Project – Data Mining

By: Vaishnavi Karelia

PGP - Data Science and Business Analytics

PGPDSBA Online Feb_D 2021

Table Of Contents

List of Tables:

List of Figures:

Problem 1: Clustering

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

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

Reading the data:

Data Dictionary for customer 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)

Data imported successfully!

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	19.940	16.920	0.875	6.675	3.763	3.252	6.550
1	15.990	14.890	0.906	5.363	3.582	3.336	5.144
2	18.950	16.420	0.883	6.248	3.755	3.368	6.148
3	10.830	12.960	0.810	5.278	2.641	5.182	5.185
4	17.990	15.860	0.899	5.890	3.694	2.068	5.837

Table - 1 Market Segmentation Table

Performing Initial Steps

```
Structure of the dataset:
--------------------------------------------------------------------------------

The dataset has 210 rows and 7 columns.

Total elements in this dataset are 1470
```

Missing Values Check

```
--------------------------------------------------------------------------------
There are no missing values in the dataset
```

Info of the dataset

```
--------------------------------------------------------------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   spending                      210 non-null    float64
 1   advance_payments              210 non-null    float64
 2   probability_of_full_payment   210 non-null    float64
 3   current_balance               210 non-null    float64
 4   credit_limit                  210 non-null    float64
 5   min_payment_amt               210 non-null    float64
 6   max_spent_in_single_shopping  210 non-null    float64
dtypes: float64(7)
memory usage: 11.6 KB
None
```

Table - 2 Market Segmentation Table Info

Duplicate Values Check

```
--------------------------------------------------------------------------------
Number of duplicate rows = 0
```

spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping

Inferences

The market dataset has 210 rows and 7 columns all of them with float64 dtypes. The data has no missing values and also there are no duplicate rows. The variables are spending, advance_payments, probability_of_full_payment, current_balance, credit_limit, min_payment_amt, max_spent_in_single_shopping. The memory usage is 11.6 KB.

Performing Univariate Analysis

This data set has no categorical values! Perform statistical analysis for numerical values only.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	210.000	210.000	210.000	210.000	210.000	210.000	210.000
mean	14.848	14.559	0.871	5.629	3.259	3.700	5.408
std	2.910	1.306	0.024	0.443	0.378	1.504	0.491
min	10.590	12.410	0.808	4.899	2.630	0.765	4.519
25%	12.270	13.450	0.857	5.262	2.944	2.561	5.045
50%	14.355	14.320	0.873	5.524	3.237	3.599	5.223
75%	17.305	15.715	0.888	5.980	3.562	4.769	5.877
max	21.180	17.250	0.918	6.675	4.033	8.456	6.550

Table - 3 Statistical Summary (Numerical)

Univariate Analysis for column: spending

Statistical Inferences

```
count   210.000
mean     14.848
std       2.910
min      10.590
25%      12.270
50%      14.355
75%      17.305
max      21.180
Name: spending, dtype: float64
```

Shapiro-Wilk test for normality

```
pvalue for spending column is 0.0,
Hence, we reject the null hypothesis that the data is normally distributed
```

Detecting outliers using z-score

```
spending variable does not have any outliers
```

Univariate Analysis for column: advance_payments

Statistical Inferences

```
count   210.000
mean     14.559
std       1.306
min      12.410
25%      13.450
50%      14.320
75%      15.715
max      17.250
Name: advance_payments, dtype: float64
```

Shapiro-Wilk test for normality

---
pvalue for advance_payments column is 0.0,
Hence, we reject the null hypothesis that the data is normally distributed

Detecting outliers using z-score

---
advance_payments variable does not have any outliers

Univariate Analysis for column: probability_of_full_payment

Statistical Inferences

---
count   210.000
mean   0.871
std   0.024
min   0.808
25%   0.857
50%   0.873
75%   0.888
max   0.918
Name: probability_of_full_payment, dtype: float64

Shapiro-Wilk test for normality

---
pvalue for probability_of_full_payment column is 0.0005,
Hence, we reject the null hypothesis that the data is normally distributed

Detecting outliers using z-score

---
probability_of_full_payment variable does not have any outliers

Univariate Analysis for column: current_balance

Statistical Inferences

---
count   210.000
mean   5.629
std   0.443
min   4.899
25%   5.262
50%   5.524
75%   5.980
max   6.675
Name: current_balance, dtype: float64

Shapiro-Wilk test for normality

---
pvalue for current_balance column is 0.0,
Hence, we reject the null hypothesis that the data is normally distributed

Detecting outliers using z-score

---
current_balance variable does not have any outliers

Univariate Analysis for column: credit_limit

Statistical Inferences

```
count   210.000
mean      3.259
std       0.378
min       2.630
25%       2.944
50%       3.237
75%       3.562
max       4.033
Name: credit_limit, dtype: float64
```

Shapiro-Wilk test for normality

---

pvalue for credit_limit column is 0.0,
Hence, we reject the null hypothesis that the data is normally distributed

Detecting outliers using z-score

---

credit_limit variable does not have any outliers

Univariate Analysis for column: min_payment_amt

Statistical Inferences

```
count   210.000
mean      3.700
std       1.504
min       0.765
25%       2.561
50%       3.599
75%       4.769
max       8.456
Name: min_payment_amt, dtype: float64
```

Shapiro-Wilk test for normality

---

pvalue for min_payment_amt column is 0.0154,
Hence, we reject the null hypothesis that the data is normally distributed

Detecting outliers using z-score

---

min_payment_amt variable has outliers

Univariate Analysis for column: max_spent_in_single_shopping

Statistical Inferences

```
count   210.000
mean      5.408
std       0.491
min       4.519
25%       5.045
```

```
50%       5.223
75%       5.877
max       6.550
Name: max_spent_in_single_shopping, dtype: float64
```

Shapiro-Wilk test for normality

```
-----------------------------------------------------------
pvalue for max_spent_in_single_shopping column is 0.0,
Hence, we reject the null hypothesis that the data is normally distributed
```

Detecting outliers using z-score

```
-----------------------------------------------------------
max_spent_in_single_shopping variable does not have any outliers
```

Inferences

The data only have numerical/float values. We can assume that the data are well spread around the mean. Standard deviation tells you how spread out the data is. It is a measure of how far each observed value is from the mean. In any distribution, about 95% of values will be within 2 standard deviations of the mean. Low standard deviation means data are clustered around the mean, and high standard deviation indicates data are more spread out. Let's further analysis the spread

Boxplot and Distribution plot for all the features:

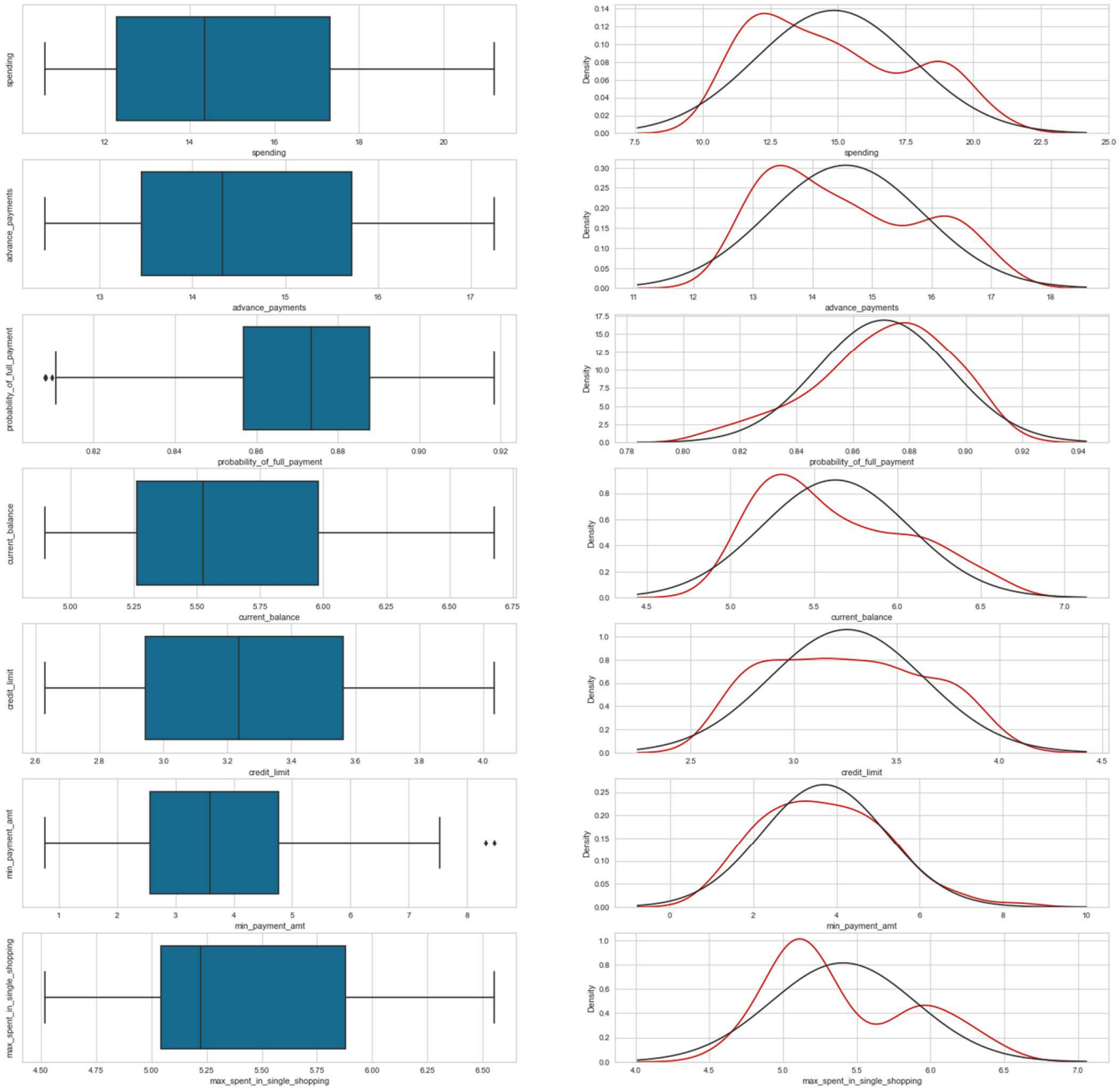


Figure - 1 Boxplot and Distribution Plot (Univariate Analysis)

Inference

Shapiro Wilkins Test for normality *calculates whether a random sample comes from (specifically) a normal distribution. Based on this test we concluded that the data is not normally distributed.*

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2},$$

Z-score check for outliers

Z score = (x -mean) / std. deviation

If the z score of a data point is more than 3, it indicates that the data point is quite different from the other data points. Such a data point can be an outlier.

Performing Multivariate Analysis

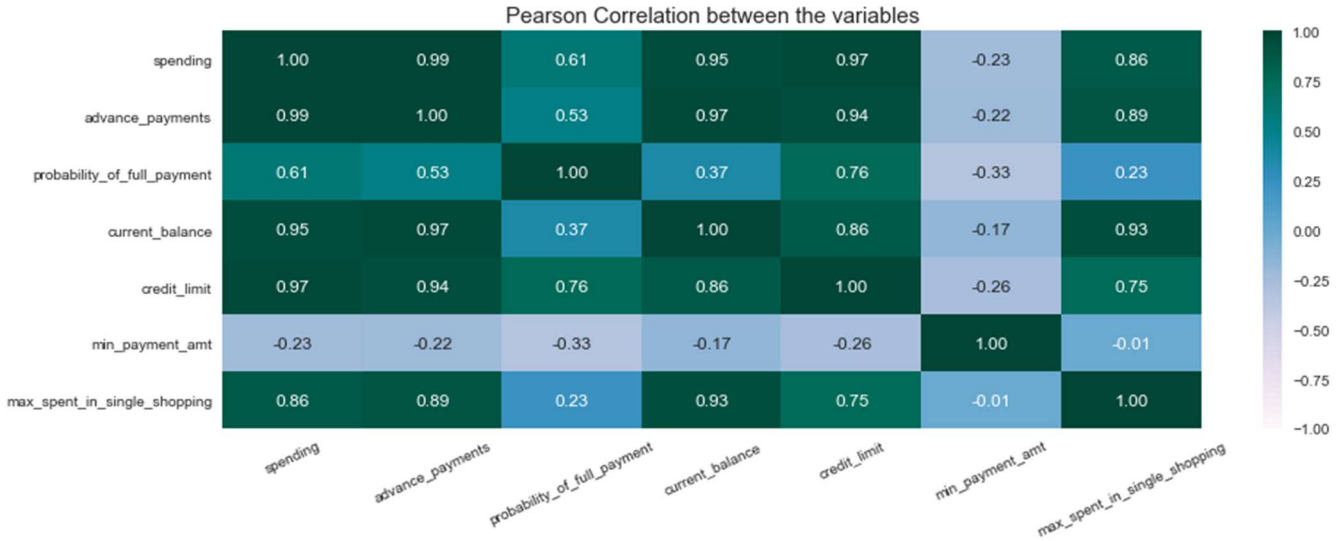


Figure - 2 Heatmap (Multivariate Analysis)

Inference

---

spending, advance_payments,probability_of_full_payment,current_balance, credit_limit and max_spent_in_single_shopping is positively correlated to each other. min_payment_amt is the only column that is negatively correlated to the rest of them. All the positively correlated columns have the correlation closer to 1 which means that there is a strong relation between all these variables. min_payment_amt and max_spent_in_single_shopping has correlation of -

0.01 which is closer to zero, this concludes that there is no correlation between these variables. min_payment_amt has negative correlation with rest of the variables which means if the other variables increase the min_payment_amt decreases.

Performing Bivariate Analysis

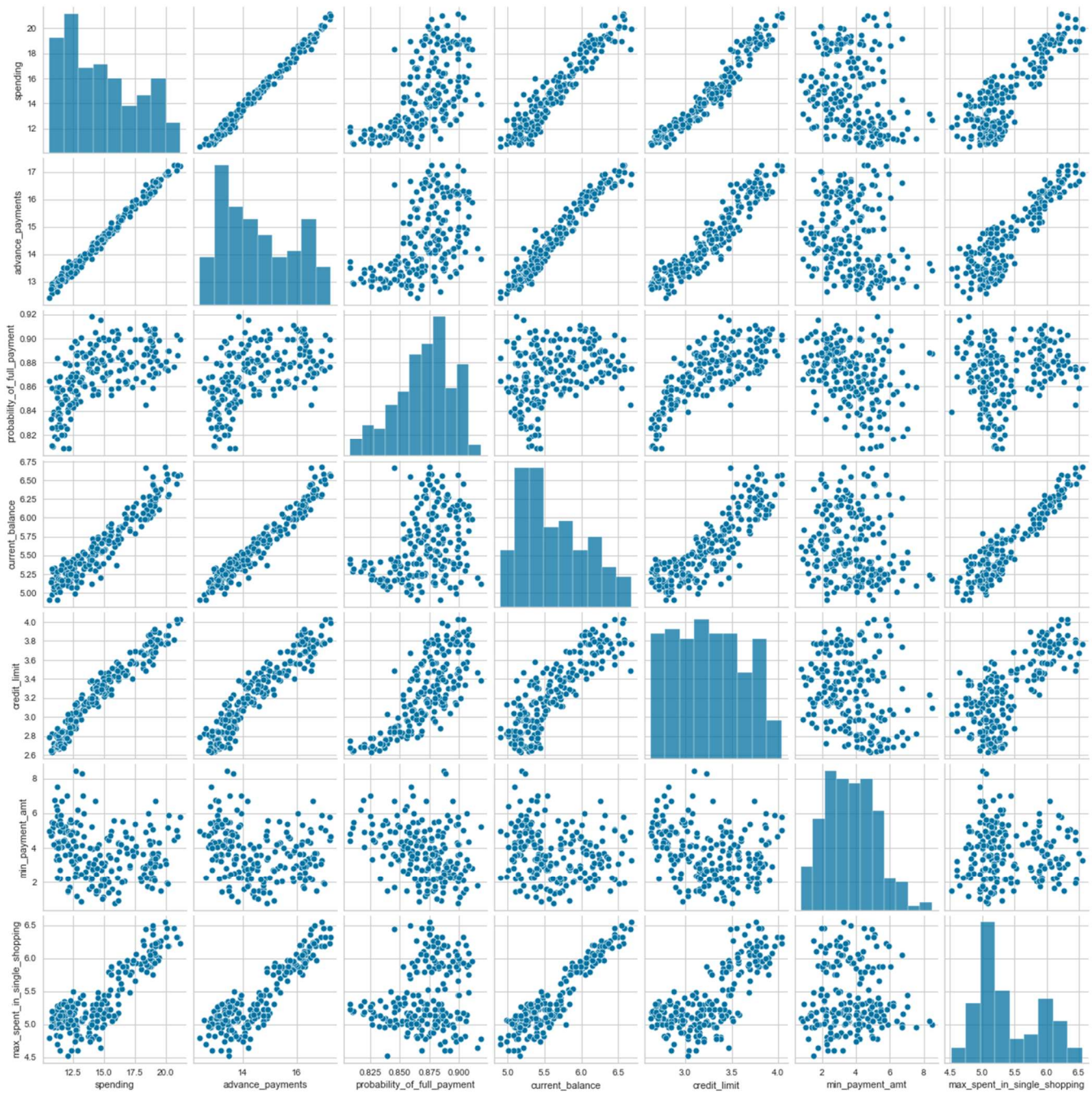


Figure - 3 Pairplot (Bivariate Analysis)

Inference

The data is positively correlated with each other except for min_payment_amt which negatively correlated with rest of the columns. Negative or inverse correlation describes when two variables tend to move in opposite size and direction from one another, such that when one increases the other decreases.

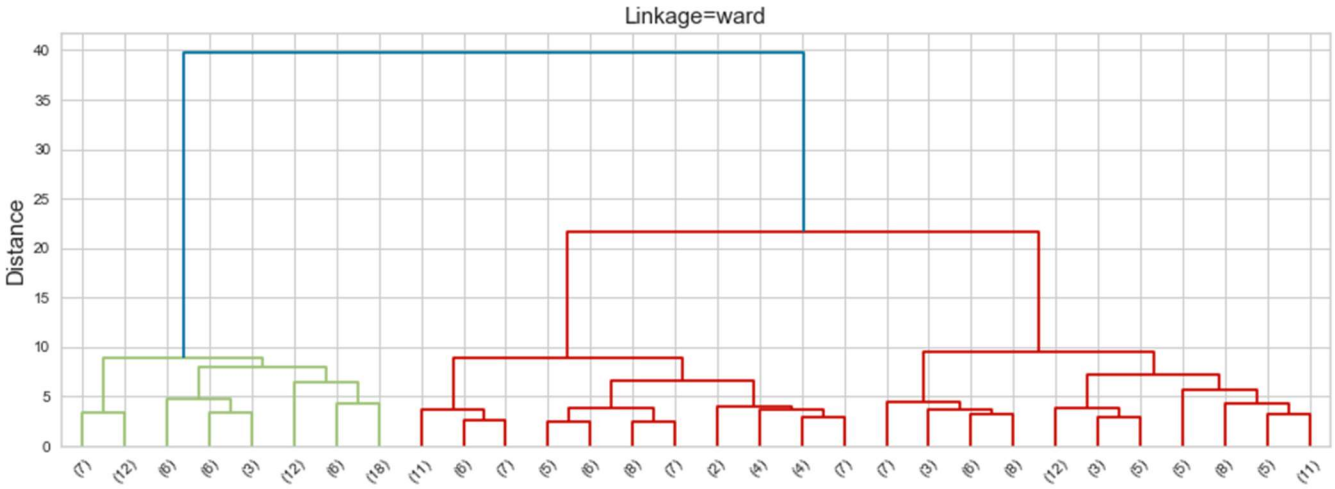
1.2 Do you think scaling is necessary for clustering in this case? Justify

A scaling transformation alters size of an object. In the scaling process, we either compress or expand the dimension of the object. In this case some of the variables are in 100s and others are in 1000s which means that at some point the variables with higher scale will dominate while calculating distances. For example, the Euclidean distance measure is sensitive to magnitudes and hence should be scaled for all features to weigh in equally.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754	1.812	0.178	2.368	1.339	-0.299	2.329
1	0.394	0.254	1.502	-0.601	0.858	-0.243	-0.539
2	1.413	1.428	0.505	1.401	1.317	-0.221	1.509
3	-1.384	-1.228	-2.592	-0.793	-1.639	0.988	-0.455
4	1.083	0.998	1.196	0.592	1.155	-1.088	0.875

Table - 4 Scaled Data (Market Segmentation)

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



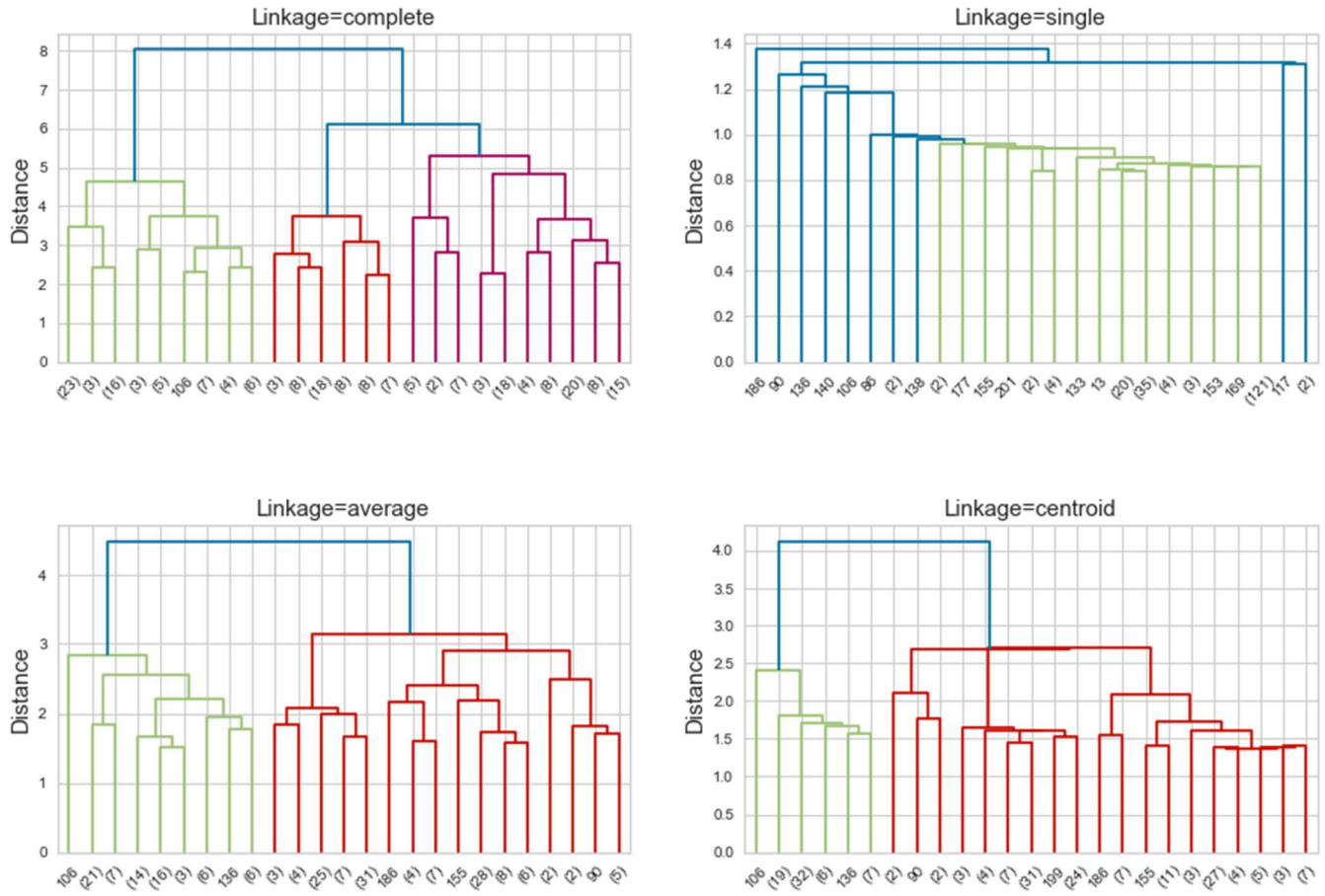
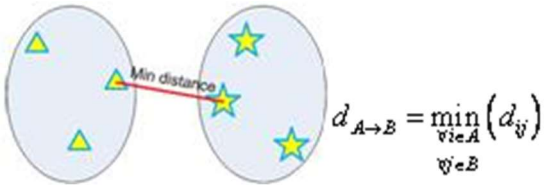


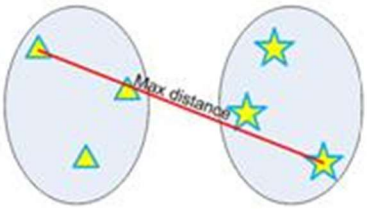
Figure - 4 Dendrogram

The above dendrogram is based on 'complete', 'single', 'average', 'centroid' and 'ward' linkage methods with metrics as Euclidean distance. Euclidean distance is the shortest path between source and destination which is a straight line.

Single Linkage: *For two clusters A and B, the single linkage returns the minimum distance between two points.*

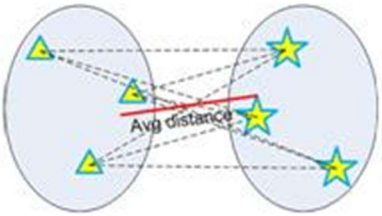


Complete Linkage: *For two clusters A and B, the single linkage returns the maximum distance between two points.*



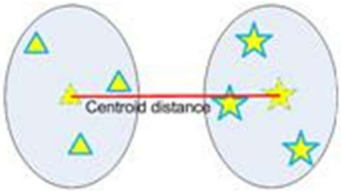
$$d_{A\to B} = \max_{\substack{\forall i\in A \\ \forall j\in B}} \left(d_{ij}\right)$$

Average Linkage: *For two clusters A and B, first for the distance between any data-point in A and any data-point in B and then the arithmetic mean of these distances are calculated. Average Linkage returns this value of the arithmetic mean.*



$$d_{A\to B} = \operatorname{average}_{\substack{\forall i\in A \\ \forall j\in B}} \left(d_{ij}\right)$$

Centroid Linkage: *The distance between two clusters A and B is the distance between the two mean vectors of the clusters. At each stage of the process, we combine the two clusters that have the smallest centroid distance.*



$$d_{A\to B} = \left\|\mathbf{c}_A - \mathbf{c}_B\right\| = \frac{1}{n_i n_j} \sum_{\substack{\forall i\in A \\ \forall j\in B}} \left(d_{ij}\right)$$

Ward's Linkage: *Like other clustering methods, Ward's method starts with n clusters, each containing a single object. These n clusters are combined to make one single cluster.*

$$d_{k\to(r,s)} = \alpha_r d_{k\to r} + \alpha_s d_{k\to s} + \beta d_{r\to s} + \gamma \left|d_{k\to r} - d_{k\to s}\right|$$

Ward Linkage as it says that the distance between two clusters, A and B, is how much the sum of squares will increase when we merge them. With hierarchical clustering, the sum of squares starts out at zero (because every point is in its own cluster) and then grows as we merge clusters. Ward's method keeps this growth as small as possible.

For better analysis, I'll go ahead with the total number of clusters as 3.

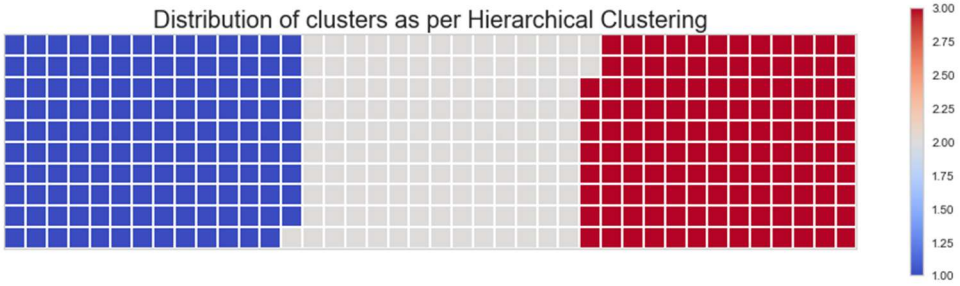


Figure - 5 Waffle Chart (Hierarchical Clustering)

```
Cluster 1 (Blue) has 73 records
Cluster 2 (Grey) has 70 records
Cluster 3 (Red) has 67 records
```

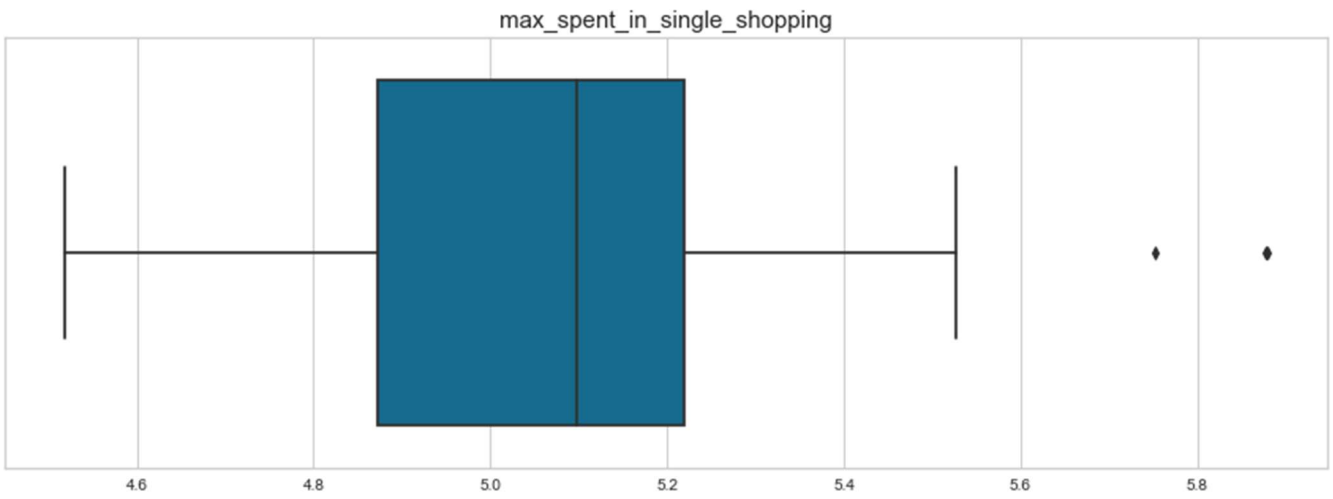
Profiling Clusters:

Cluster 1:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Cluster
1	15.990	14.890	0.906	5.363	3.582	3.336	5.144	Cluster 1
7	13.740	14.050	0.874	5.482	3.114	2.932	4.825	Cluster 1
11	14.090	14.410	0.853	5.717	3.186	3.920	5.299	Cluster 1
14	12.100	13.150	0.879	5.105	2.941	2.201	5.056	Cluster 1
16	16.140	14.990	0.903	5.658	3.562	1.355	5.175	Cluster 1

Table - 5 Cluster 1 (H-Cluster)

Visualizing Cluster 1:



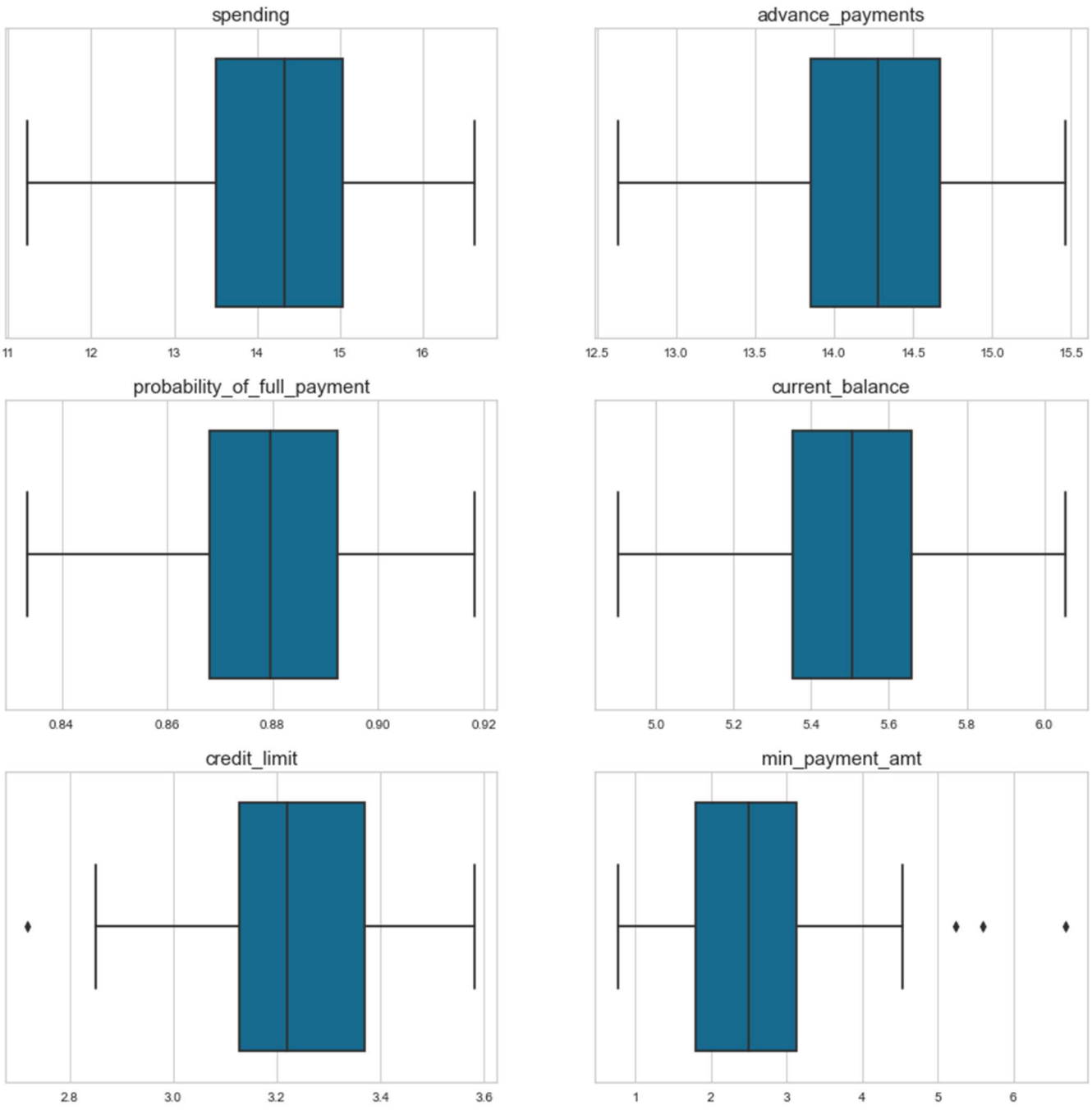


Figure - 6 Visualizing Cluster 1 (H-cluster)

```
The average spending (in 1000s) for Cluster 1 customers is: 14.199
50% of the customers have spending (in 1000s) of 14.33, 95% have of 16.05 and 99% h
ave of 16.3204.
Interquartile range for spending (in 1000s) is 28.53. IQR tells us the range of the
 middle half of the data.

The average advance_payments (in 100s) for Cluster 1 customers is: 14.2336
```

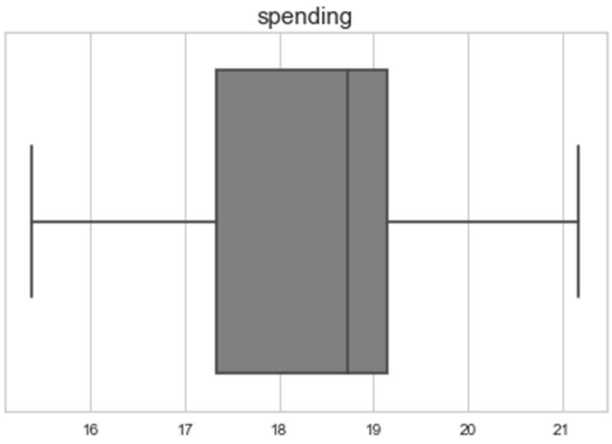
```
50% of the customers have advance_payments (in 100s) of 14.28, 95% have of 15.126 a
nd 99% have of 15.3232.
Interquartile range for advance_payments (in 100s) is 28.52. IQR tells us the range
 of the middle half of the data.

The average probability_of_full_payment for Cluster 1 customers is: 0.8792
50% of the customers have probability_of_full_payment of 0.8796, 95% have of 0.9053
 and 99% have of 0.9161.
Interquartile range for probability_of_full_payment is 1.7603. IQR tells us the ran
ge of the middle half of the data.

The average current_balance (in 1000s) for Cluster 1 customers is: 5.4782
50% of the customers have current_balance (in 1000s) of 5.504, 95% have of 5.8284 a
nd 99% have of 5.9572.
Interquartile range for current_balance (in 1000s) is 11.009. IQR tells us the rang
e of the middle half of the data.

The average credit_limit (in 10000s) for Cluster 1 customers is: 3.2265
50% of the customers have credit_limit (in 10000s) of 3.221, 95% have of 3.4824 and
 99% have of 3.5676.
Interquartile range for credit_limit (in 10000s) is 6.5. IQR tells us the range of
the middle half of the data.

The average min_payment_amt (in 100s) for Cluster 1 customers is: 2.6122
50% of the customers have min_payment_amt (in 100s) of 2.504, 95% have of 4.3282 an
d 99% have of 5.8988.
Interquartile range for min_payment_amt (in 100s) is 4.927. IQR tells us the range
of the middle half of the data.

The average max_spent_in_single_shopping (in 1000s) for Cluster 1 customers is: 5.0
862
50% of the customers have max_spent_in_single_shopping (in 1000s) of 5.097, 95% hav
e of 5.503 and 99% have of 5.8776.
Interquartile range for max_spent_in_single_shopping (in 1000s) is 10.092. IQR tell
s us the range of the middle half of the data.
```

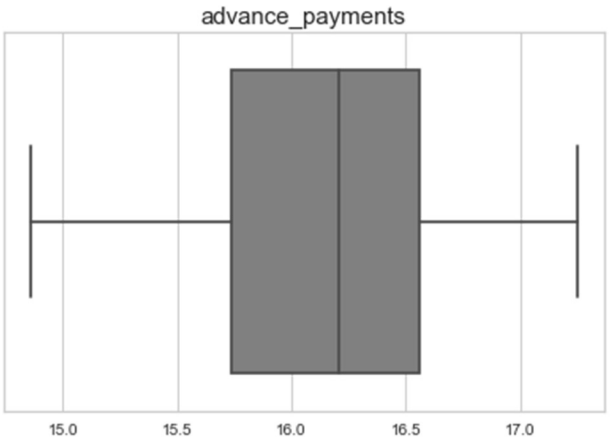
Cluster 2:

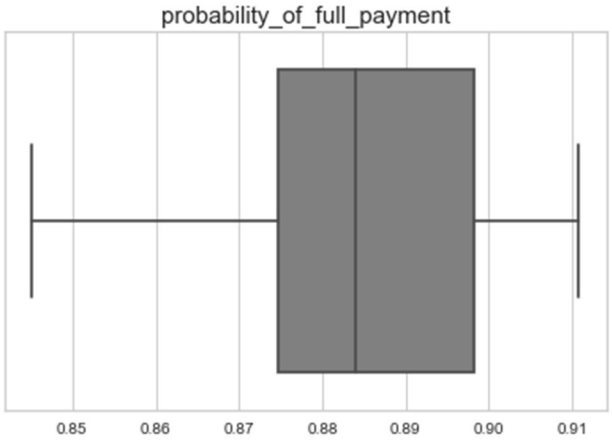
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Cluster
0	19.940	16.920	0.875	6.675	3.763	3.252	6.550	Cluster 2
2	18.950	16.420	0.883	6.248	3.755	3.368	6.148	Cluster 2
4	17.990	15.860	0.899	5.890	3.694	2.068	5.837	Cluster 2
8	18.170	16.260	0.864	6.271	3.512	2.853	6.273	Cluster 2
10	18.550	16.220	0.886	6.153	3.674	1.738	5.894	Cluster 2

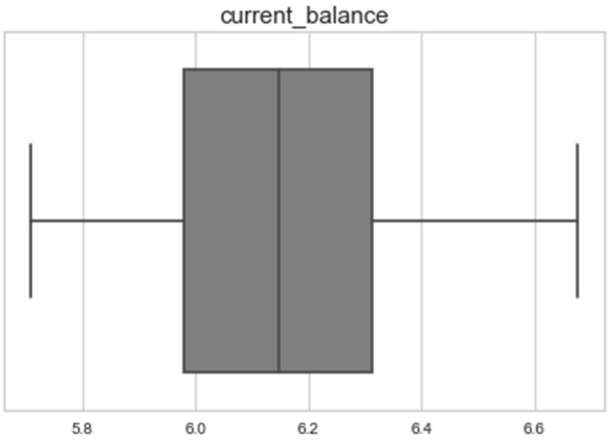
Table - 6 Cluster 2 (Hierarchical Clustering)

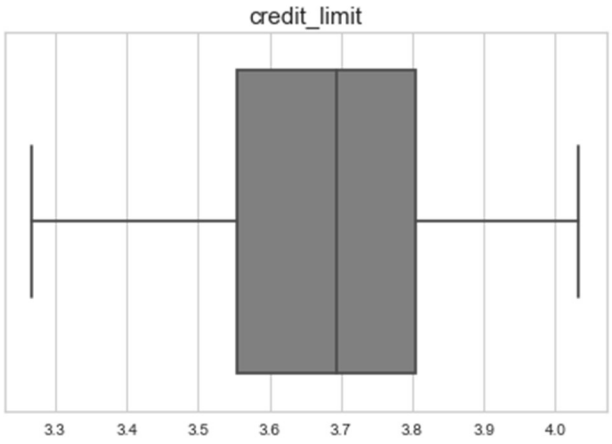
Visualizing Cluster 2:

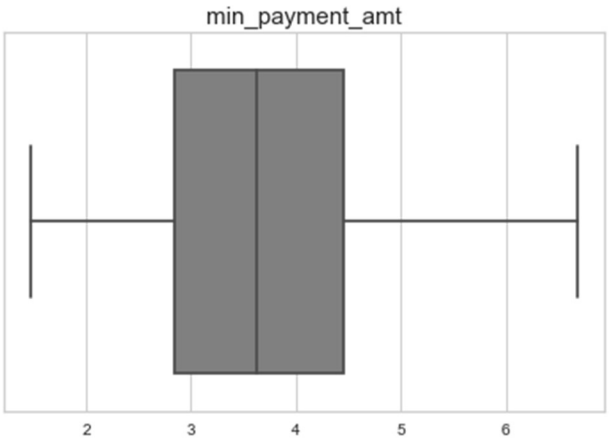

spending
16
17
18
19
20
21


advance_payments
15.0
15.5
16.0
16.5
17.0


probability_of_full_payment
0.85
0.86
0.87
0.88
0.89
0.90
0.91


current_balance
5.8
6.0
6.2
6.4
6.6


credit_limit
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4.0


min_payment_amt
2
3
4
5
6

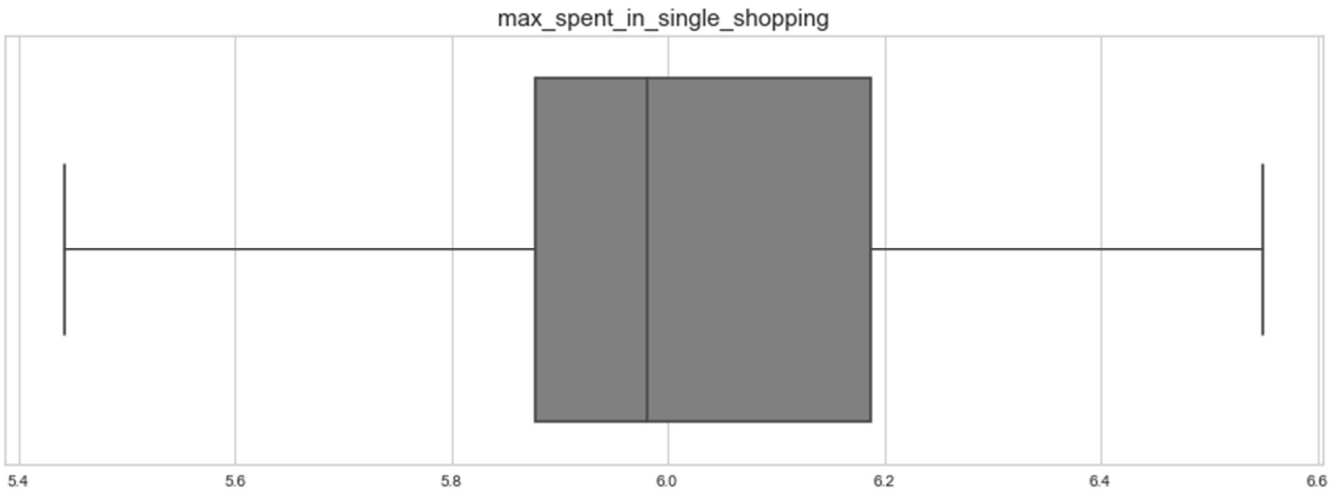


Figure - 7 Visualizing Cluster 2 (H-Cluster)

The average spending (in 1000s) for Cluster 2 customers is: 18.3714
50% of the customers have spending (in 1000s) of 18.72, 95% have of 20.4985 and 99% have of 21.0351.
Interquartile range for spending (in 1000s) is 36.4675. IQR tells us the range of the middle half of the data.

The average advance_payments (in 100s) for Cluster 2 customers is: 16.1454
50% of the customers have advance_payments (in 100s) of 16.21, 95% have of 17.041 and 99% have of 17.2362.
Interquartile range for advance_payments (in 100s) is 32.295. IQR tells us the range of the middle half of the data.

The average probability_of_full_payment for Cluster 2 customers is: 0.8844
50% of the customers have probability_of_full_payment of 0.884, 95% have of 0.9077 and 99% have of 0.9089.
Interquartile range for probability_of_full_payment is 1.7729. IQR tells us the range of the middle half of the data.

The average current_balance (in 1000s) for Cluster 2 customers is: 6.1582
50% of the customers have current_balance (in 1000s) of 6.1485, 95% have of 6.5763 and 99% have of 6.6688.
Interquartile range for current_balance (in 1000s) is 12.2913. IQR tells us the range of the middle half of the data.

The average credit_limit (in 10000s) for Cluster 2 customers is: 3.6846
50% of the customers have credit_limit (in 10000s) of 3.6935, 95% have of 3.9476 and 99% have of 4.0323.
Interquartile range for credit_limit (in 10000s) is 7.359. IQR tells us the range of the middle half of the data.

The average min_payment_amt (in 100s) for Cluster 2 customers is: 3.6392
50% of the customers have min_payment_amt (in 100s) of 3.629, 95% have of 5.6684 and 99% have of 6.2121.
Interquartile range for min_payment_amt (in 100s) is 7.3048. IQR tells us the range of the middle half of the data.

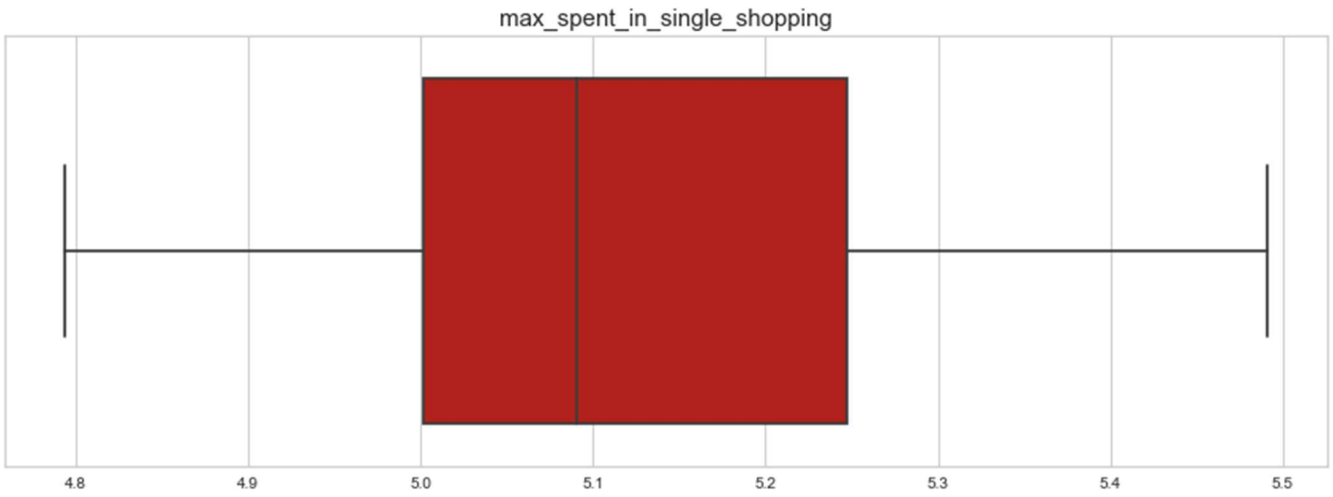
```
The average max_spent_in_single_shopping (in 1000s) for Cluster 2 customers is: 6.0
174
50% of the customers have max_spent_in_single_shopping (in 1000s) of 5.9815, 95% ha
ve of 6.4501 and 99% have of 6.5141.
Interquartile range for max_spent_in_single_shopping (in 1000s) is 12.0648. IQR tel
ls us the range of the middle half of the data.
```

Cluster 3:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Cluster
3	10.830	12.960	0.810	5.278	2.641	5.182	5.185	Cluster 3
5	12.700	13.410	0.887	5.183	3.091	8.456	5.000	Cluster 3
6	12.020	13.330	0.850	5.350	2.810	4.271	5.308	Cluster 3
9	11.230	12.880	0.851	5.140	2.795	4.325	5.003	Cluster 3
12	12.150	13.450	0.844	5.417	2.837	3.638	5.338	Cluster 3

Table - 7 Cluster 3 (H-cluster)

Visualizing Cluster 3:



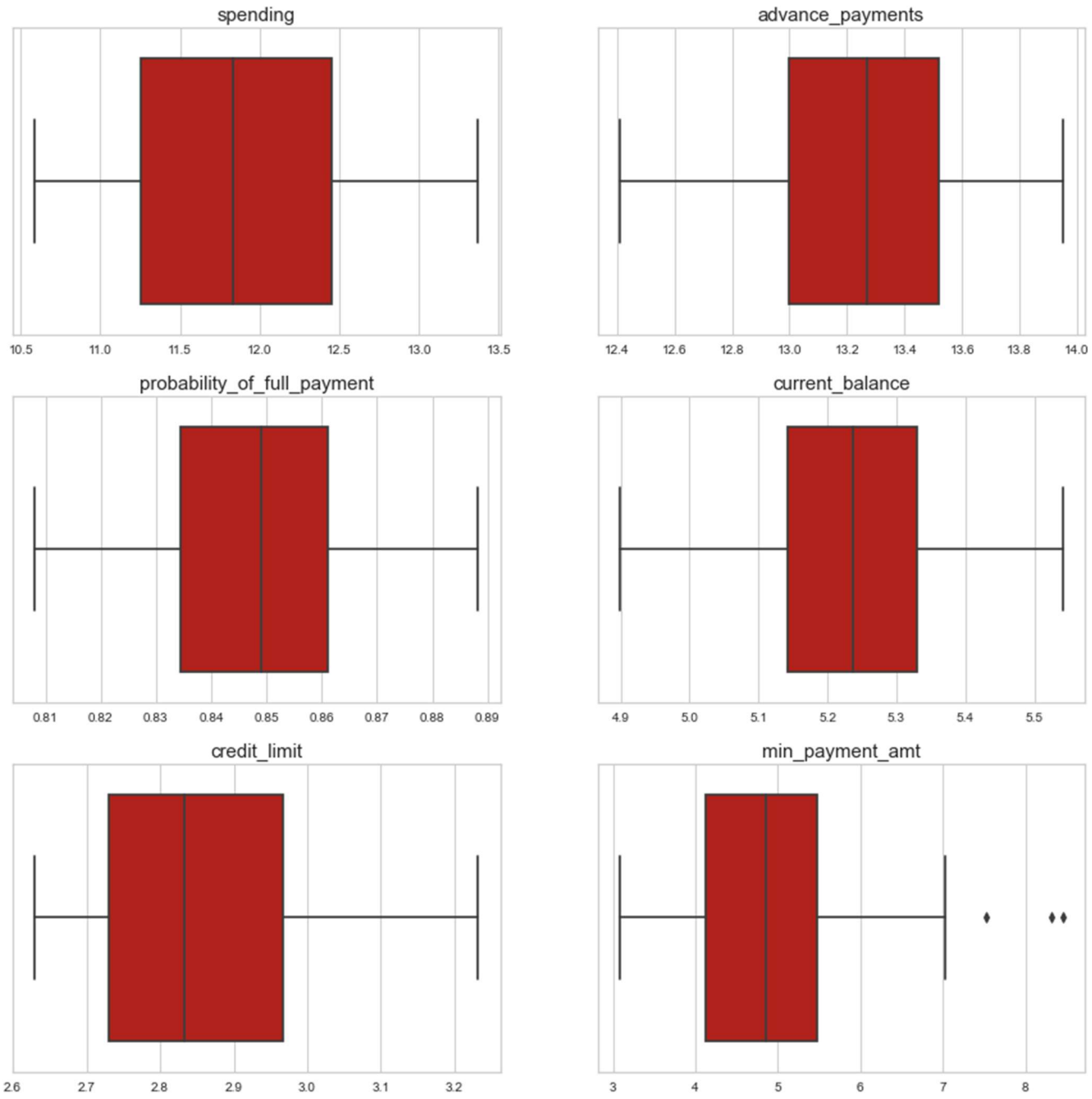


Figure - 8 Visualizing Cluster 3 (H-cluster)

The average spending (in 1000s) for Cluster 3 customers is: 11.8724
50% of the customers have spending (in 1000s) of 11.83, 95% have of 13.161 and 99% have of 13.3502.
Interquartile range for spending (in 1000s) is 23.7. IQR tells us the range of the middle half of the data.

The average advance_payments (in 100s) for Cluster 3 customers is: 13.257

```
50% of the customers have advance_payments (in 100s) of 13.27, 95% have of 13.777 a
nd 99% have of 13.9434.
Interquartile range for advance_payments (in 100s) is 26.52. IQR tells us the range
 of the middle half of the data.

The average probability_of_full_payment for Cluster 3 customers is: 0.8481
50% of the customers have probability_of_full_payment of 0.8491, 95% have of 0.8833
 and 99% have of 0.8877.
Interquartile range for probability_of_full_payment is 1.6955. IQR tells us the ran
ge of the middle half of the data.

The average current_balance (in 1000s) for Cluster 3 customers is: 5.2389
50% of the customers have current_balance (in 1000s) of 5.236, 95% have of 5.4489 a
nd 99% have of 5.5106.
Interquartile range for current_balance (in 1000s) is 10.4715. IQR tells us the ran
ge of the middle half of the data.

The average credit_limit (in 10000s) for Cluster 3 customers is: 2.8485
50% of the customers have credit_limit (in 10000s) of 2.833, 95% have of 3.0859 and
 99% have of 3.1634.
Interquartile range for credit_limit (in 10000s) is 5.698. IQR tells us the range o
f the middle half of the data.

The average min_payment_amt (in 100s) for Cluster 3 customers is: 4.9494
50% of the customers have min_payment_amt (in 100s) of 4.857, 95% have of 7.0221 an
d 99% have of 8.3629.
Interquartile range for min_payment_amt (in 100s) is 9.5875. IQR tells us the range
 of the middle half of the data.

The average max_spent_in_single_shopping (in 1000s) for Cluster 3 customers is: 5.1
222
50% of the customers have max_spent_in_single_shopping (in 1000s) of 5.091, 95% hav
e of 5.3576 and 99% have of 5.4573.
Interquartile range for max_spent_in_single_shopping (in 1000s) is 10.249. IQR tell
s us the range of the middle half of the data.
```

From the above inferences we can note that credit usage for customers in Cluster 1 is average for Cluster 2 is high and for Cluster 3 is low:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Cluster	Credit Usage
0	19.940	16.920	0.875	6.675	3.763	3.252	6.550	Cluster 2	High
1	15.990	14.890	0.906	5.363	3.582	3.336	5.144	Cluster 1	Avergae
2	18.950	16.420	0.883	6.248	3.755	3.368	6.148	Cluster 2	High
3	10.830	12.960	0.810	5.278	2.641	5.182	5.185	Cluster 3	Low
4	17.990	15.860	0.899	5.890	3.694	2.068	5.837	Cluster 2	High

Table - 8 Cluster Profile (as per Hierarchical Clustering)

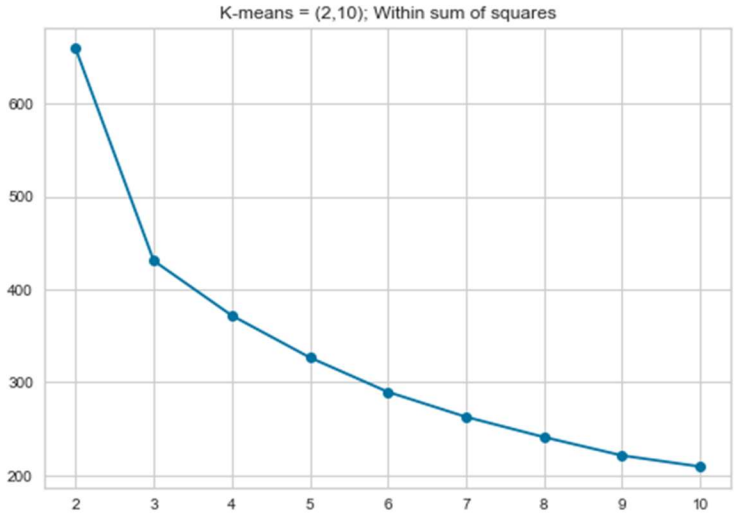
1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Explain the results properly. Interpret and write inferences on the finalized clusters.

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Where k defines the pre-defined total numer of clusters.

Below table is within sum of squares to check optimum number of clusters. To decide optimum number of clusters we will choose the range from 2 to 10 i.e., minimum number of clusters will be 2 till the total number of clusters reaches up to 10.

	0	1	2	3	4	5	6	7	8
Num_of_clusters	2.000	3.000	4.000	5.000	6.000	7.000	8.000	9.000	10.000
WSS	659.172	430.659	371.302	326.935	289.425	262.372	241.956	221.304	206.646

Table - 9 WSS Table



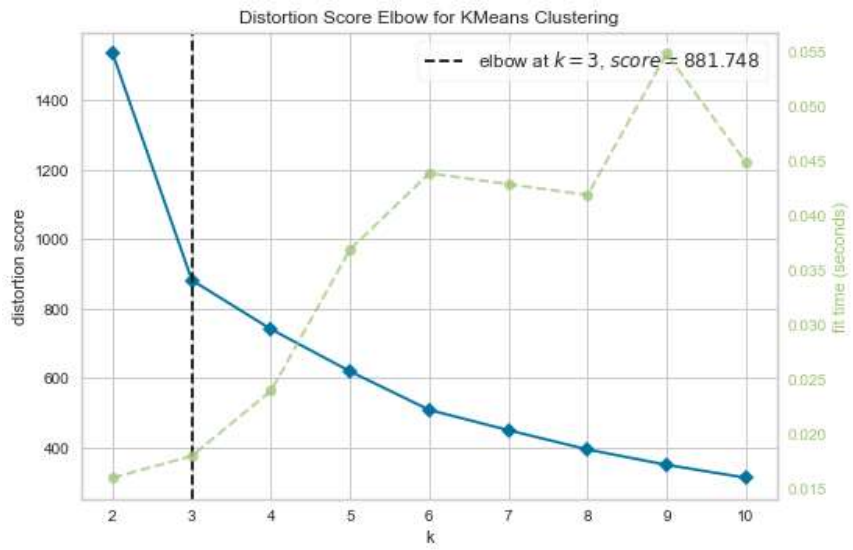
From WSS plot we can see that the optimum number of clusters is when k=3.

The within-cluster sum of squares is a measure of the variability of the observations within each cluster.

Figure - 9 WSS Plot

Let's check the Elbow curve to determine optimum number of clusters.

The elbow method runs k-means clustering on the dataset for a range of values for k (say from 1-10) and then for each value of k computes an average score for all clusters.



From this curve we can assume that the optimum number of clusters we can take is 3, where distortion score is 881.748

Distortion Score: It is calculated as the average of the squared distances from the cluster centers of the respective clusters. Typically, the Euclidean distance metric is used. Inertia: It is the sum of squared distances of samples to their closest cluster center.

Figure - 10 Elbow Plot

Next step is to check the Silhouette score for where k=3 and k=5:

Silhouette Coefficient or Silhouette score is a metric used to calculate the goodness of a clustering technique. Its value ranges from -1 to 1 (1: Means clusters are well apart from each other and clearly distinguished.)

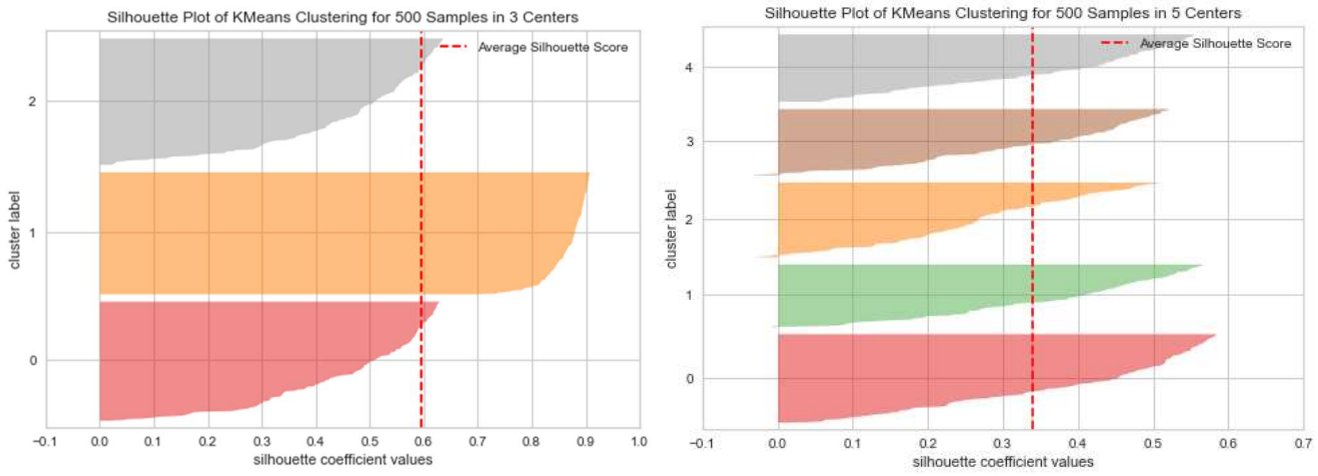
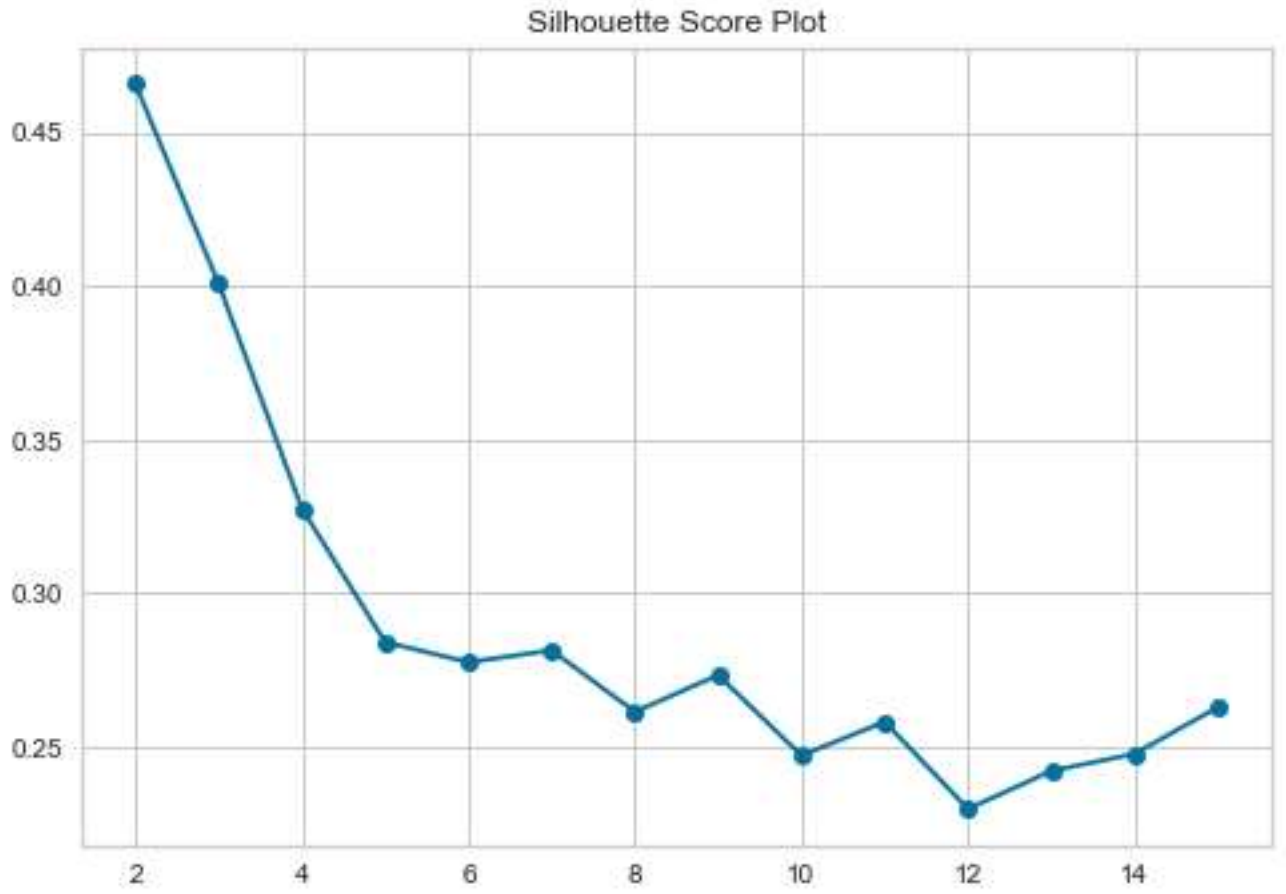


Figure - 11 Silhouette Plot (k=3 and k=5)

```
Silhouette Score of Model 1 (k=3): 0.401
Silhouette Score of Model 2 (k=5): 0.283

Silhouette Score of Model 1 (k=3) is better than Silhouette Score of Model 2 (k
=5)
```

From the below Silhouette Score Plot we will check the goodness of the model when we take n_clusters from the range of 2 to 15:



As per the Silhouette score and the plot the ideal number of clusters we should go is 3.

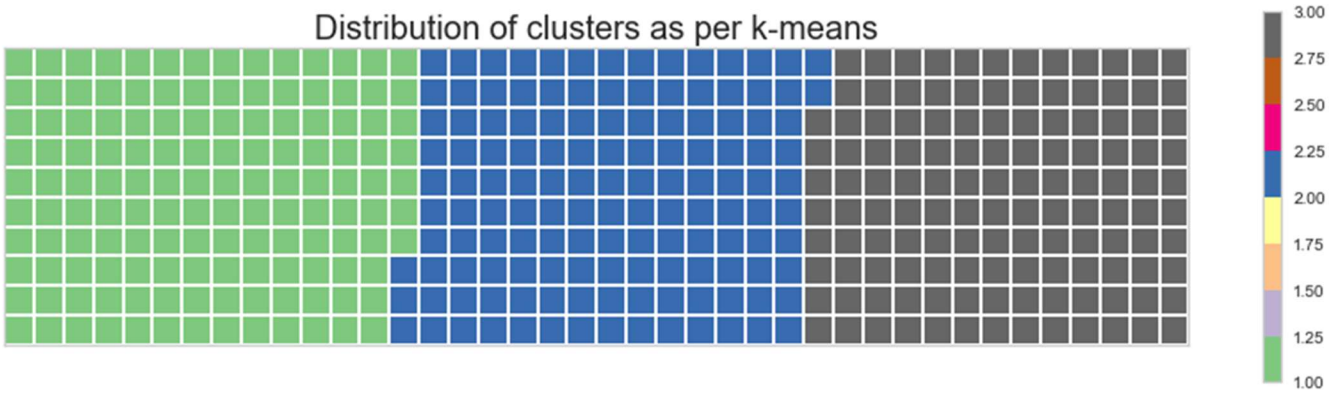


Figure - 12 Waffle Chart (K-means Clustering)

```
Cluster 1 (Green) has 72 records
```

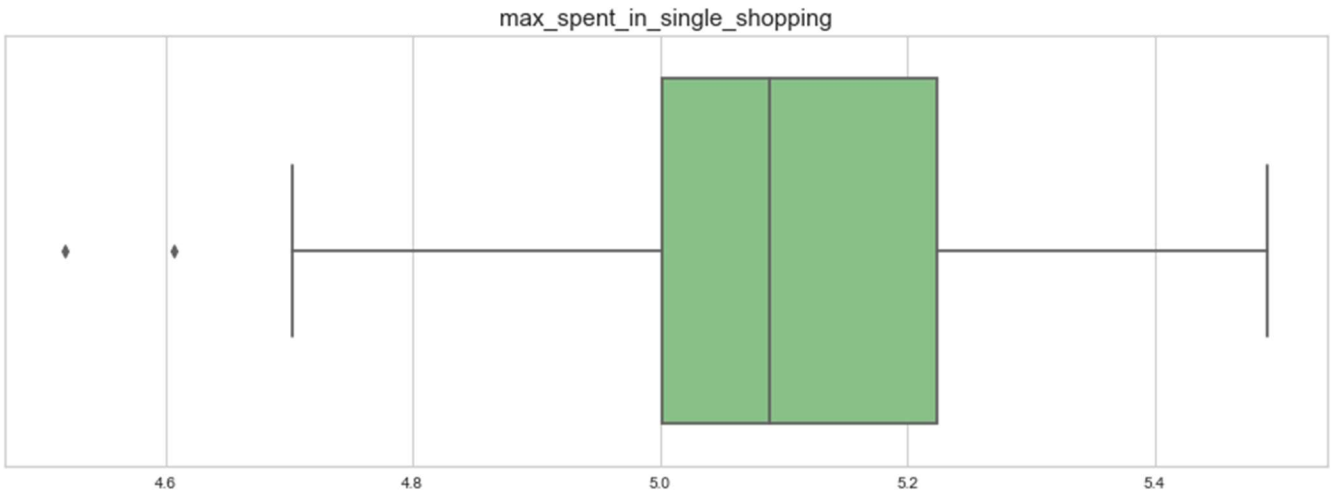
```
Cluster 2 (Blue) has 71 records
Cluster 3 (Black) has 67 records
```

Profiling Clusters as per K-means

Cluster 1:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Cluster
3	10.830	12.960	0.810	5.278	2.641	5.182	5.185	Cluster-1
5	12.700	13.410	0.887	5.183	3.091	8.456	5.000	Cluster-1
6	12.020	13.330	0.850	5.350	2.810	4.271	5.308	Cluster-1
9	11.230	12.880	0.851	5.140	2.795	4.325	5.003	Cluster-1
12	12.150	13.450	0.844	5.417	2.837	3.638	5.338	Cluster-1

Table - 10 Cluster 1 (K-means)



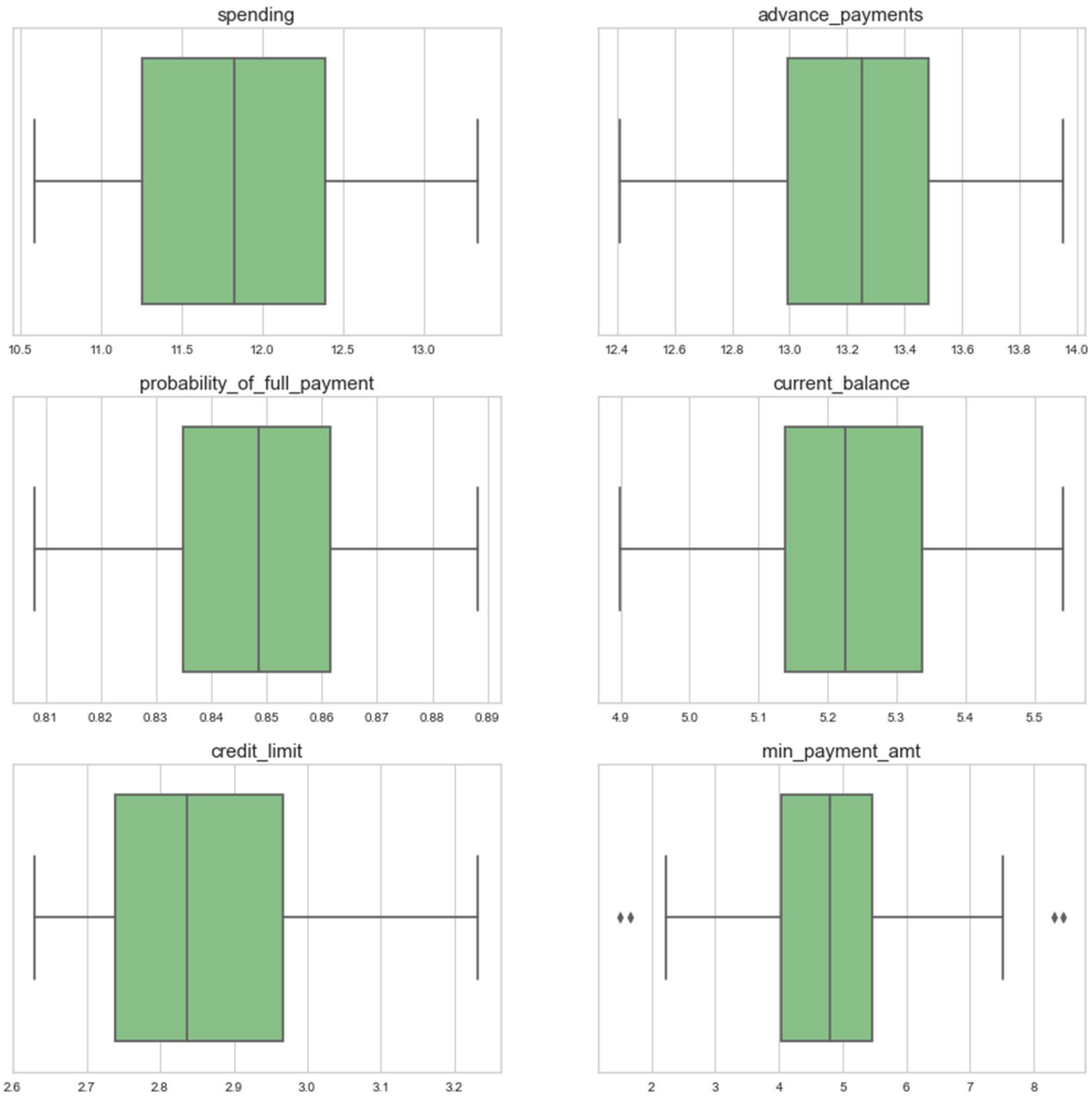


Figure - 13 Visualizing Cluster 1 (K-means)

```
The average spending (in 1000s) for Cluster 1 customers is: 14.4379
50% of the customers have spending (in 1000s) of 14.43, 95% have of 16.13 and 9
9% have of 16.272.
Interquartile range for spending (in 1000s) is 29.08. IQR tells us the range of
 the middle half of the data.

The average advance_payments (in 100s) for Cluster 1 customers is: 14.3377
```

50% of the customers have advance_payments (in 100s) of 14.39, 95% have of 15.1
3 and 99% have of 15.256.
Interquartile range for advance_payments (in 100s) is 28.79. IQR tells us the r
ange of the middle half of the data.

The average probability_of_full_payment for Cluster 1 customers is: 0.8816
50% of the customers have probability_of_full_payment of 0.8819, 95% have of 0.
9054 and 99% have of 0.9162.
Interquartile range for probability_of_full_payment is 1.7646. IQR tells us the
 range of the middle half of the data.

The average current_balance (in 1000s) for Cluster 1 customers is: 5.5146
50% of the customers have current_balance (in 1000s) of 5.541, 95% have of 5.85
5 and 99% have of 5.8948.
Interquartile range for current_balance (in 1000s) is 11.0695. IQR tells us the
 range of the middle half of the data.

The average credit_limit (in 10000s) for Cluster 1 customers is: 3.2592
50% of the customers have credit_limit (in 10000s) of 3.258, 95% have of 3.506
and 99% have of 3.568.
Interquartile range for credit_limit (in 10000s) is 6.533. IQR tells us the ran
ge of the middle half of the data.

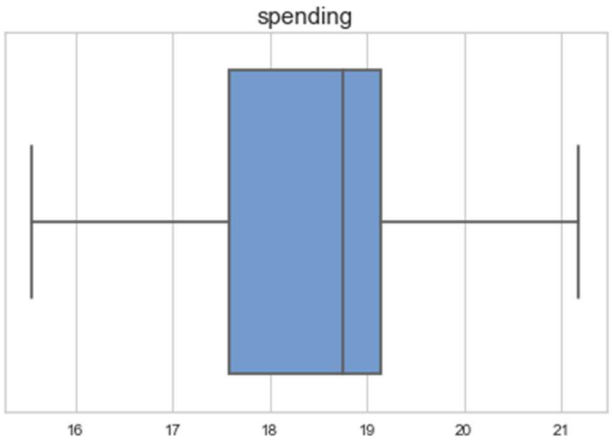
The average min_payment_amt (in 100s) for Cluster 1 customers is: 2.7073
50% of the customers have min_payment_amt (in 100s) of 2.64, 95% have of 4.6905
 and 99% have of 5.9206.
Interquartile range for min_payment_amt (in 100s) is 5.283. IQR tells us the ra
nge of the middle half of the data.

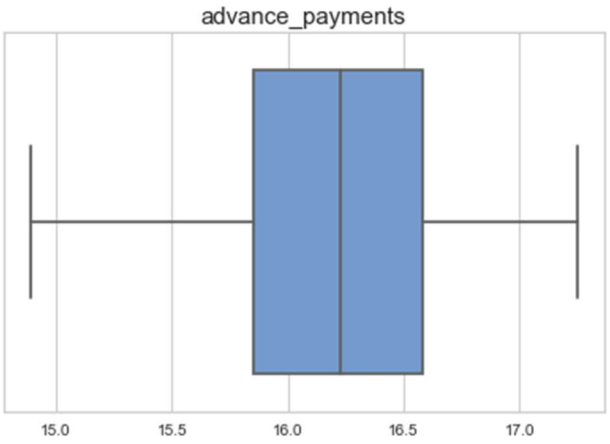
The average max_spent_in_single_shopping (in 1000s) for Cluster 1 customers is:
 5.1208
50% of the customers have max_spent_in_single_shopping (in 1000s) of 5.132, 95%
 have of 5.5305 and 99% have of 5.8202.
Interquartile range for max_spent_in_single_shopping (in 1000s) is 10.222. IQR
tells us the range of the middle half of the data.

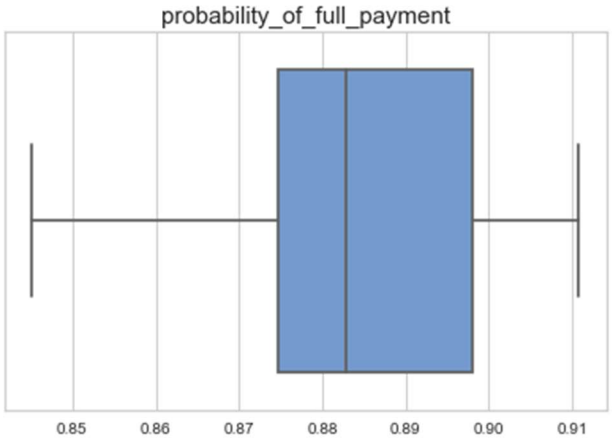
Cluster 2:

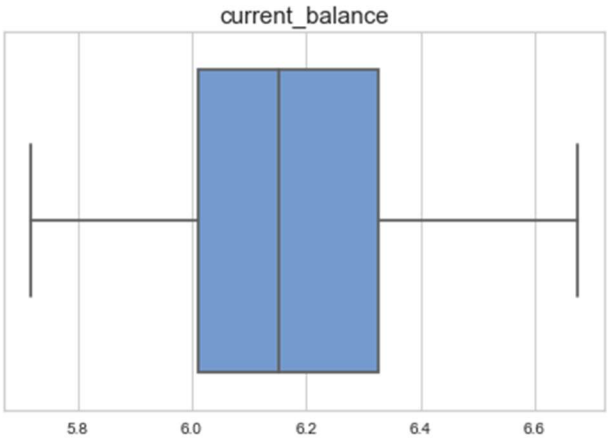
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Cluster
0	19.940	16.920	0.875	6.675	3.763	3.252	6.550	Cluster-2
2	18.950	16.420	0.883	6.248	3.755	3.368	6.148	Cluster-2
4	17.990	15.860	0.899	5.890	3.694	2.068	5.837	Cluster-2
8	18.170	16.260	0.864	6.271	3.512	2.853	6.273	Cluster-2
10	18.550	16.220	0.886	6.153	3.674	1.738	5.894	Cluster-2

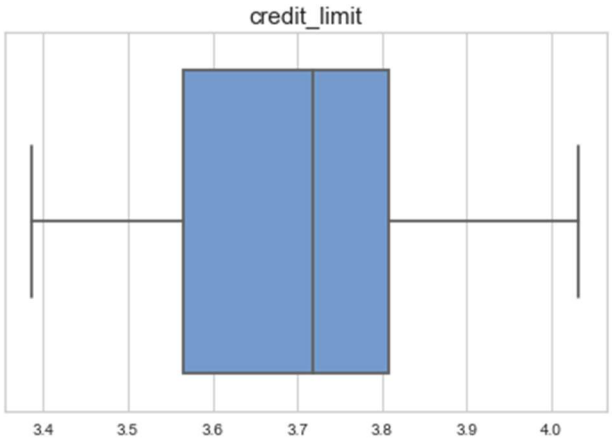
Table - 11 Cluster 2 (k-means)

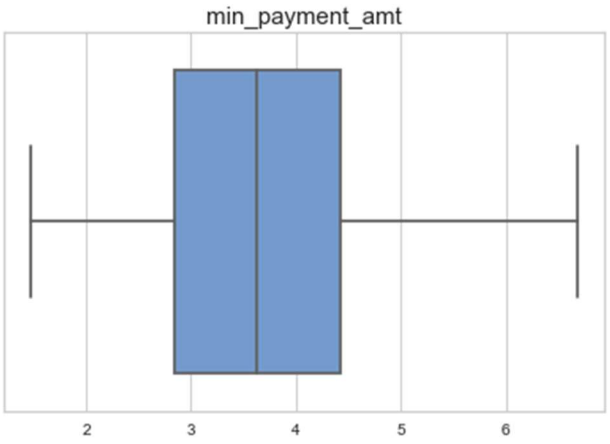

spending
16
17
18
19
20
21


advance_payments
15.0
15.5
16.0
16.5
17.0


probability_of_full_payment
0.85
0.86
0.87
0.88
0.89
0.90
0.91


current_balance
5.8
6.0
6.2
6.4
6.6


credit_limit
3.4
3.5
3.6
3.7
3.8
3.9
4.0


min_payment_amt
2
3
4
5
6

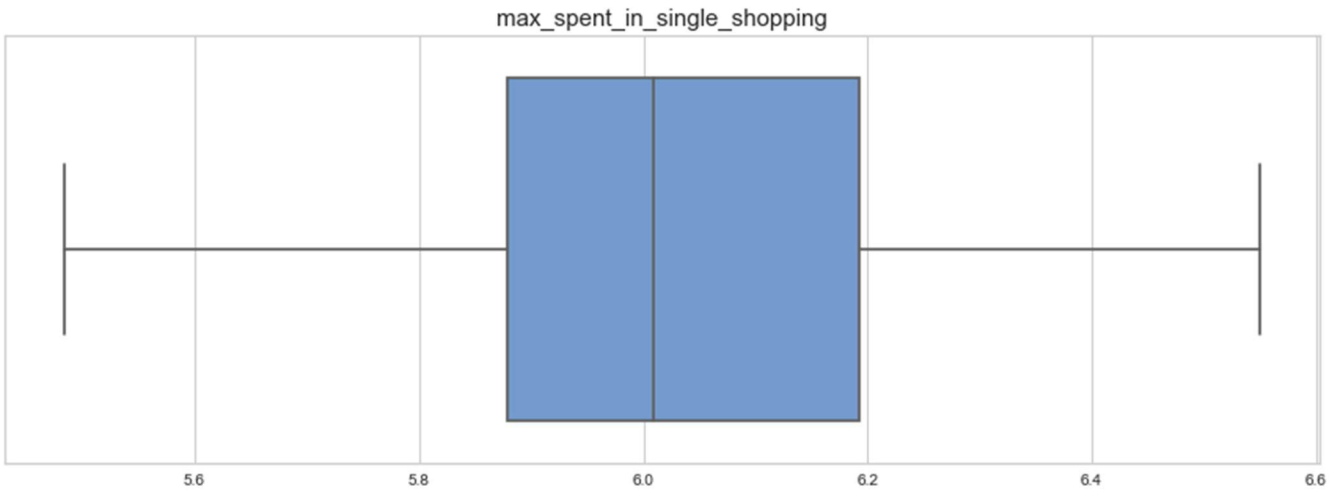


Figure - 14 Visualizing Cluster 2 (k-means)

The average spending (in 1000s) for Cluster 2 customers is: 11.8569
50% of the customers have spending (in 1000s) of 11.825, 95% have of 13.1285 and 99% have of 13.3258.
Interquartile range for spending (in 1000s) is 23.65. IQR tells us the range of the middle half of the data.

The average advance_payments (in 100s) for Cluster 2 customers is: 13.2478
50% of the customers have advance_payments (in 100s) of 13.25, 95% have of 13.8015 and 99% have of 13.9429.
Interquartile range for advance_payments (in 100s) is 26.475. IQR tells us the range of the middle half of the data.

The average probability_of_full_payment for Cluster 2 customers is: 0.8483
50% of the customers have probability_of_full_payment of 0.8486, 95% have of 0.8815 and 99% have of 0.8877.
Interquartile range for probability_of_full_payment is 1.6965. IQR tells us the range of the middle half of the data.

The average current_balance (in 1000s) for Cluster 2 customers is: 5.2317
50% of the customers have current_balance (in 1000s) of 5.225, 95% have of 5.4472 and 99% have of 5.5083.
Interquartile range for current_balance (in 1000s) is 10.4765. IQR tells us the range of the middle half of the data.

The average credit_limit (in 10000s) for Cluster 2 customers is: 2.8495
50% of the customers have credit_limit (in 10000s) of 2.8365, 95% have of 3.0734 and 99% have of 3.1567.
Interquartile range for credit_limit (in 10000s) is 5.7055. IQR tells us the range of the middle half of the data.

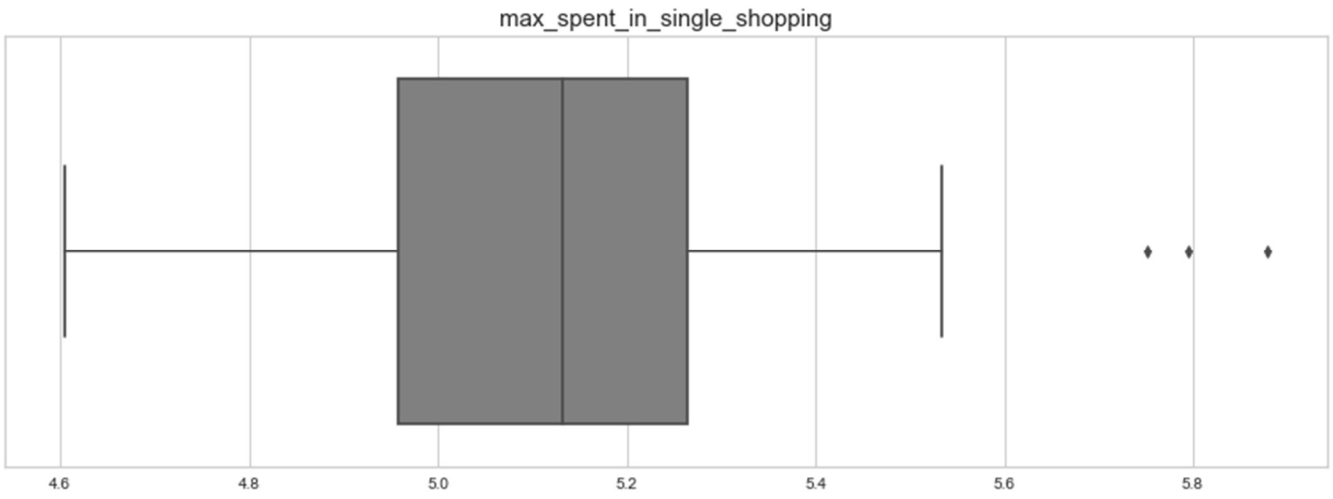
The average min_payment_amt (in 100s) for Cluster 2 customers is: 4.7424
50% of the customers have min_payment_amt (in 100s) of 4.799, 95% have of 7.0114 and 99% have of 8.3559.
Interquartile range for min_payment_amt (in 100s) is 9.496. IQR tells us the range of the middle half of the data.

```
The average max_spent_in_single_shopping (in 1000s) for Cluster 2 customers is:
 5.1017
50% of the customers have max_spent_in_single_shopping (in 1000s) of 5.089, 95%
 have of 5.3556 and 99% have of 5.4548.
Interquartile range for max_spent_in_single_shopping (in 1000s) is 10.2245. IQR
 tells us the range of the middle half of the data.
```

Cluster 3:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Cluster
1	15.990	14.890	0.906	5.363	3.582	3.336	5.144	Cluster-3
7	13.740	14.050	0.874	5.482	3.114	2.932	4.825	Cluster-3
11	14.090	14.410	0.853	5.717	3.186	3.920	5.299	Cluster-3
14	12.100	13.150	0.879	5.105	2.941	2.201	5.056	Cluster-3
16	16.140	14.990	0.903	5.658	3.562	1.355	5.175	Cluster-3

Table - 12 Cluster 3 (k-means)



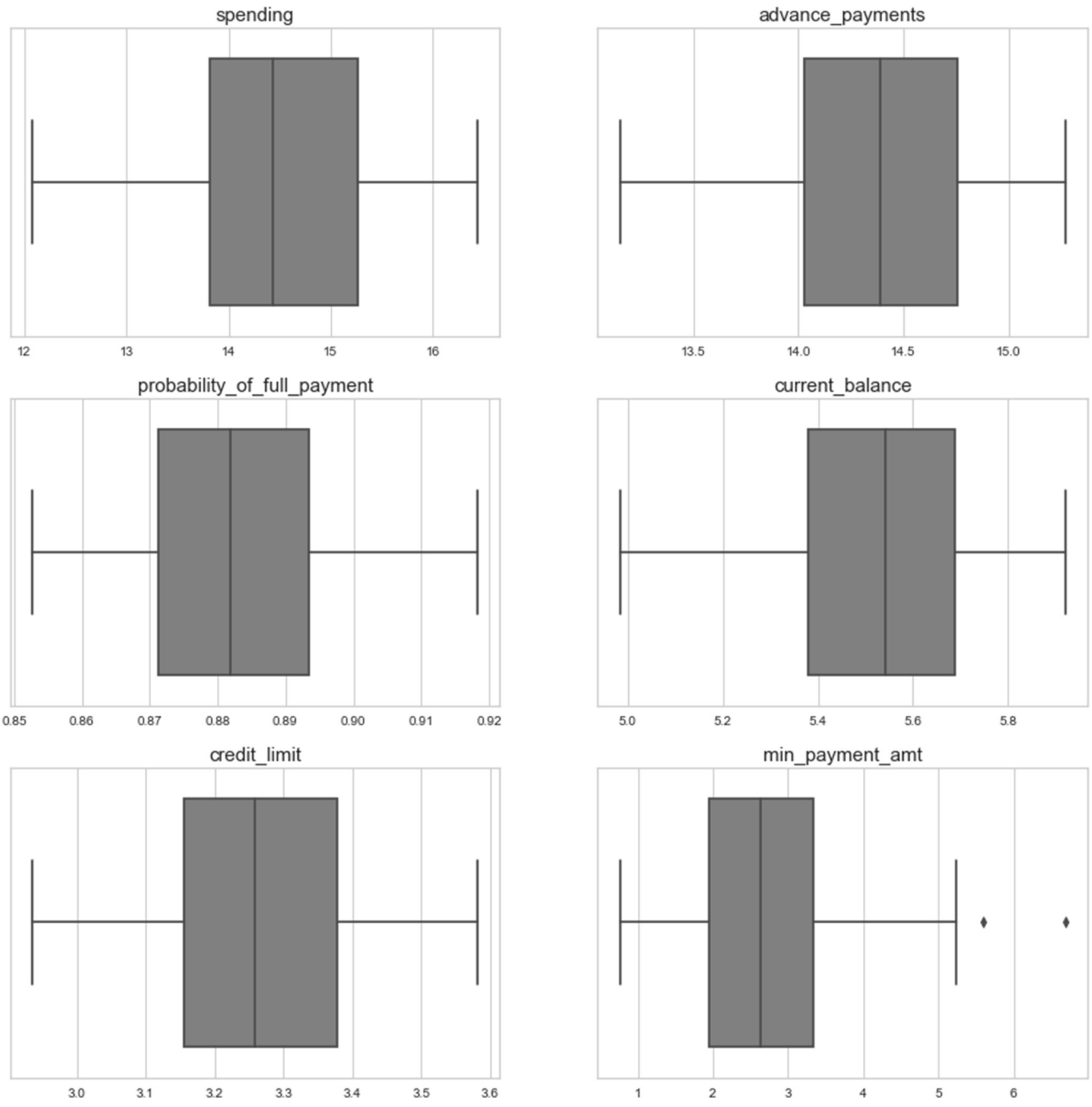


Figure - 15 Visualizing Cluster 3 (k-means)

```
The average spending (in 1000s) for Cluster 3 customers is: 18.4954
50% of the customers have spending (in 1000s) of 18.75, 95% have of 20.569 and
99% have of 21.0414.
Interquartile range for spending (in 1000s) is 36.735. IQR tells us the range o
f the middle half of the data.

The average advance_payments (in 100s) for Cluster 3 customers is: 16.2034
```

```
50% of the customers have advance_payments (in 100s) of 16.23, 95% have of 17.0
44 and 99% have of 17.2368.
Interquartile range for advance_payments (in 100s) is 32.435. IQR tells us the
range of the middle half of the data.

The average probability_of_full_payment for Cluster 3 customers is: 0.8842
50% of the customers have probability_of_full_payment of 0.8829, 95% have of 0.
9077 and 99% have of 0.909.
Interquartile range for probability_of_full_payment is 1.7727. IQR tells us the
 range of the middle half of the data.

The average current_balance (in 1000s) for Cluster 3 customers is: 6.1757
50% of the customers have current_balance (in 1000s) of 6.153, 95% have of 6.57
72 and 99% have of 6.6691.
Interquartile range for current_balance (in 1000s) is 12.3395. IQR tells us the
 range of the middle half of the data.

The average credit_limit (in 10000s) for Cluster 3 customers is: 3.6975
50% of the customers have credit_limit (in 10000s) of 3.719, 95% have of 3.9524
 and 99% have of 4.0323.
Interquartile range for credit_limit (in 10000s) is 7.3725. IQR tells us the ra
nge of the middle half of the data.

The average min_payment_amt (in 100s) for Cluster 3 customers is: 3.6324
50% of the customers have min_payment_amt (in 100s) of 3.619, 95% have of 5.705
6 and 99% have of 6.2325.
Interquartile range for min_payment_amt (in 100s) is 7.269. IQR tells us the ra
nge of the middle half of the data.

The average max_spent_in_single_shopping (in 1000s) for Cluster 3 customers is:
 6.0417
50% of the customers have max_spent_in_single_shopping (in 1000s) of 6.009, 95%
 have of 6.4504 and 99% have of 6.5157.
Interquartile range for max_spent_in_single_shopping (in 1000s) is 12.0715. IQR
 tells us the range of the middle half of the data.
```

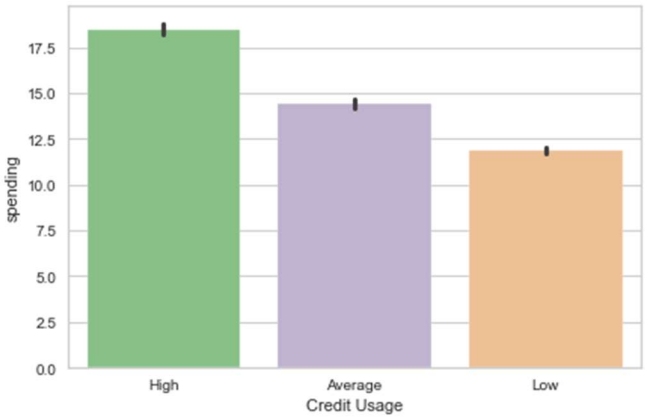
From the above inferences we can note that credit usage for customers in Cluster 1 is average for Cluster 2 is low and for Cluster 3 is high.

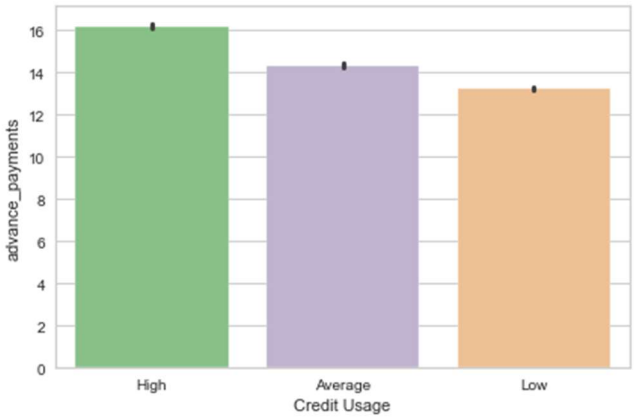
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Cluster	Credit Usage
0	19.940	16.920	0.875	6.675	3.763	3.252	6.550	Cluster-2	Low
1	15.990	14.890	0.906	5.363	3.582	3.336	5.144	Cluster-3	Average
2	18.950	16.420	0.883	6.248	3.755	3.368	6.148	Cluster-2	Low
3	10.830	12.960	0.810	5.278	2.641	5.182	5.185	Cluster-1	High
4	17.990	15.860	0.899	5.890	3.694	2.068	5.837	Cluster-2	Low

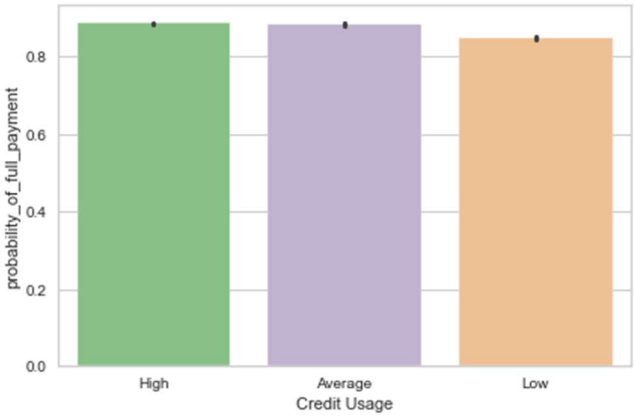
Table - 13 Cluster Profile as per K-means

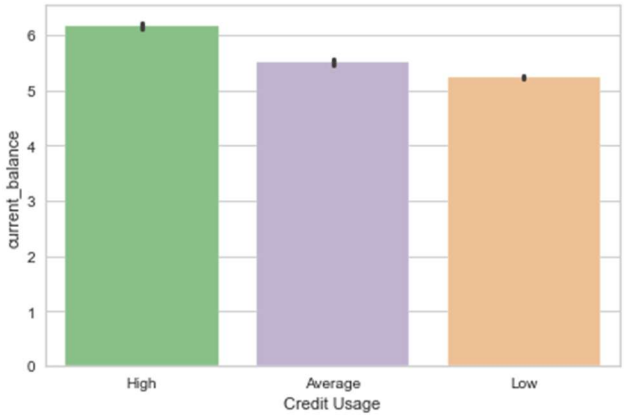
1.5 Describe cluster profiles for the clusters defined. Recommend different promotional strategies for different clusters.

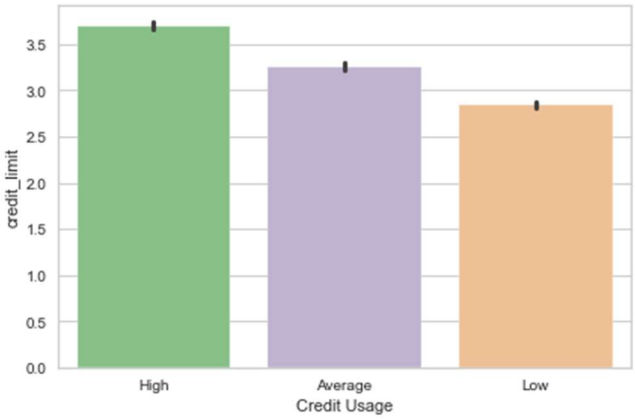
Cluster profile:

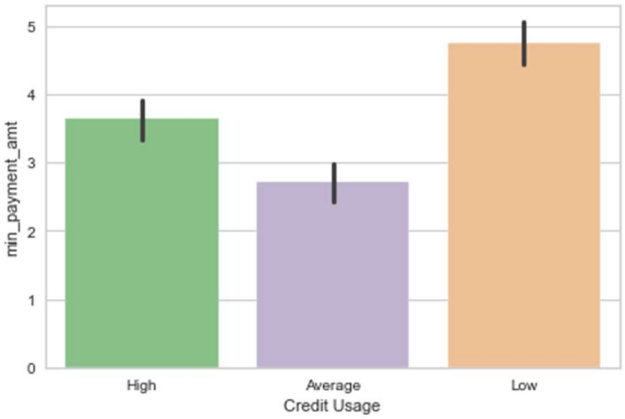


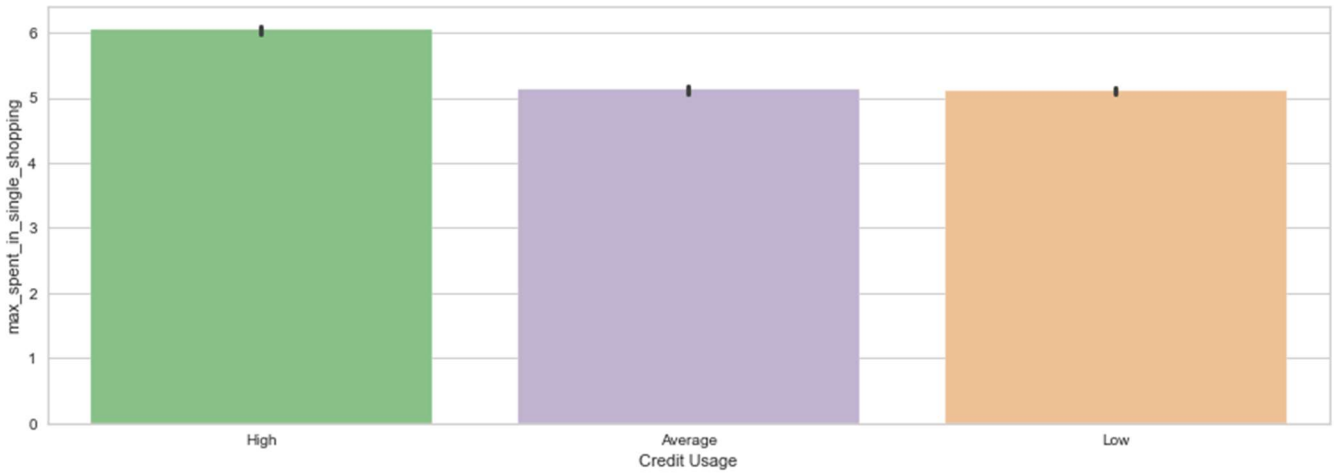












Customer Profile based on credit usage:

High: *Customers profiled as high credit usage have an average credit limit of 36,000 and average total spends of approx. 18,000. The maximum spend is around 6000 approx. on single shopping. The average probability that the customer ranking as high in credit usage will make a full payment is 0.8842 (88.42% likely to make full payment). ~ 31% of total customers are in this profile.*

Average: *Customers profiled as average credit usage have an average credit limit of 32,000 and average total spends of approx. 14,000. The maximum spend is around 5000 approx. on single shopping. The average probability that the customer ranking as average in credit usage will make a full payment is 0.8816 (88.16% likely to make full payment). ~ 34% of total customers are in this profile.*

Low: *Customers profiled as high credit usage have an average credit limit of 28,000 and average total spends of approx. 11,000. The maximum spend is around 5000 approx. on single shopping. The average probability that the customer ranking as low in credit usage will make a full payment is 0.8483 (84.83% likely to make full payment). ~ 33% of total customers are in this profile.*

Promotional Strategies

- *For the customers ranked in low credit use, the bank can provide discount or cash back promotions for each time a credit card bill is paid. This will increase the probability of the full payment and increase in loyalty with rewards in return. This strategy can be applicable for rest of the groups as well.*
- *The third-party marketing collaboration one of the examples where brand related credit cards such as Amazon pay cashback or Uber points can be more attractive to the customers for both with average and low credit use. Customers are likely to be loyal to a credit card provider by rewards and incentives.*

- *Particular credit card product for each user needs – for example platinum cards with high rewards for customers segmented in high credit use and promotional offers. EMI credit card or for the customers segmented in low credit usage which will promote them to purchase more and will likely to make full payment over the set period of time or Zero interest credit cards let cardholders skip paying an annual percentage rate on purchases, balance transfers or both for a set period of time. Promotional cards for the average users which provides discounts every month like 10% off on flights/food/gas or 0% intrest on annual fees.*
- *Personalized marketing emails/notifications for each user based on their recent promotional or offers availed. Bank can offer similar or same type of offers to the customers for all the segments.*
- *Attractive offers for loan, EMI or offering rewards once a while for a minimum amount spent over a period of time.*

Problem 2: CART-RF-ANN

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

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

Reading the data:

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

Data imported successfully!

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	C2B	Airlines	No	0.700	Online	7	2.510	Customised Plan	ASIA
1	36	EPX	Travel Agency	No	0.000	Online	34	20.000	Customised Plan	ASIA
2	39	CWT	Travel Agency	No	5.940	Online	3	9.900	Customised Plan	Americas
3	36	EPX	Travel Agency	No	0.000	Online	4	26.000	Cancellation Plan	ASIA
4	33	JZI	Airlines	No	6.300	Online	53	18.000	Bronze Plan	ASIA

Table - 14 Insurance Dataset

Performing Initial Steps

Structure of the dataset:

```
---------------------------------------------------------------------------------------

The dataset has 3000 rows and 10 columns.

Total elements in this dataset are 30000
```

Missing Values Check

```
---------------------------------------------------------------------------------------
There are no missing values in the dataset
```

Info of the dataset

```
---------------------------------------------------------------------------------------
<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
None
```

Duplicate Values Check

```
---------------------------------------------------------------------------------------
Number of duplicate rows = 139
```

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
63	30	C2B	Airlines	Yes	15.000	Online	27	60.000	Bronze Plan	ASIA
329	36	EPX	Travel Agency	No	0.000	Online	5	20.000	Customised Plan	ASIA
407	36	EPX	Travel Agency	No	0.000	Online	11	19.000	Cancellation Plan	ASIA
411	35	EPX	Travel Agency	No	0.000	Online	2	20.000	Customised Plan	ASIA
422	36	EPX	Travel Agency	No	0.000	Online	5	20.000	Customised Plan	ASIA
473	36	EPX	Travel Agency	No	0.000	Online	26	24.000	Customised Plan	ASIA
524	36	EPX	Travel Agency	No	0.000	Online	3	10.000	Cancellation Plan	ASIA
540	33	C2B	Airlines	Yes	54.000	Online	365	216.000	Silver Plan	ASIA
567	36	EPX	Travel Agency	No	0.000	Online	19	20.000	Customised Plan	ASIA
569	36	EPX	Travel Agency	No	0.000	Online	14	20.000	Customised Plan	ASIA

Table - 15 Duplicate records

Inferences

The dataset has 3000 rows and 10 columns and total elements in this dataset are 30000. There are no missing values however there are 139 duplicate values. Based on our problem statement I've decided to keep the duplicate values as it might give us some insight on similarity of few cases and its frequency. The data uses 234.5+ KB of the total memory. Age, Commision, Duration and Sales are the numerical variables in the dataset and AgencyCode, Type, Channel, Sales, Product Name and Destination are the categorical variables with Claimed being the target variable for our business problem.

Performing Univariate Analysis

This dataset has 4 numerical values and 6 categorical values! Perform statistical analysis for both!

	count	mean	std	min	25%	50%	75%	max
Age	3000.000	38.091	10.464	8.000	32.000	36.000	42.000	84.000
Commision	3000.000	14.529	25.481	0.000	0.000	4.630	17.235	210.210
Duration	3000.000	70.001	134.053	-1.000	11.000	26.500	63.000	4580.000
Sales	3000.000	60.250	70.734	0.000	20.000	33.000	69.000	539.000

Table - 16 Statistical Summary for Insurance dataset (numerical)

```
Univariate Analysis for column: Age
--------------------------------------------------------------------------------
Statistical Inferences
------------------------------------------------------------
count    3000.000
mean       38.091
```

```
std       10.464
min        8.000
25%       32.000
50%       36.000
75%       42.000
max       84.000
Name: Age, dtype: float64
```

Shapiro-Wilk test for normality

```
-----------------------------------------------------------
p-value for Age column is 0.0,
Hence, we reject the null hypothesis that the data is normally distributed
```

Detecting outliers using z-score

```
-----------------------------------------------------------
Age variable has outliers
```

Univariate Analysis for column: Commision

```
-------------------------------------------------------------------------------------------
```

Statistical Inferences

```
-----------------------------------------------------------
count   3000.000
mean      14.529
std       25.481
min        0.000
25%        0.000
50%        4.630
75%       17.235
max      210.210
Name: Commision, dtype: float64
```

Shapiro-Wilk test for normality

```
-----------------------------------------------------------
pvalue for Commision column is 0.0,
Hence, we reject the null hypothesis that the data is normally distributed
```

Detecting outliers using z-score

```
-----------------------------------------------------------
Commision variable has outliers
```

Univariate Analysis for column: Duration

```
-------------------------------------------------------------------------------------------
```

Statistical Inferences

```
-----------------------------------------------------------
count   3000.000
mean      70.001
std      134.053
min       -1.000
25%       11.000
50%       26.500
75%       63.000
max     4580.000
Name: Duration, dtype: float64
```

Shapiro-Wilk test for normality

```
------------------------------------------------------------
pvalue for Duration column is 0.0,
Hence, we reject the null hypothesis that the data is normally distributed


Detecting outliers using z-score
------------------------------------------------------------
Duration variable has outliers

Univariate Analysis for column: Sales
------------------------------------------------------------------------------------------
Statistical Inferences
------------------------------------------------------------
count   3000.000
mean      60.250
std       70.734
min        0.000
25%       20.000
50%       33.000
75%       69.000
max      539.000
Name: Sales, dtype: float64

Shapiro-Wilk test for normality
------------------------------------------------------------
pvalue for Sales column is 0.0,
Hence, we reject the null hypothesis that the data is normally distributed


Detecting outliers using z-score
------------------------------------------------------------
Sales variable has outliers
```

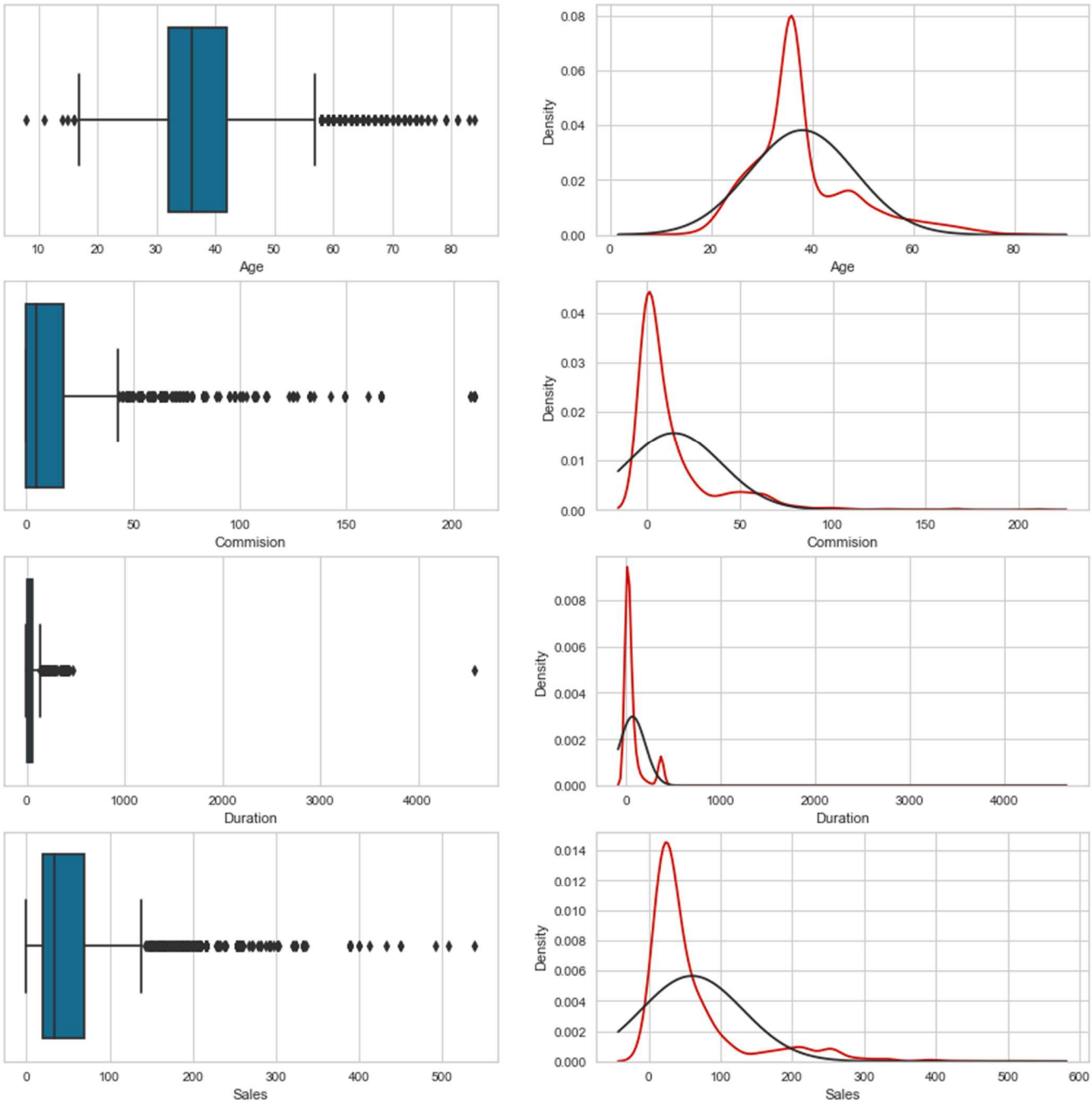


Figure - 16 Univariate Analysis (Numerical)

Based on the descriptive statistics on numerical variables we see that the min Age of the person insured is 8 years and max age is 84 years. Highest commission earned is 210.210 and highest sales is 539.

*Based on **Shapiro-Wilks Test** the columns have outliers and as per the **z-score check** the variables in the data are not normally distributed.*

Univariate Analysis for categorical variables:

	Agency_Code	Type	Claimed	Channel	Product Name	Destination
count	3000	3000	3000	3000	3000	3000
unique	4	2	2	2	5	3
top	EPX	Travel Agency	No	Online	Customised Plan	ASIA
freq	1365	1837	2076	2954	1136	2465

Table - 17 Statistical Summary - Insurance dataset (Categorical)

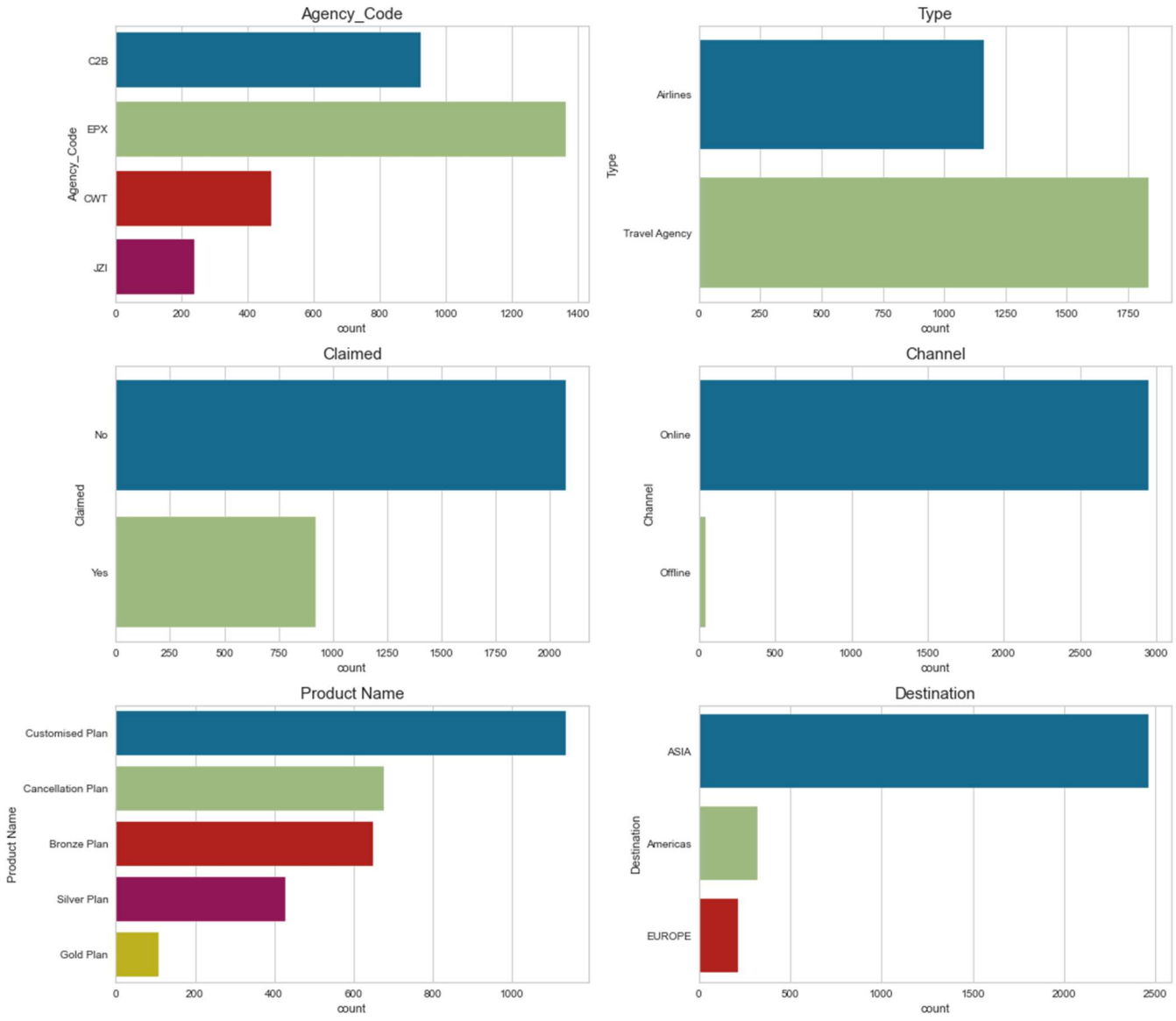
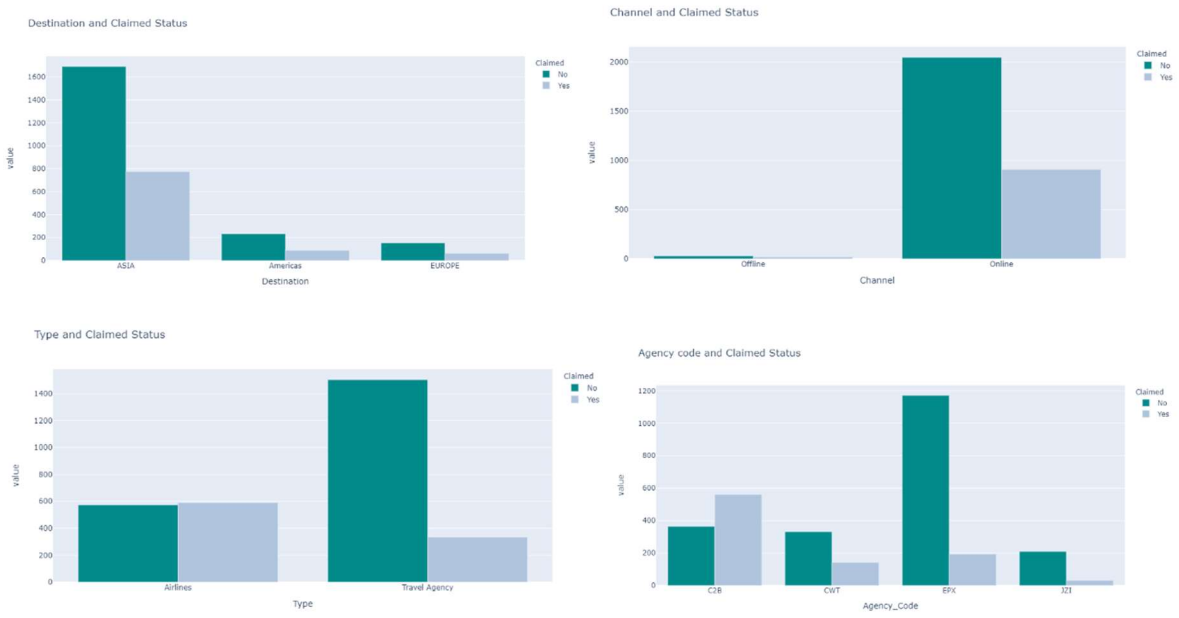


Figure - 17 Univariate Analysis – Categorical

Inferences

- *Out of 4 unique values in `Agency_Code` the top agency code is **EPX** with frequency of 1365.*
- *Out of 2 unique values in `Type` the type with highest frequency is **Travel Agency** with frequency of 1837.*
- *Out of 2 unique values in `Claimed` the highest claimed is **No** with frequency of 2076.*
- *Out of 2 unique values in `Channel` the channel with highest frequency is **Online** with frequency of 2954.*
- *Out of 5 unique values in `Product Name` the highest frequented product is **Customised Plan** with frequency of 1136.*
- *Out of 3 unique values in `Destination` the one highest frequented is **ASIA** with frequency of 2465.*

Performing Bivariate Analysis:



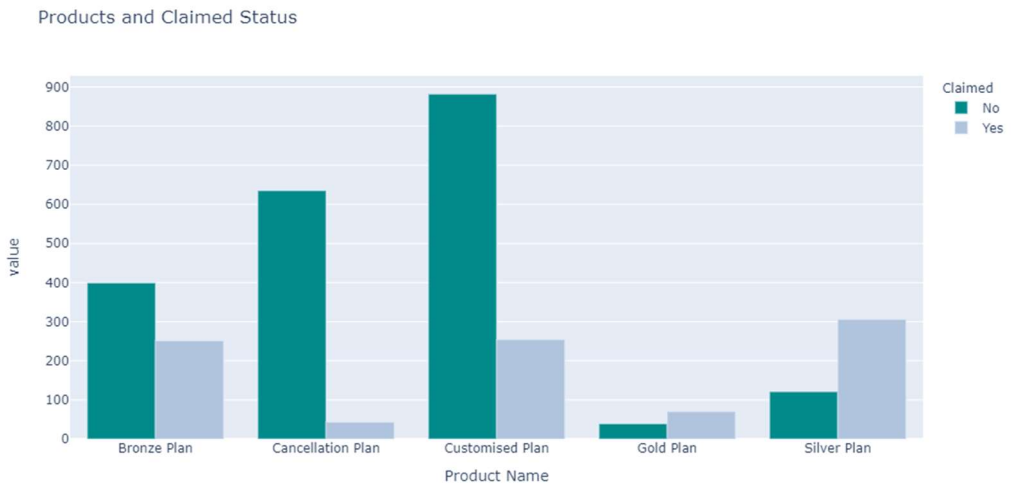


Figure - 18 Bivariate Analysis

Inferences

- ***Customised Plan*** *has the highest value with no claims.* ***Gold and Silver Plan*** *have more claims than rest of the plans.*
- ***EPX agency code*** *has highest number of records with no claims.* ***CWT*** *and* ***JZI*** *have more claims than rest of the agencies.*
- ***Airlines seems*** *to have more or less a balanced claim status compared to Travel Agencies.* ***Travel Agencies*** *have high records of no claims.*
- ***Online channel*** *is more dominant than the offline channel. Online channels have high records of no claims as well.*
- ***Asia*** *is the most popular destination compared to Americas and Europe. All the three destinations have no claims more than yes.*

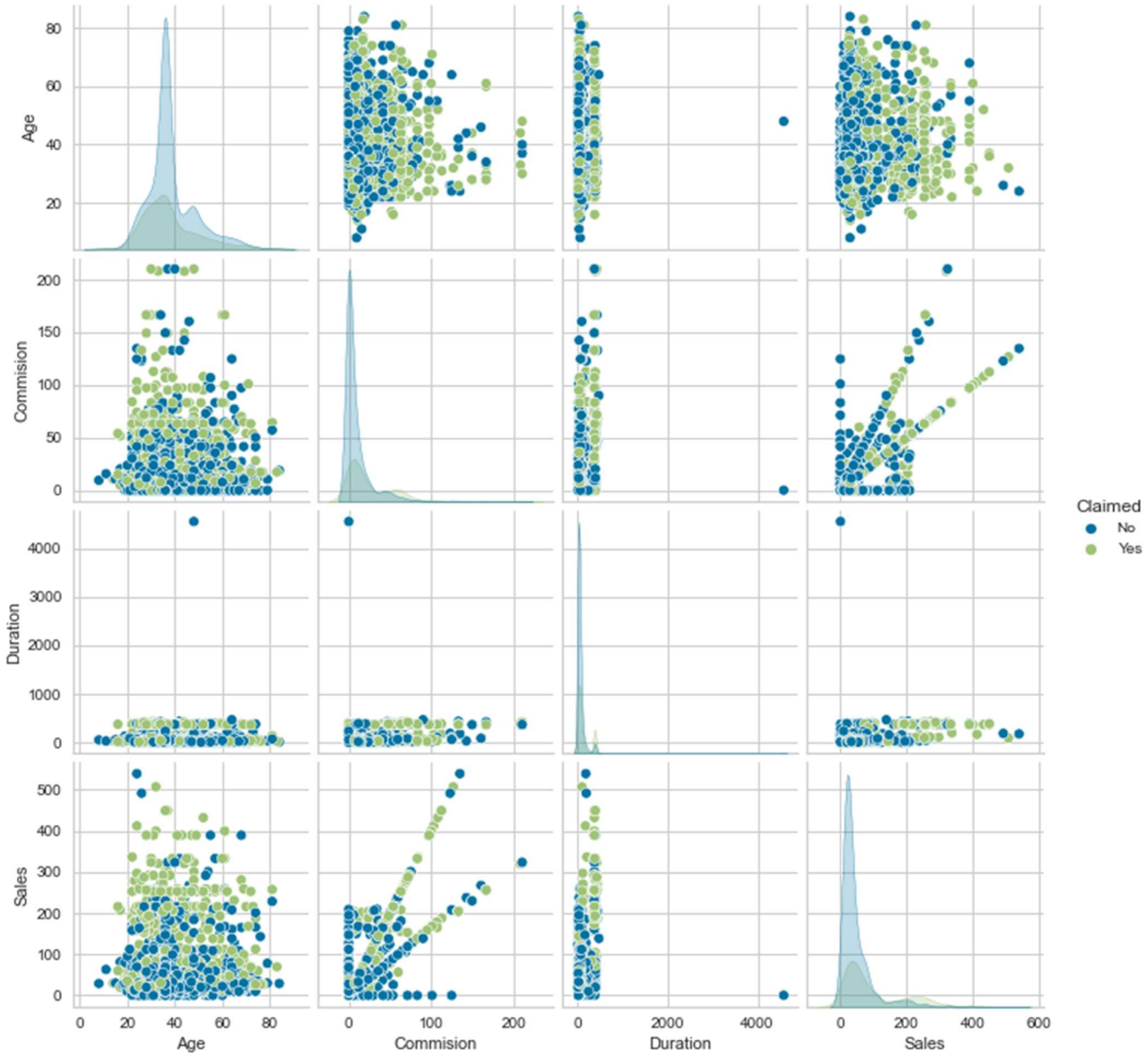
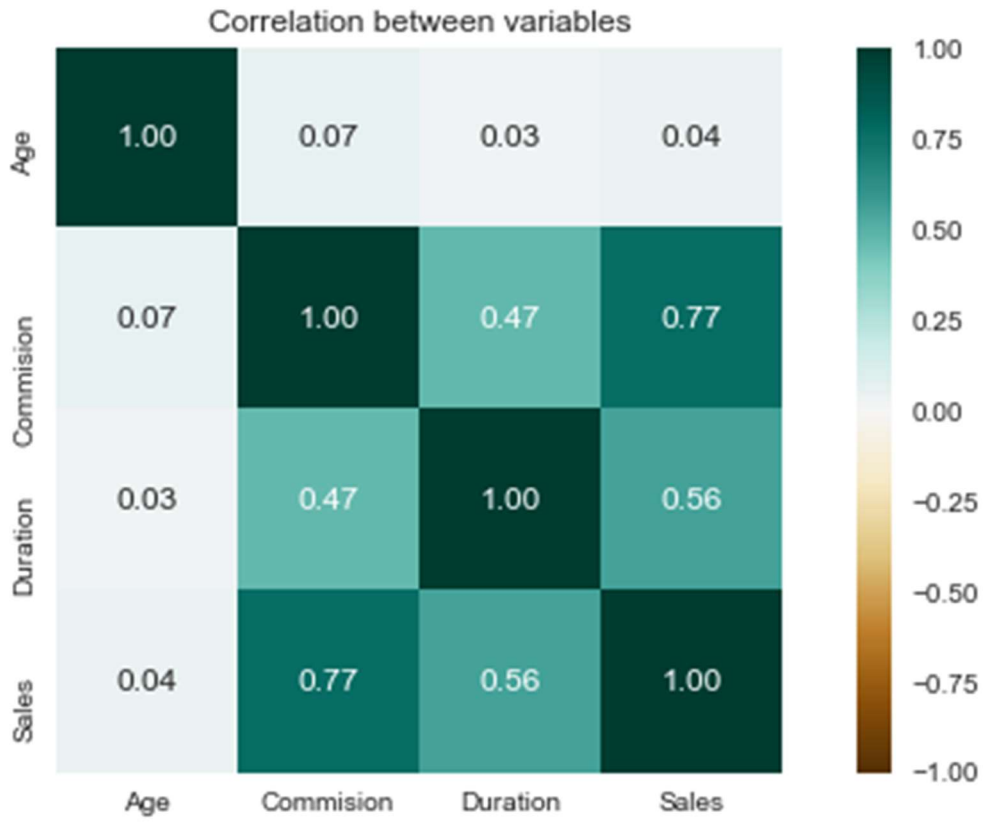


Figure - 19 Pair plot

Inferences

Only visible correlation between the data can be seen in `Commision` *and* `Sales` *column.*

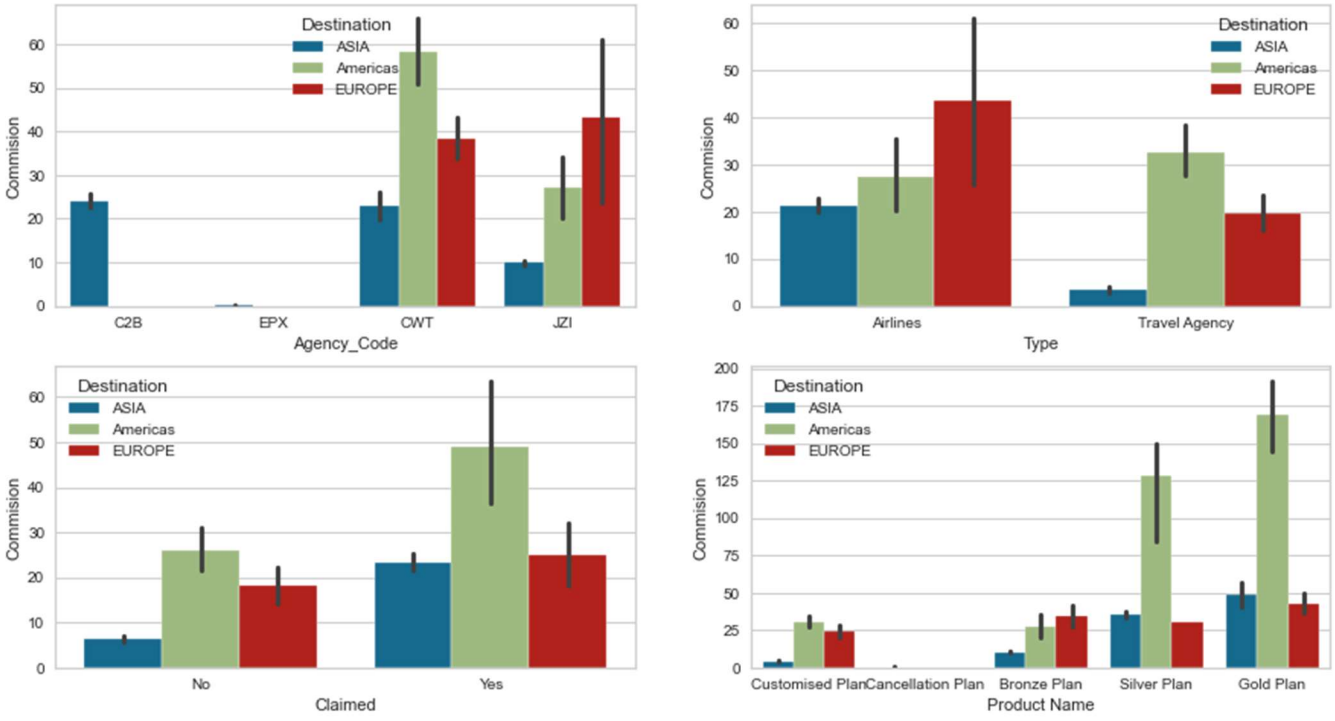
Performing Multivariate Analysis:



Inferences

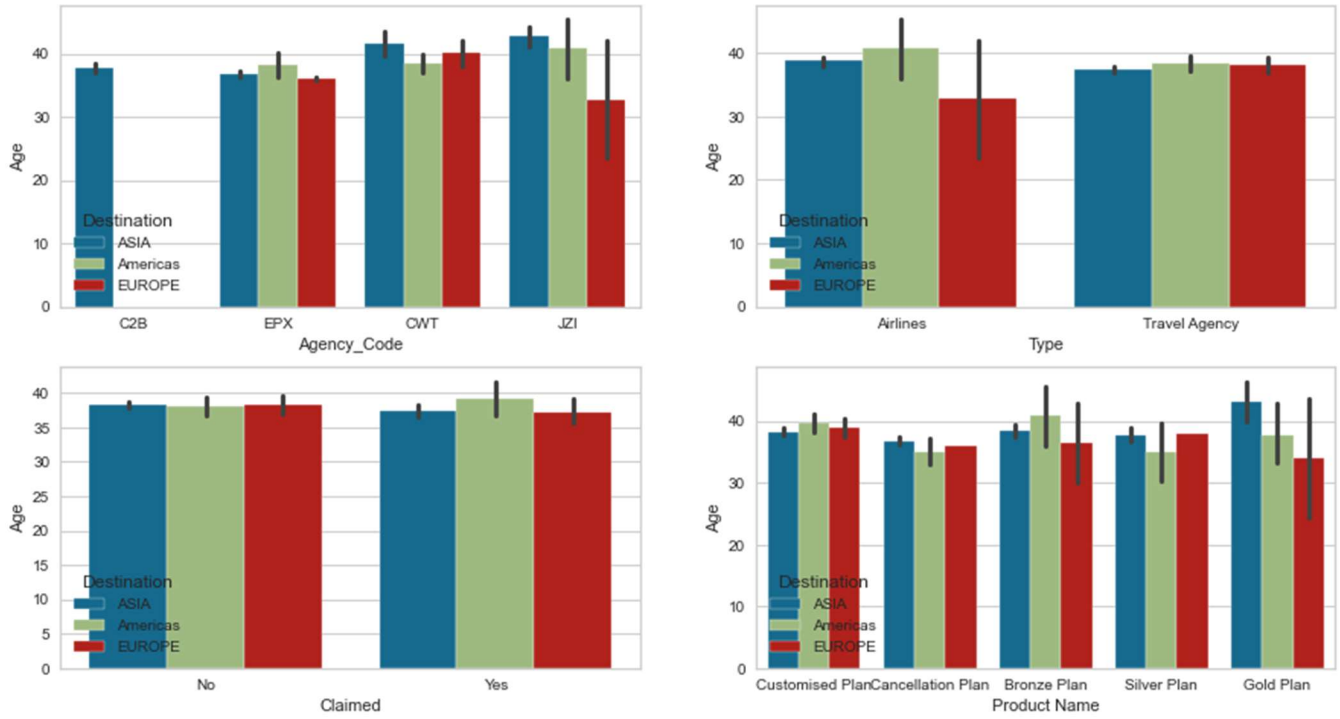
- Variable Age has no correlation with any of the other numerical variables.
- Sales, Commision and Duration are all positively correlated to each other.

Figure - 20 Heatmap (numerical vs numerical)



- Highest commission is received is from CWT agency code with America destination.
- America destination has highest commissions with yes claims followed by Europe.

- Airlines commissions are high in the type of tour insurance firms. Though the travel agencies rank high with the Americas tour destination.
- Silver Plan and the Gold Plan are having high commissions compared to other plans.



- *Asia destination and CWT and JZI agency have most of the insured in age group of 40 above.*
- *All the insured are above age of 30. Most claims are from America destination and ages above 35.*
- *Age group of 30-35 prefer cancelation plan. Insured in age group of 40 prefer Gold Plan.*

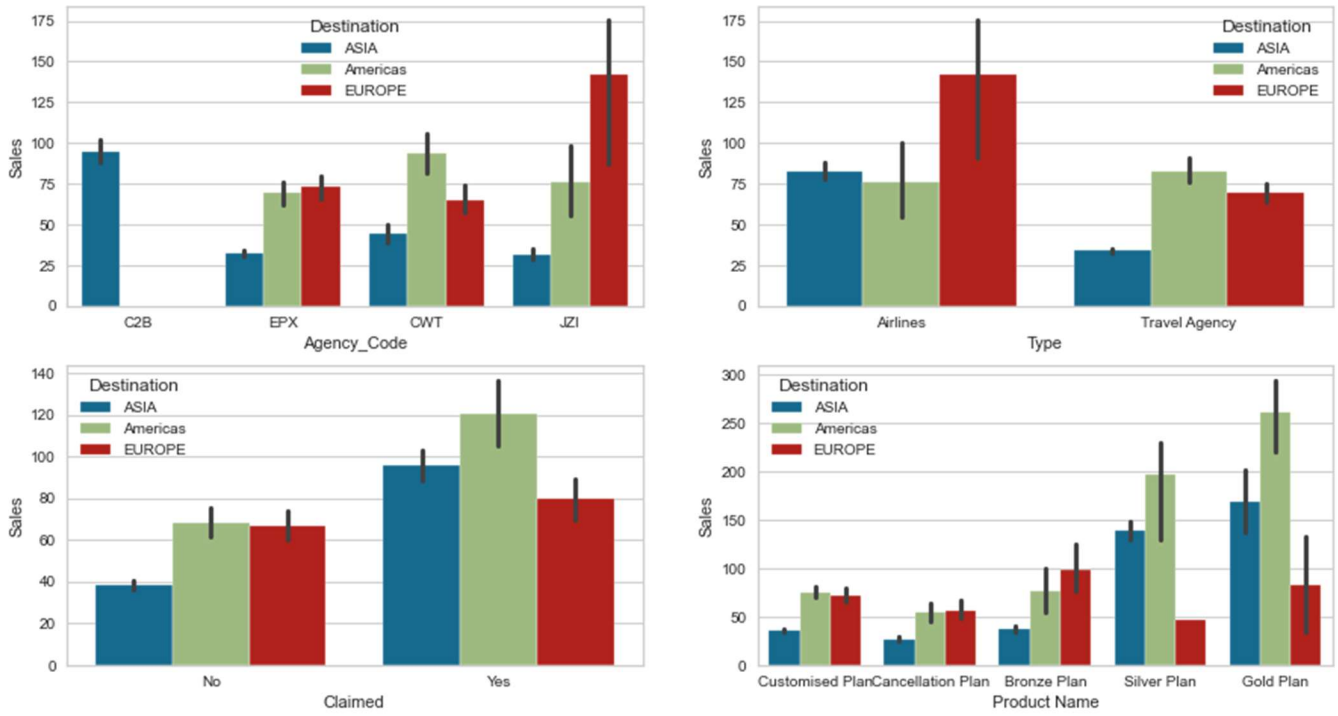


Figure - 21 Numerical vs Categorical

- Overall sales are highest in JZI agency code with Europe destination. Sales with Europe destination is high in airlines tourist firms.
- Most of the claims are as we have seen is high in America destination though sales are also high.
- Highest sales generating plans are Gold and Silver.

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

Converting features in insurance columns to binary/multiclass

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


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


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

```
[0 1]


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


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


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

Scaling of data

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	0.947	-1.314	-1.257	-0.667	-0.543	0.125	-0.470	-0.816	0.269	-0.435
1	-0.200	0.698	0.796	-0.667	-0.570	0.125	-0.269	-0.569	0.269	-0.435
2	0.087	-0.308	0.796	-0.667	-0.337	0.125	-0.500	-0.712	0.269	1.304
3	-0.200	0.698	0.796	-0.667	-0.570	0.125	-0.492	-0.484	-0.526	-0.435
4	-0.487	1.704	-1.257	-0.667	-0.323	0.125	-0.127	-0.597	-1.320	-0.435

Table - 18 Scaled data – Insurance

Splitting data into Train/Test

Train size used is 70% of the total data and test size is 30%. Splitting ratio is an important aspect of evaluating data as it can minimize the effects of data discrepancies and better understand the characteristics of the model. Random state is used to ensure that the splits that is generated are reproducible.

```
Dimensions for X_train = (2100, 9)
Dimensions for X_test = (900, 9)
Dimensions for train_labels = (2100,)
Dimensions for test_labels = (900,)
Total Observation = 3000
```

Decision Tree Model

```
Best parameters: {'max_depth': 8, 'min_samples_leaf': 15, 'min_samples_split':
```

75}

The maximum depth of the tree is 8, The minimum number of samples required to b e at a leaf node is 15 and the minimum number of samples required to split an i nternal node is 75. The parameters are selected using grid search from sklearn model selection with cross validation up to 3 folds. Random state is used to en sure that the splits that is generated are reproducible.

Cross-validation, sometimes called rotation estimation or out-of-sample testing , is any of various similar model validation techniques for assessing how the r esults of a statistical analysis will generalize to an independent data set.

Random Forest

Best parameters: {'max_depth': 7,'max_features': 8, 'min_samples_leaf': 20, 'min _samples_split': 75, 'n_estimators': 101}

The maximum depth of the tree is 7, the number of features to consider when loo king for the best split is 8, The minimum number of samples required to be at a leaf node is 20 and the minimum number of samples required to split an interna l node is 75 and number of trees in the forest are 101. The parameters are sele cted using grid search from sklearn model selection with cross validation up to 3 folds. Random state is used to ensure that the splits that is generated are reproducible.

Artificial Neural Network (ANNs)

Best parameters: {'activation': logistic, 'hidden_layer_sizes': (100, 100, 100), 'm ax_iter': 10000, 'solver': 'adam', 'tol': 0.01}

The activation used is 'logistic', the logistic sigmoid function, returns f(x) = 1 / (1 + exp(-x)). Solver used is 'adam' which refers to a stochastic gradient-based optimizer proposed by Kingma, Diederik, and Jimmy Ba.
The ith element represents the number of neurons in the ith hidden layer which is (100,100,100). Maximum number of iterations: the solver iterates until convergence (determined by 'tol') or this number of iterations which in this model is 10000. Tolerance for the optimization. 'Tol' is 0.01.
When the loss or score is not improving by at least tol for n_iter_no_change consecutive iterations, convergence is considered to be reached and training stops.
The parameters are selected using grid search from sklearn model selection with cross validation up to 3 folds. Random state is used to ensure that the splits that is generated are reproducible.

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.

Performance Check

Decision Tree

Classification Reports

Performance Metrics: Train

	precision	recall	f1-score	support
0	0.83	0.88	0.85	1464
1	0.68	0.57	0.62	636
accuracy			0.79	2100
macro avg	0.75	0.73	0.74	2100
weighted avg	0.78	0.79	0.78	2100

Confusion Matrix and ROC Curve– Train:

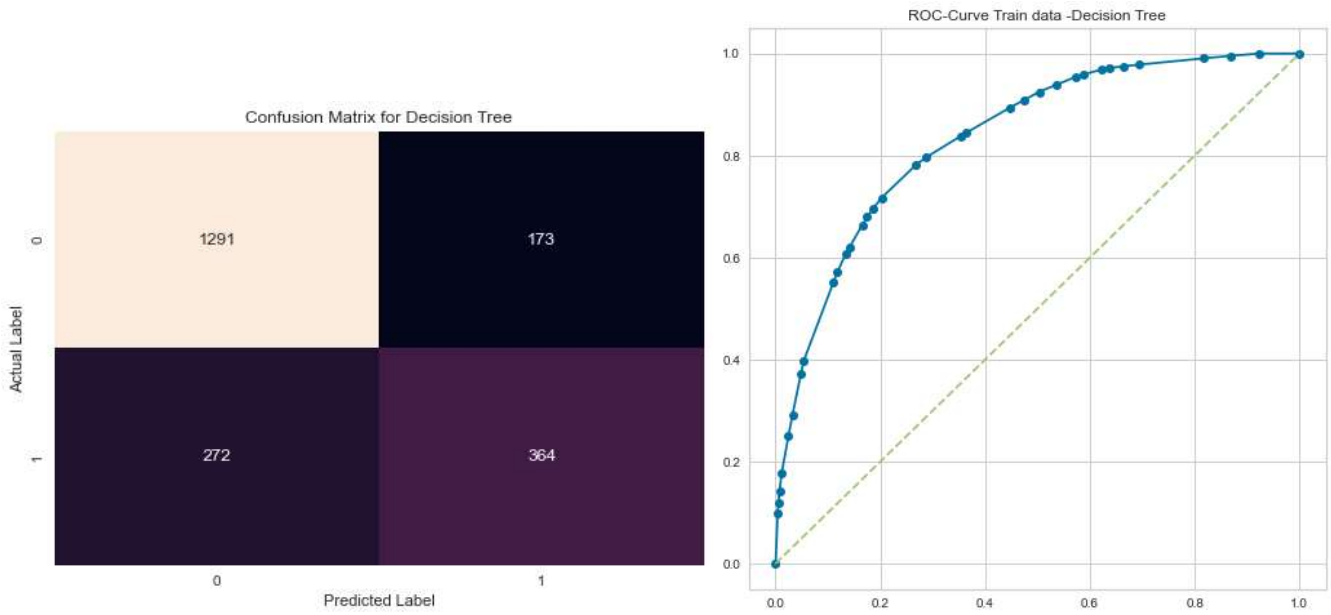


Figure - 22 Confusion Matrix and ROC Curve (CART) - Train

AUC for train dataset (Decision Tree): 0.837

Performance Metrics: Test

	precision	recall	f1-score	support
0	0.80	0.90	0.85	612
1	0.71	0.53	0.61	288
accuracy			0.78	900
macro avg	0.75	0.72	0.73	900
weighted avg	0.77	0.78	0.77	900

Confusion Matrix - Test

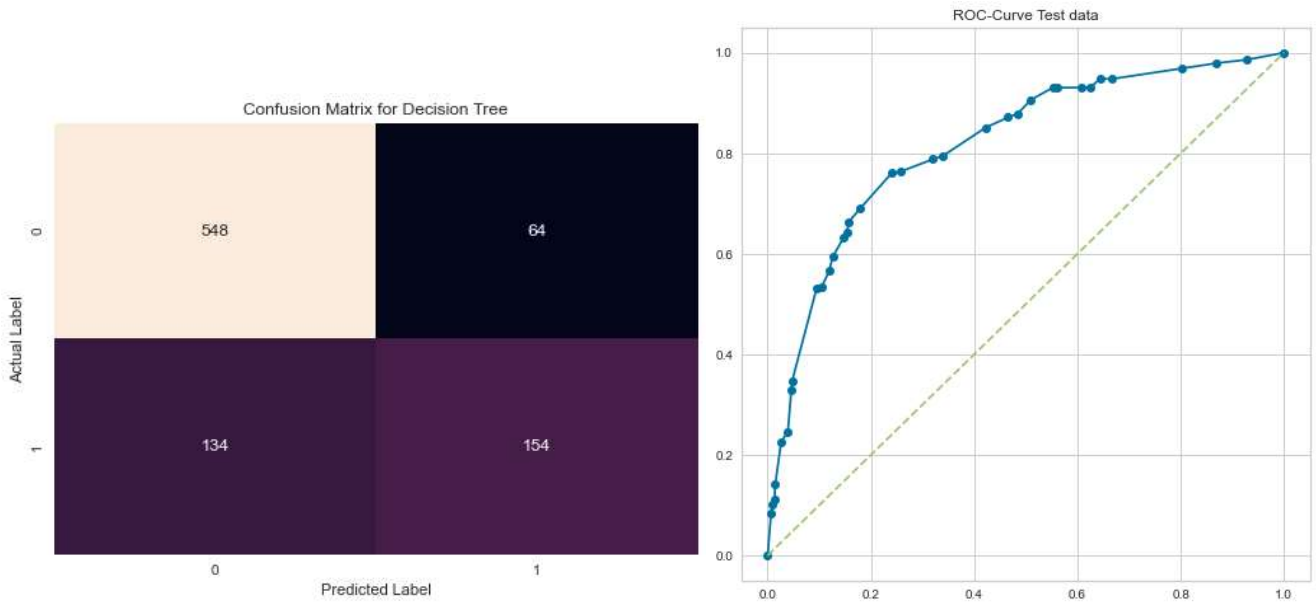


Figure - 23 Confusion Matrix and ROC Curve (CART) - Test

AUC for test dataset (Decision Tree): 0.817

```
Accuracy for train dataset =  0.7880952380952381
Recall for train dataset =  0.5723270440251572
Precision for train dataset =  0.6778398510242085
AUC_ROC_score for train dataset =  0.7270788225590267
f1_score for train dataset =  0.620630861040068

Accuracy for test dataset =  0.78
Recall for test dataset =  0.5347222222222222
Precision for test dataset =  0.7064220183486238
AUC_ROC_score for test dataset =  0.7150735294117647
f1_score for test dataset =  0.608695652173913
```

Accuracy of the model for train and test dataset are 0.79 and 0.78 respectively. Recall for train and test data set is 0.57 and 0.53 respectively which represents the model's ability to classify all the positive samples with the precision of 0.68 in train dataset and 0.71 in test dataset. Based o n the AUC_ROC score for train and test dataset which is 0.73 and 0.72 respectively, the model

has a good sense to separate and distinguish between the classes. F1 score (harmonic mean of precision and recall) for train dataset is 0.62 and for test dataset is 0.61.

Random Forest

Classification Reports

Performance Metrics: Train

	precision	recall	f1-score	support
0	0.84	0.88	0.86	1464
1	0.69	0.60	0.64	636
accuracy			0.80	2100
macro avg	0.76	0.74	0.75	2100
weighted avg	0.79	0.80	0.79	2100

Confusion Matrix and ROC – Train (Random Forest)

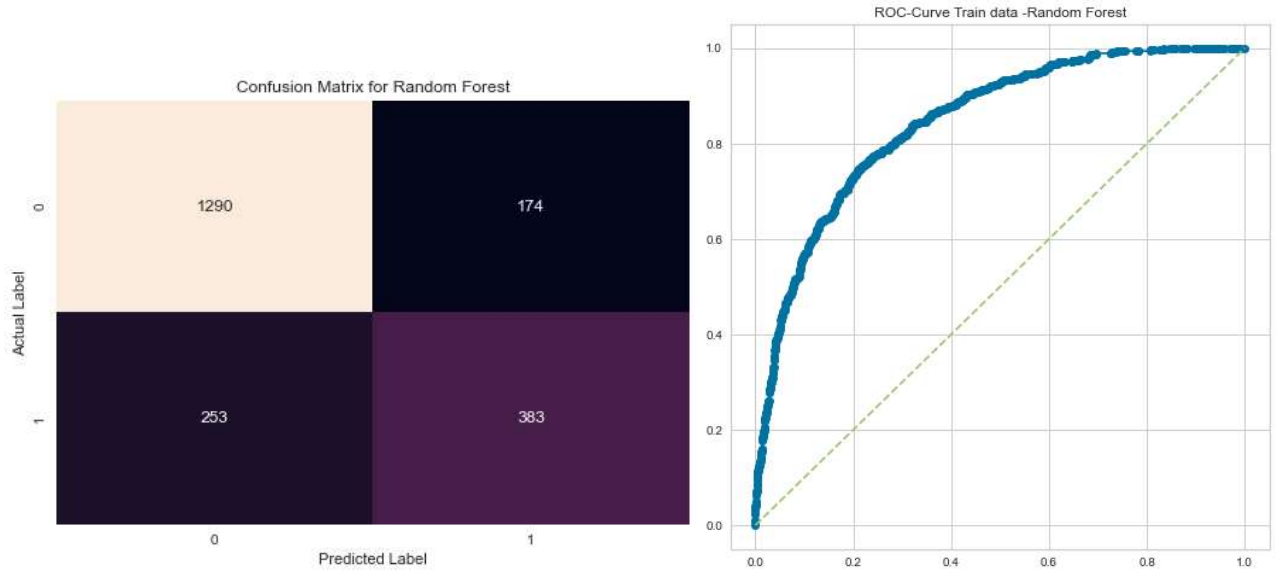


Figure - 24 Confusion Matrix and ROC Curve (RF) - Train

AUC for train dataset (Random Forest): 0.846

Performance Metrics: Test

	precision	recall	f1-score	support
0	0.82	0.90	0.86	612
1	0.73	0.58	0.65	288

```
    accuracy                           0.80       900
   macro avg       0.78      0.74      0.75       900
weighted avg       0.79      0.80      0.79       900
```

Confusion Matrix and ROC – Test (Random Forest)

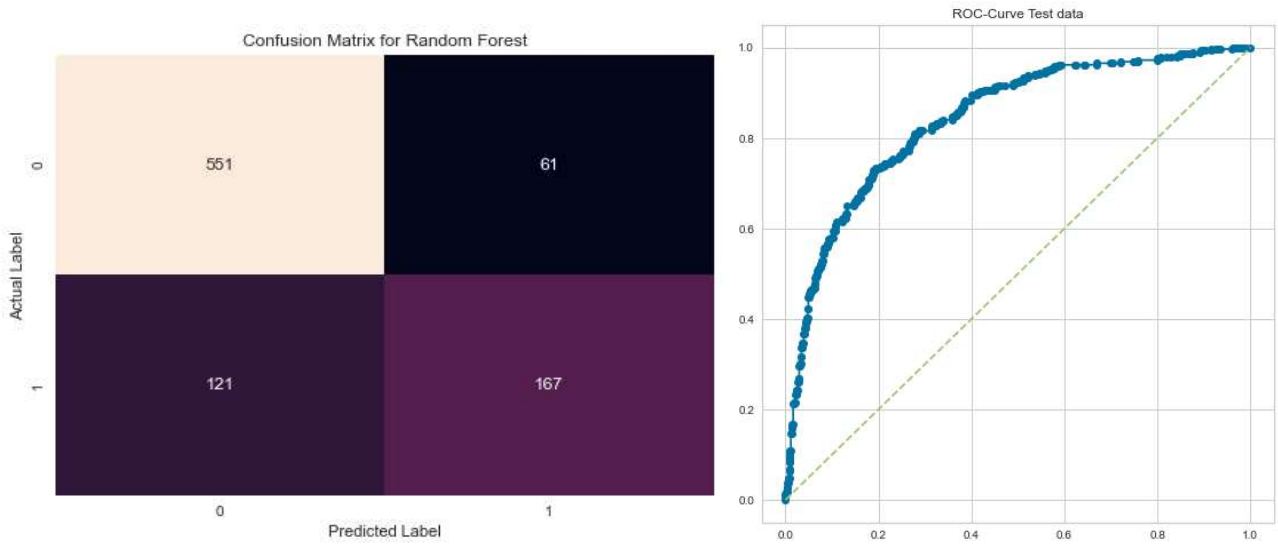


Figure - 25 Confusion Matrix and ROC Curve (RF) - Test

AUC for test dataset (Random Forest): 0.842

```
Accuracy for train dataset =  0.7966666666666666
Recall for train dataset =  0.6022012578616353
Precision for train dataset =  0.6876122082585279
AUC_ROC_score for train dataset =  0.7416743994226208
f1_score for train dataset =  0.6420787929589272

Accuracy for test dataset =  0.7977777777777778
Recall for test dataset =  0.5798611111111112
Precision for test dataset =  0.7324561403508771
AUC_ROC_score for test dataset =  0.740093954248366
f1_score for test dataset =  0.6472868217054264
```

Accuracy of the model for train and test dataset are 0.796 and 0.797 respectively. Recall for tr ain and test data set is 0.60 and 0.58 respectively which represents the model's ability to classify all the positive samples with the precision of 0.69 in train dataset and 0.73 in test dataset. Base d on the AUC_ROC score for train and test dataset which is 0.741 and 0.74 respectively, the m odel has a good sense to separate and distinguish between the classes. F1 score (harmonic m ean of precision and recall) for train dataset is 0.642 and for test dataset is 0.647.

Artificial Neural Networks

Classification Reports

Performance Metrics: Train

	precision	recall	f1-score	support
0	0.80	0.91	0.85	1464
1	0.70	0.47	0.57	636
accuracy			0.78	2100
macro avg	0.75	0.69	0.71	2100
weighted avg	0.77	0.78	0.77	2100

Confusion Matrix and ROC – Train (Artificial Neural Network)

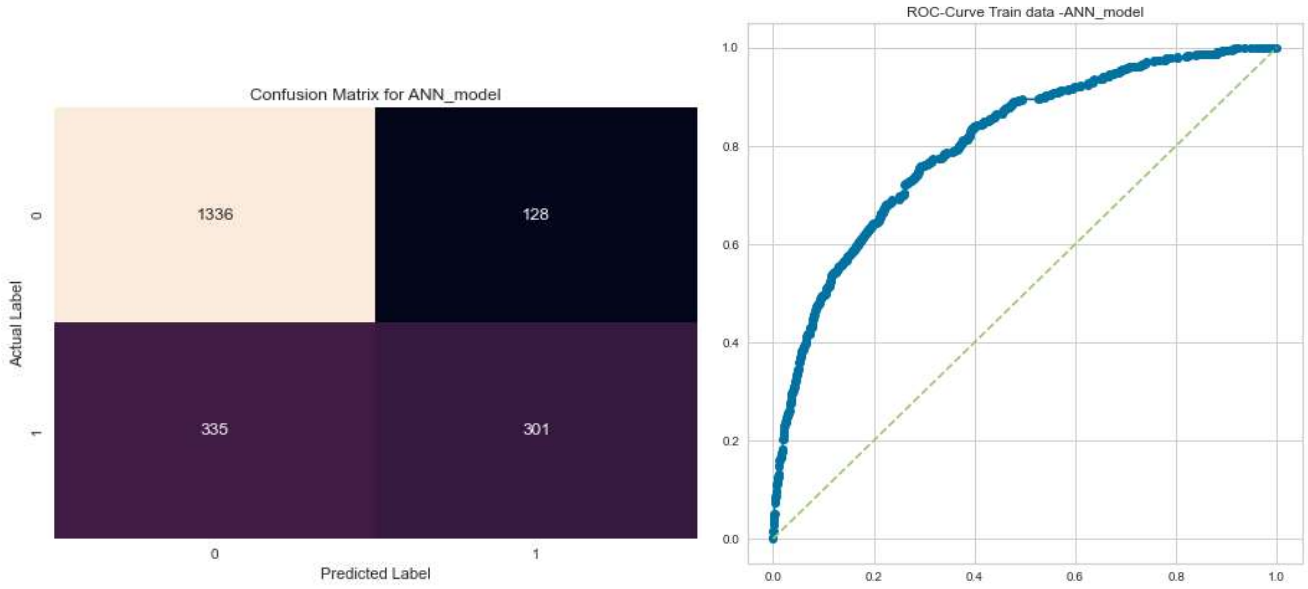


Figure - 26 Confusion Matrix and ROC Curve (ANN) - Train

AUC for train dataset (Artificial Neural Network): 0.806

Performance Metrics: Test

	precision	recall	f1-score	support
0	0.78	0.93	0.85	612
1	0.76	0.45	0.57	288
accuracy			0.78	900
macro avg	0.77	0.69	0.71	900
weighted avg	0.78	0.78	0.76	900

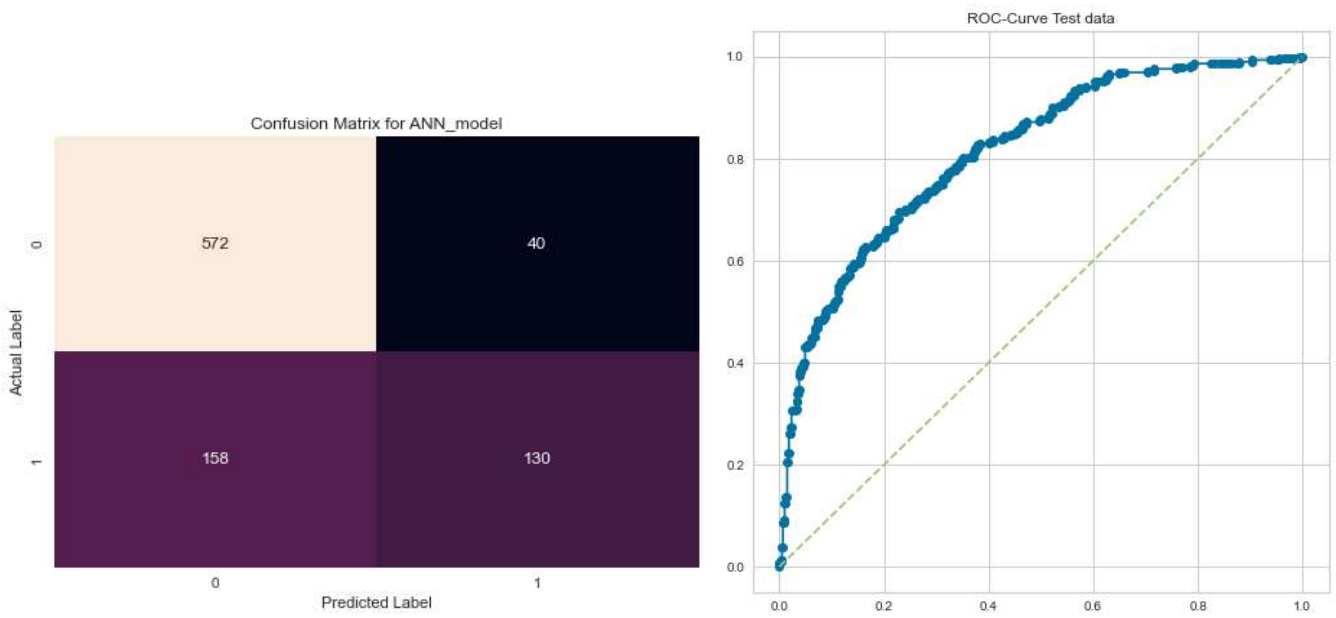


Figure - 27 Confusion Matrix and ROC Curve (ANN) - Test

AUC for test dataset (Artificial Neural Network): 0.814

```
Accuracy for train dataset =  0.7795238095238095
Recall for train dataset =  0.47327044025157233
Precision for train dataset =  0.7016317016317016
AUC_ROC_score for train dataset =  0.6929193731312507
f1_score for train dataset =  0.5652582159624414

Accuracy for test dataset =  0.78
Recall for test dataset =  0.4513888888888889
Precision for test dataset =  0.7647058823529411
AUC_ROC_score for test dataset =  0.6930147058823529
f1_score for test dataset =  0.5676855895196506
```

Accuracy of the model for train and test dataset are 0.779 and 0.78 respectively. Recall for trai n and test data set is 0.47 and 0.45 respectively which represents the model's ability to classify al l the positive samples with the precision of 0.70 in train dataset and 0.76 in test dataset. Based on the AUC_ROC score for train and test dataset which is 0.692 and 0.693 respectively, the mo del has a good sense to separate and distinguish between the classes. F1 score (harmonic me an of precision and recall) for train dataset is 0.565 and for test dataset is 0.567.

Recall and Precision are important as it reflects model's ability to identify the positive records with accuracy and f1-score is the harmonic mean of recall and precision.

High recall indicates that many of the data were predicted and high relevant data were selec ted. Other high value of f1-score shows that best result values are obtained at the precision an d recall performance measures.

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

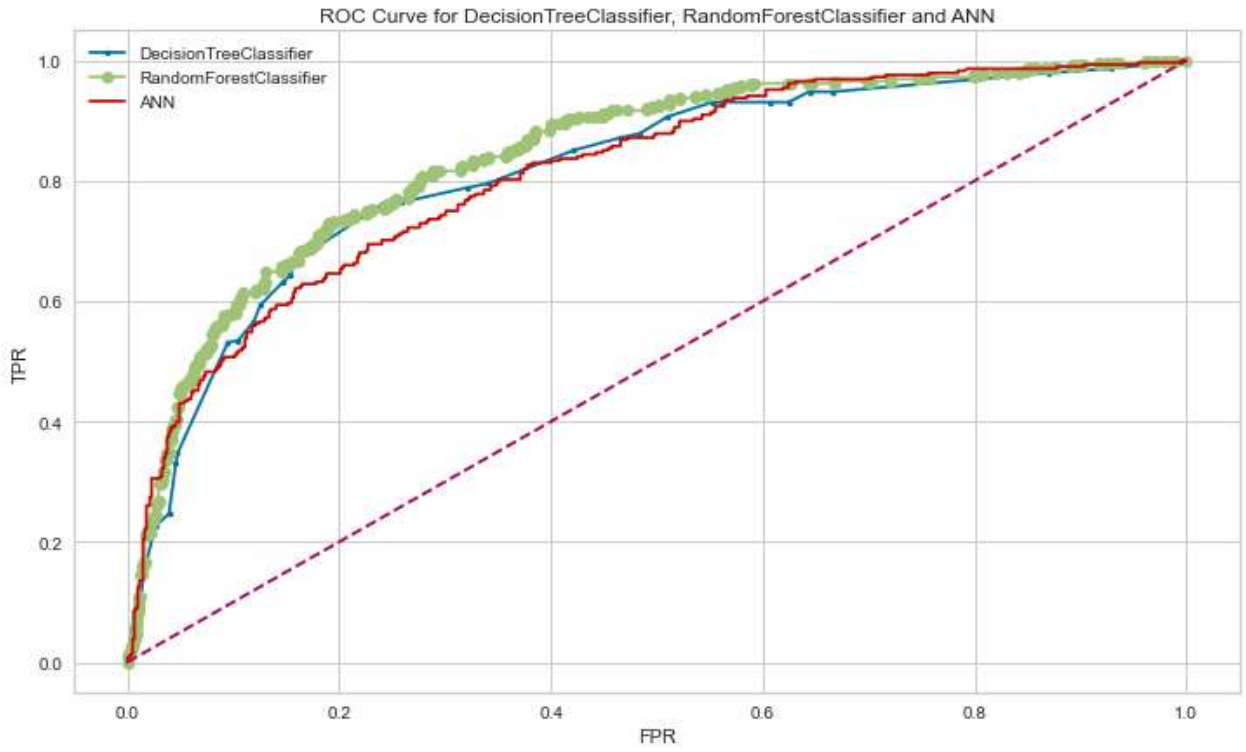


Figure - 28 Comparing ROC for CART-RF-ANN

Area under the curve for Decision Tree Classification Model is 0.8174785539215688

Area under the curve for Random Forest Classification Model is 0.8423911810094408

Area under the curve for Artificial Neural Network Model is 0.8140971087509078

Performance Matrix on Training data:

Algorithm	Accuracy	Recall	Precision	AUC_ROC Score	f1-score
Decision Tree	0.79	0.57	0.67	0.72	0.62
Random Forest	0.80	0.61	0.69	0.74	0.64
Artificial Neural Network	0.78	0.47	0.71	0.69	0.57

Performance Matrix on Testing data:

Algorithm	Accuracy	Recall	Precision	AUC_ROC Score	f1-score
Decision Tree	0.78	0.53	0.71	0.72	0.61
Random Forest	0.80	0.58	0.73	0.74	0.65
Artificial Neural Network	0.78	0.45	0.76	0.69	0.57

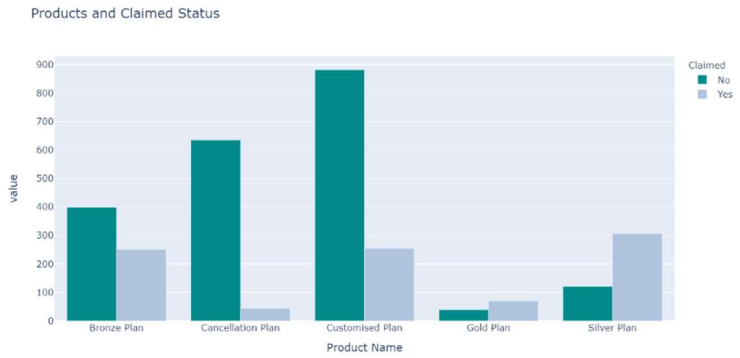
Table - 19 Comparing models (CART-RF-ANN)

The performances of the two classifiers were evaluated by using four metrics (Accuracy, Precision, Recall and F-score). All the three models have high prediction accuracy and two of them shows promising prediction accuracy with high recall and precision. ANN (Artificial Neural Networks) has high accuracy however the recall and precision are low compared to other models. RF (Random Forest) model prediction accuracy is higher than CART (Decision Tr ee) model in predicting insurance claims.

For the final model, RF have higher accuracy in prediction of claims for train and test dataset. The recall for train dataset is 61% with precision of 69%; recall for test dataset is 58% with precisi on of 73%. The f1-score for RF model is 65%.

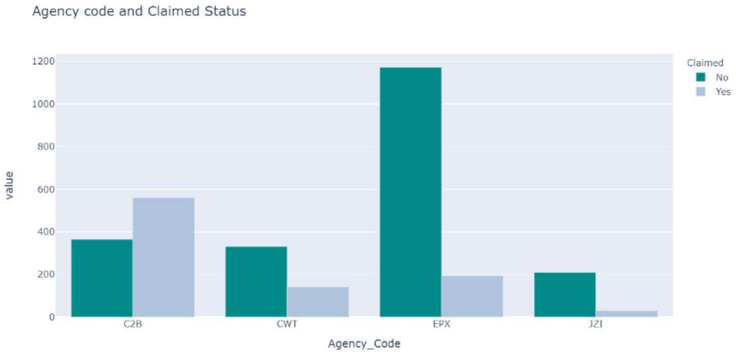
2.5 Inference: Based on the whole Analysis, what are the business insights and recommendations

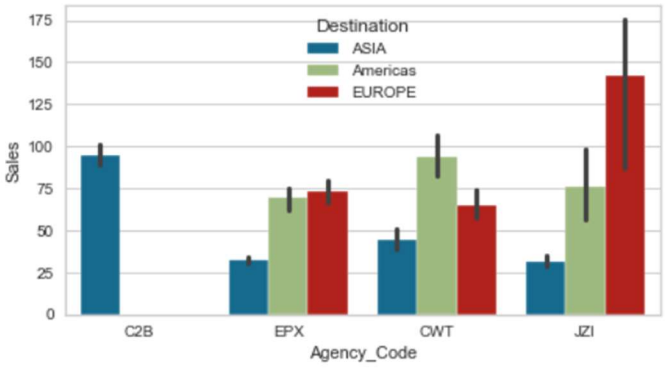
Based on the insurance claim analysis, the overall data only represents the claim status, if the company can place a preventive measure to track the description of the incidents it can prevent the claim from reoccurring.



Gold Plan and Silver Plan have highest claim ratio. However, the customized plan has lowest claim and performing better than rest of the plans. The company should focus on marketing strategies for this customized plan and should revisit the policies for the gold and silver plans.

The C2B agency code has highest claims and preferred selling destination is Asia. Implementing a proper risk mitigation strategy is important for such instances. Through trend analysis the company can find the ones that are most frequent or costly and work on preventing those.





Large percent of the claim is from Airlines insurance type, C2B agency code, 'Gold' plan and 'Silver' Plan and Asia destination. Highest profitable destination is Americas, however most of the claims are accepted with this destination than not.

Instead of accepting the claims, a thorough investigation to figure out what's causing them can lead to prevention of such claims in the future.