hist plot & box plot (column:Sales)

- The hist plot shows the Sales of data ranges from 100-500.
- Skewness indicate that feature 'Duration' got right skewed and has high positive kurtosis.
- The box plot of 'Sales' shows more outliers.

Univariant Analysis :- Count Plot (Categorical Columns)

A count plot is kind of histogram or a bar graph used to visualize the categorical variables.

Column:-Agency_Code



Column:-Channel

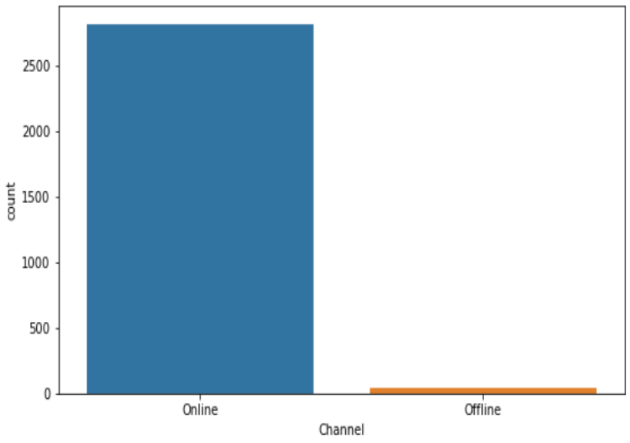
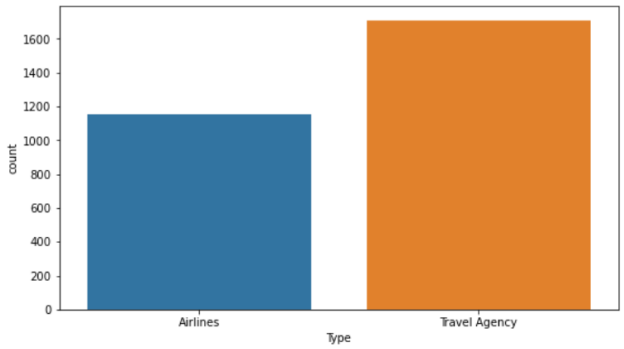


Fig.no 5:Univariant Analysis :- Count Plot (Categorical Column – Agency code & Channel)

Column:-Type



Column:-Claimed

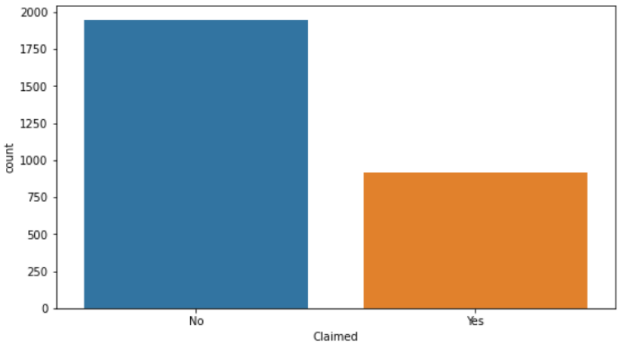
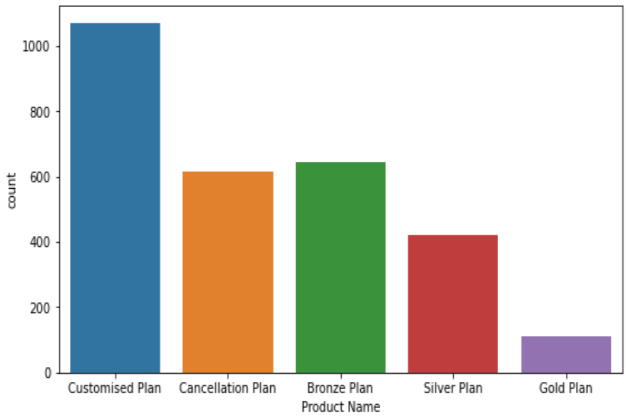


Fig.no 6:Univariant Analysis :- Count Plot (Categorical Column – Type & Claimed)

Column:-Product Name



Column:-Destination

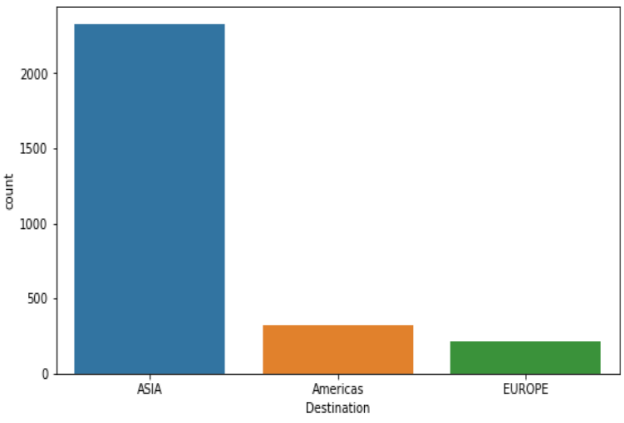


Fig.no 7:Univariant Analysis :- Count Plot (Categorical Column – Product name &destination)

Observation:

- Most of the people got segregated under EPX agency code with maximum frequency of 1238.
- Most of the customers prefers Travel agency as their tour insurance firms with the maximum frequency of 1709 than Airlines.
- Customers who claimed their insurance are 914 and 1947 customers didn't claim their insurance.
- The majority of customers have used online medium for travel and very less customer used an offline medium.
- Across 5 products customer who choose customised plan is high and their value count is 1071 and Gold plan is the least purchased insurance plan by the customers with the value of 109.
- Asia is the most preferred tourist destination by customers with the value count of 2327 and Europe is a least preferred tourist destination by customers with the value count of 215.

Bivariant Analysis:- Heatmap

A heatmap gives us the correlation between numerical variables. If the correlation value is tending to 1, the variables are highly positively correlated whereas if the correlation value is close to 0, the variables are not correlated. Also, if the value is negative, the correlation is negative. That means, higher the value of one variable, the lower is the value of another variable and vice-versa.

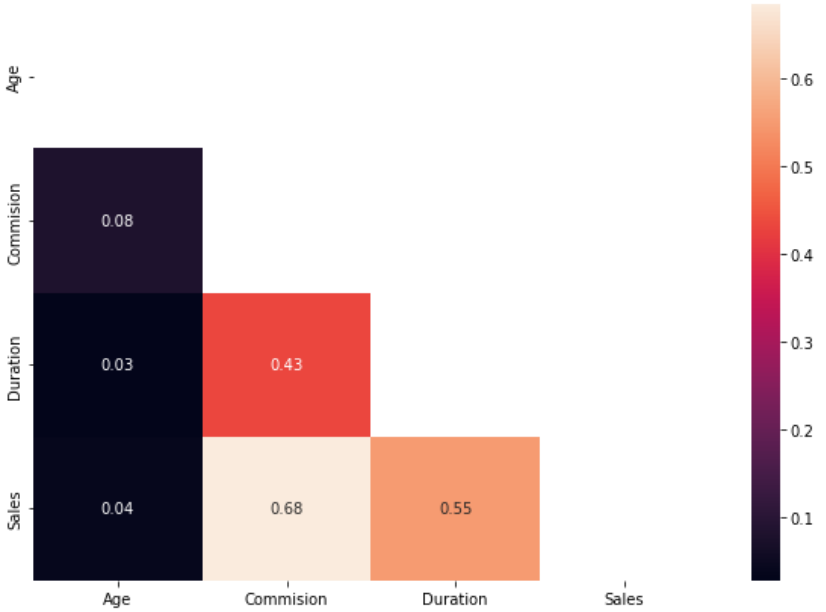


Fig.no 8:Bivariant Analysis :- Heatmap

Observation:-

- There is some correlation found between the feature 'Sales' & 'Commision' - (0.76).

- There is a weak correlation found between the variables 'Duration' & 'Commision' (0.46) and 'Duration' & 'sales' (0.55).
- Between variable Age & Sales, Duration & Commision correlation are very poor.
- There is no negative correlation found among 10 features.

Multivariant Analysis :- Pair Plot

A pair plot gives us correlation graphs between all numerical variables in the dataset. Thus, from the graphs we can identify the relationships between all numerical variables.

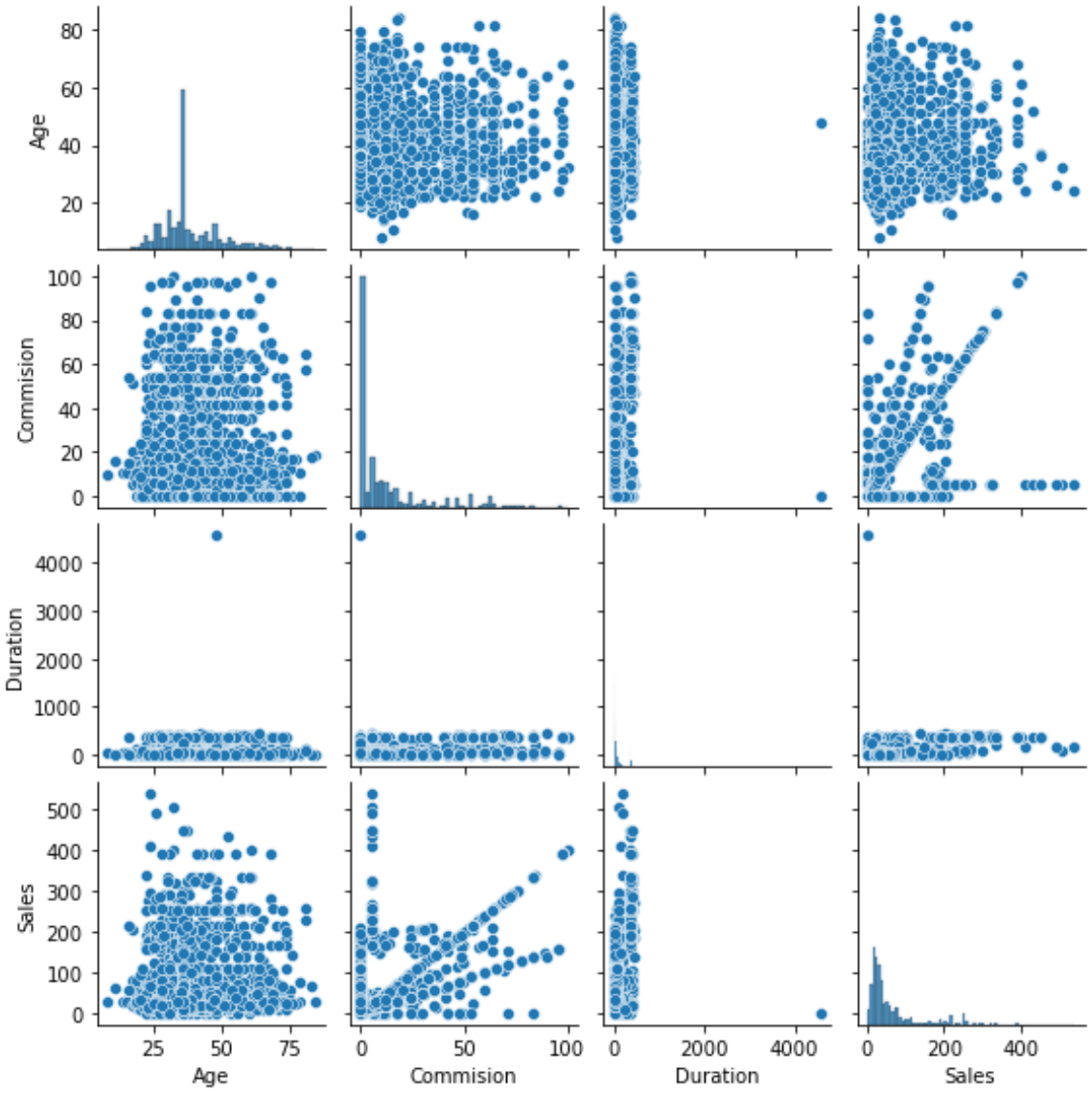


Fig.no 9:Multivariant Analysis :- Pair plot

Observation:-

- From above paid plot we visualize that when feature Sales increases the feature commission also increases that shows a positive relationship and that is an indication, as the number of sales per customer for insurance increases the commission received for tour insurance firm also increase.

- ❖ Apart from the feature Sales and commission there no much relationship found between other variables.

After performing EDA , data pre-processing & data preparation steps our dataset is now ready for supervised modelling algorithms like Decision Tree , Random Forest & ANN. Here we didn't perform outlier treatment though treating outliers sometimes results in the models having better performance, but the models lose out on the generalization and supervised algorithms are not case sensitive for outliers.

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

Before splitting the dataset into train and test, we need to perform the below step that is feature engineering:

1. **Feature Engineering**

Feature engineering is a technique used to encode categorical features into numerical values so that machine learning algorithm can understand. Most popular categorical converting technique is One hot encoding or Label encoding. Here, we use label encoding for categorical values to converted into simple numerical values without losing an information. During Label encoding all categorical features are labelled in a numeric values by alphabetical order.

Below is the output retrieved from Jupyter:-

```
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', 'Customise
d Plan', 'Gold Plan', 'Silver Plan']
[2 1 0 4 3]

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

```
[0 1 2]
```

2. **Checking the head of the dataset after label 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

Table 5 :Label encoded sample data

3. **Checking the Info of the dataset after label 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
```

Table 6 :Label encoded info data

By checking the head and info of the dataset after label encoding all object types got converted to number.

4. **Checking the proportion of observations using Target variable 'Claimed'.**

	Claimed
0	0.68
1	0.32

Table 7 :Target Variable

By observing an above output retrieved from Jupyter, we can say that there is no issue of class imbalance, and we have reasonable proportions in both the classes, the dataset is now ready for train and test.

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

- Here we store the impendent features in variable X (‘Age’, ‘Agency_Code’, ‘Type’, ‘Commision’, ‘Channel’, ‘Duration Sales’ , ’Product Name’ , ’Destination’) and dependent feature/Target feature in Y variable(‘Claimed’).
- Train data will hold an independent variables whereas test data will hold a dependent variable of the dataset.

6. **splitting data into training and test set.**

- Inorder to perform this step, from the package sklearn.model_selection we imported train_test_split.
- Now we split the data into 70 -30 ratio, where the train data hold 70% of the data and test data holds 30% of the data. The random state mentioned here is 1.

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

Below output is retrieved from Jupter using shape command

```
X_train (2002, 9)
X_test (859, 9)
train_labels (2002,)
test_labels (859,)
```

- Train dataset has 2002 records i.e., 70% of the total dataset.
- Test dataset contains 859 records i.e., 30% of the total dataset.

Now we have our train and test data ready. We will start building our classification model of Cart, random forest, and Artificial Neural Network.

Model 1 – CART /Decision Tree Model

Decision Tree model for a for a supervised learning algorithm which can be used for both classification and regression type of problems. Here we train the model on the training set and validate it on the testing set. The parent node gets split into child nodes and pruning is done to avoid overgrowing of sub-trees/branches.

Separation of purity from impurity is the subject behind the decision tree model. Here model by default uses ‘GINI’ index as a measure of impurity.

Decision tree is highly greedy in nature, in order to overcome this situation, we need to perform a cross validation technique, that is instead of performing single train and test model, we can subject the data to cross validation technique i.e., is K-forth validation.

1. **Building a Decision Tree Classifier**

Import the necessary library of DecisionTreeClassifier from sklearn.tree package and import GridSearchCV from the package sklearn.model_selection.

In this step we fit the train data and labels in the CART model, based on model performance, model will be tunned using Grid search.

2. **Hyperparameter Tuning**

'criterion': ['gini'],

'max_depth': [10,15,20,30],
'min_samples_leaf': [30,40,50,60],
'min_samples_split': [90,120,150,180]

Grid Search are used in 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, they are as follows:-

- Max_depth = (This can be of any range) - To prune our decision tree to the best height.
- min_samples_leaf' = 2-3 % of the dataset observation.
- min_samples_split = 3 times of min_sample_ leaf.

3. Best Estimator.

Below output is obtained from Jupyter that results the best estimator for building our decision tree and that is obtained using a grid search cv function.

'criterion': 'gini',
'max_depth': 10,
'min_samples_leaf': 30,
'min_samples_split': 150,

4. Generating Tree.

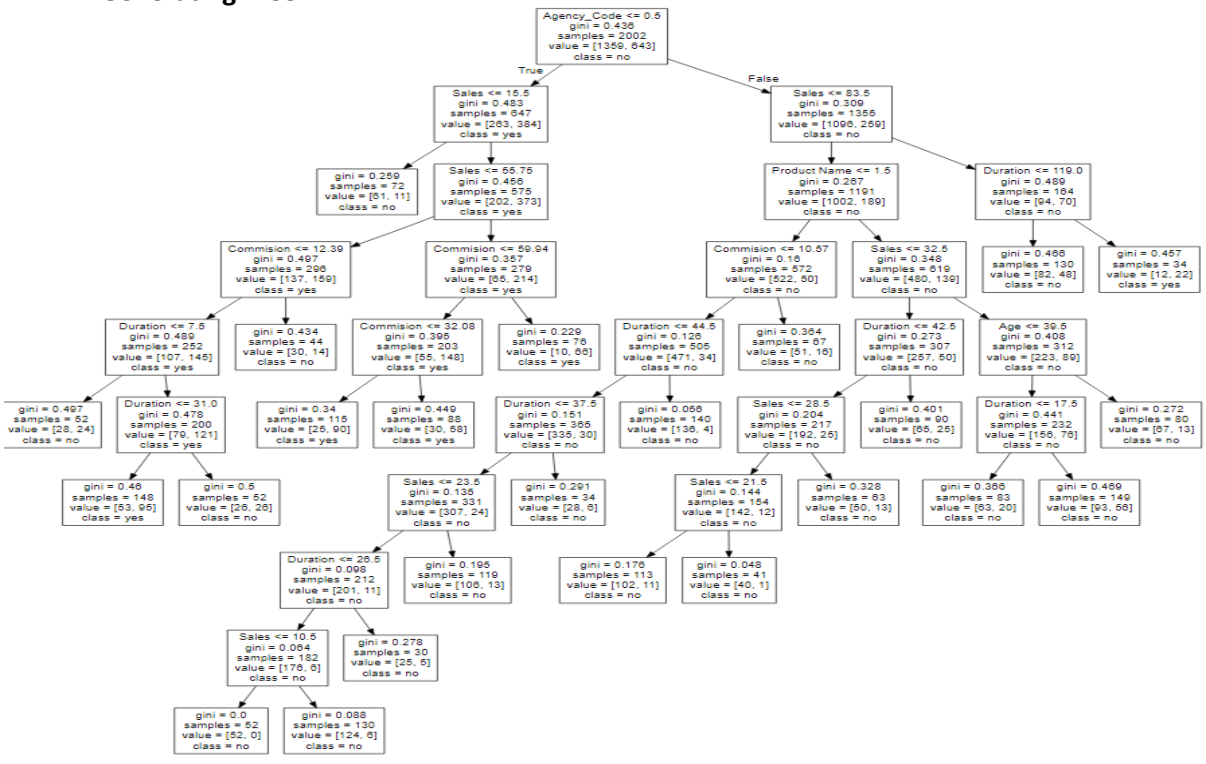


Fig.no 10:Random forest tree

5. Variable Importance

Variable importance is used to check the importance of variables and how the Decision tree is split into branches.

Below is the output retrieved from Jupyter using feature_importances_ command:-

	Imp
Agency_Code	0.619558
Sales	0.286769
Product Name	0.048864
Duration	0.023182
Commision	0.021627
Age	0.000000
Type	0.000000
Channel	0.000000
Destination	0.000000

Table 8: feature_importances

The Agency code variable has the highest importance with approximately 62% followed by Sales with 29%. Destination and Channel have the least importance with almost 0%.

Model 2 – Random Forest Model:-

Random Forest is based on the concept of ensemble learning, which is a process of combining multiple models to solve a complex problem and to improve the performance of the model. It is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the majority to improve the classification accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

1. **Building a random forest model.**

Import the necessary library of RandomForestClassifier from sklearn.ensemble package and import GridSearchCV from the package sklearn.model_selection.

In this step we fit the train data and labels in the CART model, based on model performance, model will be tunned using Grid search.

Grid Search are used in finding out the optimal values for the hyper parameters.

2. **Hyperparameter Tuning**

'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]

As per the industries standards we are taking various hyper parameters to build our random forest model, they are as follows:-

- To hyper tune the random forest trees to the best height, we take the values of max depth as 10,12,14,15 in our grid search to get optimum results.
- Max depth represents the depth of the tree in the forest. Max_features in the algorithm is the maximum number of features random forest model is allowed to try in an individual tree. There are many ways to take max_features. SQRT is one of them. Here we have 10 variables and taking 2 as max_features doesn't make much sense. For grid search we keep 6,7,8,9, to reduce correlation.
- Min samples leaf should be 1% - 3% of the total records. We take 50,100,150 and min samples split are approximately 3 times of Min samples life. So, we take for grid search 150,300,450.
- Cross validation (CV) sqrt of dataset value 3.
- N_estimators is the number of trees that we want to build before taking the maximum voting or averages of predictions. Higher number of trees give you better performance but makes your code slower. Here we check for the value 100,200 in our grid search to speed up the excecution.

3. Choosing best hyper prameters.

Below output is obtained from Jupyter that results the best estimator for building our decision tree and that is obtained using a grid search cv function.

```
max_depth=10,
max_features=7,
min_samples_leaf=50,
min_samples_split=150,
n_estimators=200,
random_state=1
```

4. Variable Importance

Variable importance is used to check the importance of variables and how the Decision tree is split into branches.

Below is the output retrieved from Jupyter using feature_importances_ command:-

	Imp
Agency_Code	0.479136
Sales	0.222068
Product Name	0.186787
Commision	0.043135
Duration	0.037456
Age	0.021512
Type	0.007092
Destination	0.002814
Channel	0.000000

Table 9: feature_importances

In both CART & Random Forest model feature 'Agency code' is having a high importance. In CART, 'Agency code' importance value is approximately 62%, here 'Agency code' importance value is approximately 48%

In CART model feature 'Sales' holds an importance of approximately 29%, but in Random forest model feature 'Sales' hold an importance of approximately 22% and feature destination and Channel have the least importance with almost 0%.

Model 3: Artificial Neural Network classifier:-

Artificial Neural Networks (ANN) are multi-layer fully connected neural nets. They consist of an input layer, multiple hidden layers, and an output layer. An artificial neuron receives a signal then processes it and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called *edges*. Neurons and edges typically have a *weight* that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold.

1. **Building a Neural Network forest model.**

Import the necessary library of MLPClassifier from sklearn.neural_network package and import GridSearchCV from the package sklearn.model_selection.

In this step we fit the train data and labels in the Artificial model, based on model performance, model will be tunned using Grid search.

2. **Hyperparameter Tuning**

Grid Search are used in finding out the optimal values for the hyper parameters.

hidden_layer_sizes': [50,100,200],
'max_iter': [4000,5000],
'solver': ['sgd','adam'],
'activation': ['logistic', 'relu'],
'tol': [0.01,0.001]
Cross validation (cv) =3

As per the industries standards we are taking various hyper parameters to build our decision tree, they are as follows:-

hidden layer = 20 to 200
max_iter'= 1000 to 5000
solver = sgd ,
Adam tol'=0.01 to 0.0000001

- A hidden layer is positioned between the input and output layer in neural networks, it takes the weighted sum of all the inputs, weighted by the *weights* of the *connections* from the inputs to the neuron. We add a *bias* term to this sum. This weighted sum is sometimes called the *activation*. This weighted sum is then passed through a (usually nonlinear) activation function to produce the output. The number of hidden neurons should be proportional to

the size of the input and output layers. In our grid search, we use 50, 100, and 200 to get the optimal parameter for our model.

- There are no set criteria for maximum iterations. The solver will make the model to run till it reaches convergence or the max iterations. In this case we have given 4000 and 5000 as inputs.
- The solver is the process that runs for optimization of the weights in the model. Here we imported sgd (Stochastic gradient descent) and adam as a solver.
- Activation is used to decide which activation would be best for the hidden layer calculation. Here we are using relu as an activation function.

  Relu is the rectified linear unit function [f(x) = max(0, x)]
- Tol is known as threshold values, they are used to stop the iteration for loss function to decrease the error. Lowering the tolerance, we are asking for higher model value. Higher the threshold we are asking for speed of computation. Here we use the tolerance as 0.01,0.001

3. **Choosing best hyper prameters.**

Below output is obtained from Jupyter that results the best estimator for building our decision tree and that is obtained using a grid search cv function.

```
'activation': 'relu',
'hidden_layer_sizes': 50,
'max_iter': 4000,
'solver': 'adam',
'tol': 0.001
```

4. **For the Artificial Neural Network model, there is no feature importance parameter.**

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.

Model performance:

This helps us to understand how good our model got trained. Model performance can be done only after the prediction of training and testing dataset . Here we validate if the model is underfitting or overfitting by checking certain parameters. Following methods are used to evaluate the model performance:

- Confusion Matrix

A confusion matrix is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm.

		Predicted	
Actual		Negative	Positive
	Negative	TN	FP
	Positive	FN	TP

-TN,TP - Correct Prediction (True Negative, True Positive)
- FP,FN - Incorrect prediction (False Positive, False Negative)

- Classification Report

1. Accuracy : Accuracy are used to identify how accurately/Cleanly , the model classifies the data point. Lesser the false predictions, more the accuracy.
Accuracy = (TP + TN) / (TP + TN + FP + FN).

2. Precision: Among the points identified as positive by the model, but how many points are actual positive.
If type(I) error is low pression will he high. Type(I) error and precision are inversely proportional to each other.
Precision = TP/(TP + FP).

3. Recall (Sensitivity): How many of the actual true data points are identified as True data points by the model. False Negative are those points should have been identified as True. Higher the sensitivity lowers the false negative(Type(II) error). Type(II) error and sensitivity are inversely proportional to each other
Recall = TP/(TP + FN)

4. Score: F1 Score computes an hormonic mean between Precision and Recall. It tells us both Type(I) and Type(II) error in a particular model is higher or lower on an average. If the F1 is good, that indicated model contains less false positives and less false negatives. F1 score is considered to be perfect when it tends to be 1 and model is a total failure when it tends to be 0.
F1 score = 2 x [(Precision x Recall) / (Precision + Recall)

- ROC Curve: ROC curve is a graphic representation of classifier performance. This curve plots two parameters: True Positive Rate. False Positive Rate. Higher the cure stronger the model, flatter the ROC curve weakest the model.

- AUC Score: AUC score gives the value of area under the ROC curve . The higher the AUC score, the better the performance of the model at distinguishing between the positive and negative.

Model 1 – CART /Decision Tree Model

1. **Prediction of training and testing dataset using predict command in train and test data.**
2. **Getting the Predicted Classes and Probs.**

Below is the output retrieved in Jupyter using proba command on train data:-

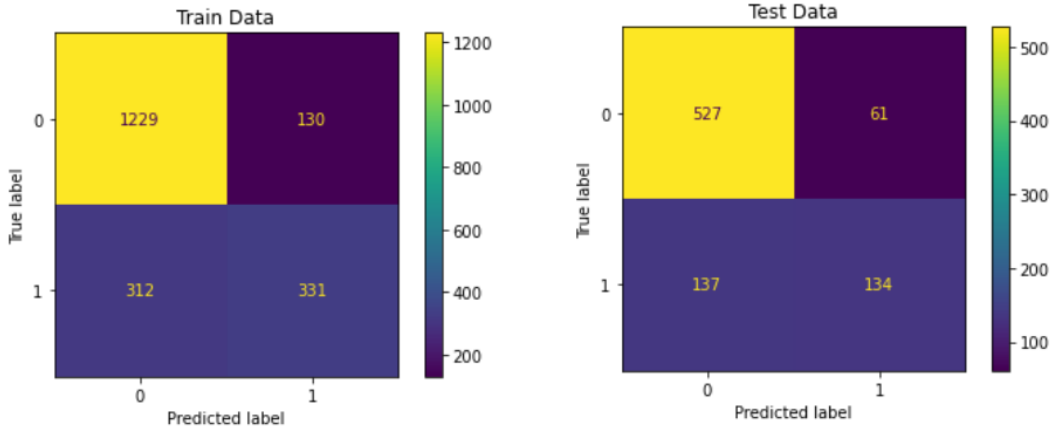
	0	1
0	0.711102	0.288898
1	0.748407	0.251593
2	0.772506	0.227494
3	0.454970	0.545030
4	0.884725	0.115275

Below is the output retrieved in Jupyter using proba command on test data:-

	0	1
0	0.564135	0.435865
1	0.924133	0.075867
2	0.290781	0.709219
3	0.757905	0.242095
4	0.694827	0.305173

3. Confusion Matrix

Below is the output retrieved from Jupyter using the confusion matrix command on train data and test data:-



4. Classification Report

Train Report

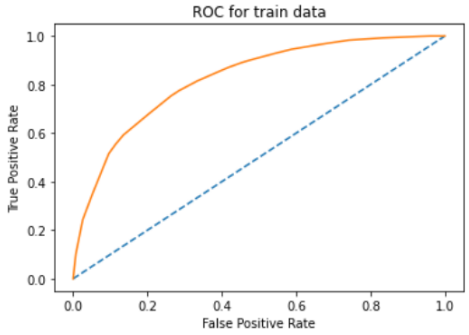
	precision	recall	f1-score	support
0	0.80	0.90	0.85	1359
1	0.72	0.51	0.60	643
accuracy			0.78	2002
macro avg	0.76	0.71	0.72	2002
weighted avg	0.77	0.78	0.77	2002

Test Report

	precision	recall	f1-score	support
0	0.79	0.90	0.84	588
1	0.69	0.49	0.58	271
accuracy			0.77	859
macro avg	0.74	0.70	0.71	859
weighted avg	0.76	0.77	0.76	859

5. ROC Curve and AUC score

AUC: 0.825



AUC: 0.781

6. Train and Test report comparison table

Card Model			
Sl.No	Index	Train Data	Test Data
1	TN	1229	527
2	TP	331	134
3	FN	312	137
4	FP	130	61
5	Accuracy	0.78	0.77
6	Precision	0.72	0.69
7	Recall	0.51	0.49
8	F1 Score	0.60	0.58
9	AUC Score	0.82	0.78

Observation:

- Cases customer actually claimed their tour insurance, there are 137 instances where model predicted the customer not claimed.
- Cases where the is customer actually not claimed their tour insurance, but model predicted them to be Malignant are 61.
- In both train and test data an accuracy, precision and recall not much difference found and they are nearly identical to each other. This shows that, model is neither overfitting nor underfitting.
- Due to false negative score/Type I error our recall went down to 0.49%.
- This might be because of uniformity of cell size, which plays an important part in deciding whether the customer claimed their tour insurance or not (Highest feature importance).

Model 2 – Random Forest Model

1. **Prediction of training and testing dataset using predict command in train and test data.**
2. **Getting the Predicted Classes and Probs.**

Below is the output retrieved in Jupyter using proba command on train data:-

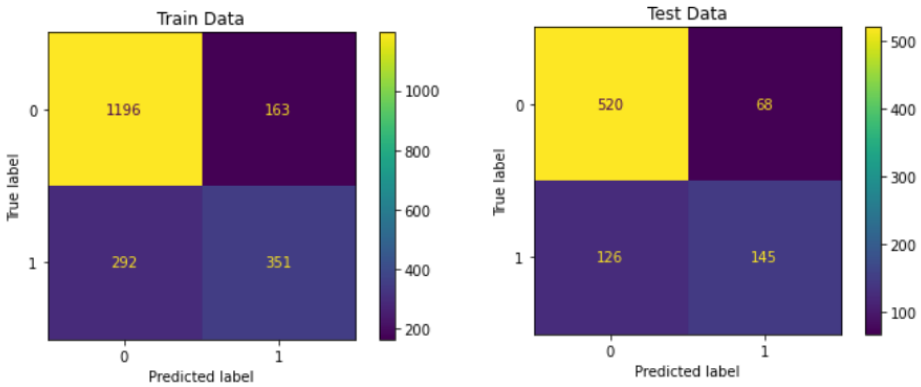
	0	1
0	0.711102	0.288898
1	0.748407	0.251593
2	0.772506	0.227494
3	0.454970	0.545030
4	0.884725	0.115275

Below is the output retrieved in Jupyter using proba command on test data:-

	0	1
0	0.564135	0.435865
1	0.924133	0.075867
2	0.290781	0.709219
3	0.757905	0.242095
4	0.694827	0.305173

7. **Confusion Matrix**

Below is the output retrieved from Jupyter using the confusion matrix command on train data and test data:-



8. **Classification Report**

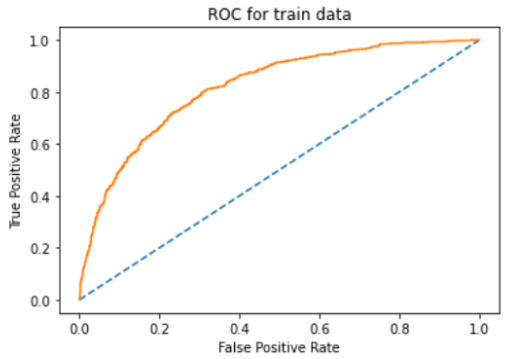
Train Report Test Report

	precision	recall	f1-score	support
0	0.80	0.88	0.84	1359
1	0.68	0.55	0.61	643
accuracy			0.77	2002
macro avg	0.74	0.71	0.72	2002
weighted avg	0.76	0.77	0.77	2002

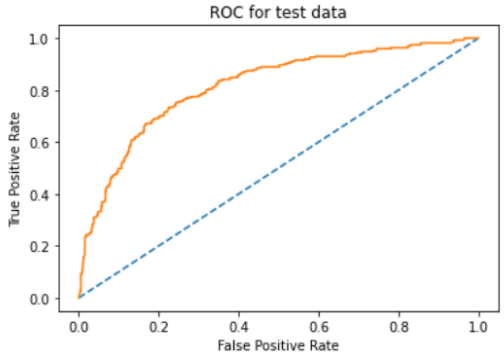
	precision	recall	f1-score	support
0	0.80	0.88	0.84	588
1	0.68	0.54	0.60	271
accuracy			0.77	859
macro avg	0.74	0.71	0.72	859
weighted avg	0.77	0.77	0.77	859

9. ROC Curve and AUC score

AUC: 0.820



AUC: 0.815



10. Train and Test report comparison table

Random Forest Model			
Sl.No	Index	Train Data	Test Data
1	TN	1196	520
2	TP	351	145
3	FN	292	126
4	FP	163	68
5	Accuracy	0.77	0.77
6	Precision	0.68	0.68
7	Recall	0.55	0.54
8	F1 Score	0.61	0.60
9	AUC Score	0.82	0.82

Observation:

- Cases customer actually claimed their tour insurance, there are 126 instances where model predicted the customer not claimed.
- Cases where the is customer actually not claimed their tour insurance, but model predicted them to be Malignant are 68.
- In both train and test data an accuracy, precision and recall not much difference found and they are nearly identical to each other. This shows that, model is neither overfitting nor underfitting.
- Due to false negative score/Type I error our recall went down to 0.54%.

- ❖ This might be because of uniformity of cell size, which plays an important part in deciding whether the customer claimed their tour insurance or not (Highest feature importance).

Model 2 – Artificial Neural Network

1. **Prediction of training and testing dataset using predict command in train and test data.**
2. **Getting the Predicted Classes and Probs.**

Below is the output retrieved in Jupyter using proba command on train data:-

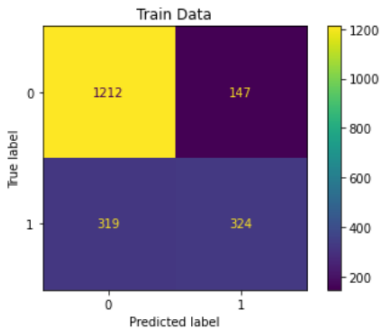
	0	1
0	0.762039	0.237961
1	0.766210	0.233790
2	0.700522	0.299478
3	0.248614	0.751386
4	0.745536	0.254464

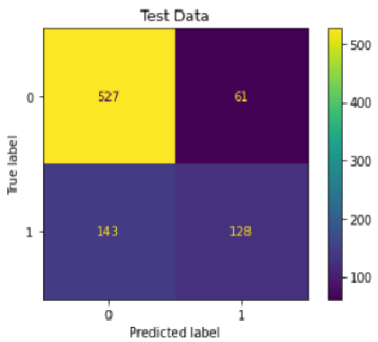
Below is the output retrieved in Jupyter using proba command on test data:-

	0	1
0	0.509838	0.490162
1	0.982913	0.017087
2	0.480214	0.519786
3	0.735416	0.264584
4	0.773997	0.226003

3. **Confusion Matrix**

Below is the output retrieved from Jupyter using the confusion matrix command on train data and test data:-





4. Classification Report

Train Report

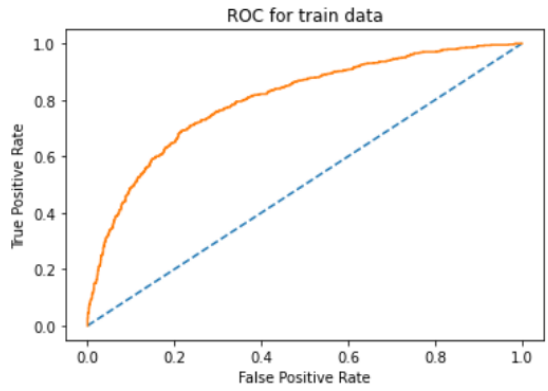
	precision	recall	f1-score	support
0	0.79	0.89	0.84	1359
1	0.69	0.50	0.58	643
accuracy			0.77	2002
macro avg	0.74	0.70	0.71	2002
weighted avg	0.76	0.77	0.76	2002

Test Report

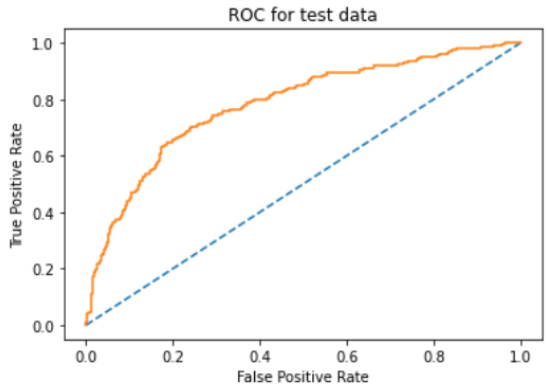
	precision	recall	f1-score	support
0	0.79	0.90	0.84	588
1	0.68	0.47	0.56	271
accuracy			0.76	859
macro avg	0.73	0.68	0.70	859
weighted avg	0.75	0.76	0.75	859

5. ROC Curve and AUC score

AUC: 0.799



AUC: 0.783



6. Train and Test report comparison table

Artificial Neural Network			
Sl.No	Index	Train Data	Test Data
1	TN	1212	527
2	TP	324	128
3	FN	319	143
4	FP	147	61
5	Accuracy	0.77	0.76
6	Precision	0.69	0.68
7	Recall	0.50	0.47
8	F1 Score	0.58	0.56
9	AUC Score	0.80	0.78

Observation:

- Cases customer actually claimed their tour insurance, there are 143 instances where model predicted the customer not claimed.

- Cases where the is customer actually not claimed their tour insurance, but model predicted them to be Malignant are 61.
- In both train and test data an accuracy, precision and recall not much difference found and they are nearly identical to each other. This shows that, model is neither overfitting nor underfitting.
- This might be because of uniformity of cell size, which plays an important part in deciding whether the customer claimed their tour insurance or not (Highest feature importance).

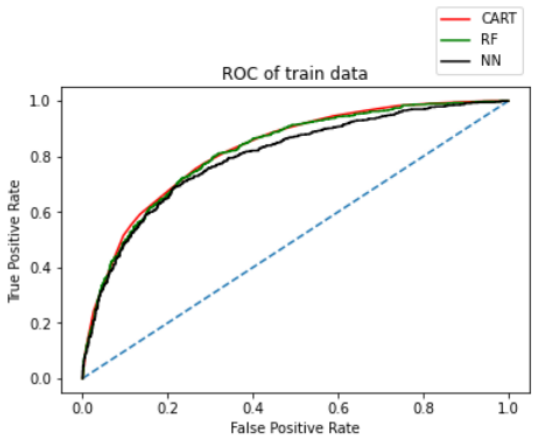
2.4 Final Model: Compare all the models and write an inference which model is best/optimized.

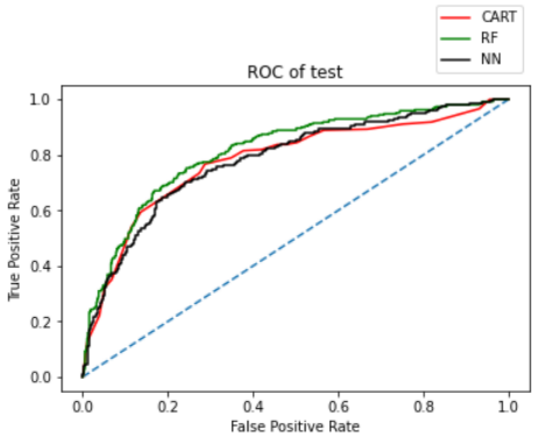
So far we have created three models such as CART, Random Forest, and Artificial neural network and the evaluation is performance across all the three models by training and testing the data. In this step we will compare the results of three models in training and test dataset to view and select which model gives the best for result for making predictions.

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	0.78	0.77	0.77	0.77	0.77	0.76
AUC	0.82	0.78	0.82	0.82	0.80	0.78
Recall	0.51	0.49	0.55	0.54	0.50	0.47
Precision	0.72	0.69	0.68	0.68	0.69	0.68
F1 Score	0.60	0.58	0.61	0.60	0.58	0.56

Comparison of ROC Curve and AUC scores of all models for Training data:





Inferences on Comparison of Model performance:

- Here we built 3 models namely CART, random forest, and artificial neural network. We have checked the
- Problem statement, states that “An Insurance firm providing tour insurance is facing higher claim frequency”, so we need to have clear understanding on False positive and False Negative.

- ✓ False positives are the customers who actually did not claim for the insurance, but the algorithm predicted that they would claim.
- ✓ False Negatives are the customers who actually claimed for the insurance, but the model predicted that won't claim.

- ❖ As a result, we can conclude that false positives will not have a significant impact on our firm, however false negative will impact the prediction. As per the model sensitivity/ recall will be more crucial.
- ❖ All the 3 model doesn't give best result, but it has performed reasonably stable enough to be used for making any future predictions. The train and test values aren't that so far for all the three modules, thus there seems to be no concern in overfitting or underfitting.
- ❖ Comparing the recall value among three models, random forest gives sightly high recall test value as 0.54 than Card model of 0.49 and Artificial Neural model as 0.47.
- ❖ Comparing the Accuracy value among three models, Cart model and random forest gives sightly high accuracy score as 0.77 than Artificial Neural model as 0.76.
- ❖ By Comparing the F1 Score among three models, random forest gives sightly high F1 Score value as 0.60 than Card model of 0.58 and Artificial Neural model as 0.56.
- ❖ Comparing to the AUC score among three models, random forest gives sightly high AUC score as 0.82 than card model of 0.78 and Artificial Neural model as 0.78.
- ❖ From Cart and Random Forest Model, we see that variable Agency_Code is found to be the most useful feature among all other features, and this will be useful in predicting if a customer has claimed the insurance or not.
- ❖ By visualizing the ROC curves of the three models, Random Forest has the slightly high line than other two models.
- ❖ From the above interpretation we conclude that random Forest model slightly higher values in terms of Accuracy, AUC score, Precision, Recall and F1 score. From this we conclude Random Forest model seems to be optimised model.

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

According to the problem statement, the Insurance firm providing tour insurance is facing higher claim frequency. This means that people are filing insurance claims more frequently. The company wants to go into historical data to figure out why the frequency is so high and come up with a strategy to reduce it by using the model performances.

In our extensive analysis so far, we have thoroughly examined historical data and developed a model that predicts claim status based on the characteristics in our dataset. Let us now look at the key points in our past data first and try to find out some recommendations for the firm.

Following are the insights and recommendations to help the management solve the business objective:

Insights from the Graphs and Analysis from EDA:

Claimed: Although the number of insurances claimed recorded was less than half that of insurances not claimed, the sales for both claimed and unclaimed insurances were almost equal. As a result, sales of claimed insurances were significantly higher than those of non-claimed insurances.

Agency: JZI agency has the lowest number of records as well as the least sales for both claimed and unclaimed insurances. For C2B agency, claimed insurances are more than the unclaimed ones; both in terms of number and sales. EPX agency has the highest unclaimed to claimed ratio. The sales are also good and the claimed insurances are very less as well.

Product: Customised plan and Cancellation plan are the best products when it comes to insurance claims. Their Unclaimed to claimed insurance ratio is very high for these. Silver plan is having claimed insurance bookings way more. The sales of these bookings which have claimed insurances are too high as well. Gold plan is the least taken plan.

Booking Type: The people who booked airlines had equal number for both who claimed and did not claim for insurance. However, the sales for those who claimed were way higher than who did not claim. It can be said that higher sales value bookings opted for insurance claim more. Travel Agency performed good. Unclaimed to claimed ratio was considerably less both in terms of numbers and sales.

Channel: Offline bookings are negligible as compared to Online bookings. Number of Online bookings getting claimed for insurance is lower but sales value is higher.

Destination: Bookings for ASIA is the highest. AMERICA and EUROPE have very low bookings as compared to ASIA. In ASIA, number of bookings that got claimed is lower but the sales of bookings which got claimed was higher.

Recommendations:

- Records having a greater sales value should be prioritised since they are more likely to be claimed for insurance.
- JZI agency needs to equip resources to pick up sales as they fall through the cracks, promotional
- marketing campaign can be launched or assess whether the insurance firm need to tie up with another alternate agency.
- The business should promote bookings from Travel Agency
- As a result, insurance firm requires customers to buy airline tickets, as well as cross-sell insurance based on claim data patterns.
- Customers benefited from streamlining online experiences, which resulted in a rise in conversions and, as a result, subsequently raises profits. Online reservations with a higher value should be prioritised.
- The gold plan should be restructured so that more customers choose to use it. The Silver plan should review its procedures and strive to reduce the number of insurance claims.
- ASIA's high sales bookings should be monitored. To attract more clients, several marketing plans for America and Europe tours should be implemented. Perhaps some reductions or special offers can be made.

Key performance indicators of insurance claims:

- 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.