PROJECT: CLUSTERING & DATA MINING

SUBMITTED BY:

Sarthak Diwan

Intern – IT

Contents

PROBLEM STATEMENT 1 - CLUSTERING

A leading bank wants to develop a customer segmentation to give promotional offers to its customers. They collected a sample that summarizes the activities of users during the past few months. Please note that it is a summarized data that contains the average values in all the columns considering all the months, and not for any particular month. You are given the task to identify the segments based on credit card usage.

Data Dictionary:

1. spending: Amount spent by the customer using the credit card per month (in 1000s). For example, if the spending is 19.94, then the customer has actually spent (19.94 * 1000 = 19940) 19940 Rs per month on an average.

2. advance_payments: Amount paid by the customer in advance by cash even before the credit card bill got generated for any particular month (in 100s). For example, if the advance_payments is 16.92, then the customer has paid (16.92*100 = 1692) 1692 Rs on an average per month.

3. probability_of_full_payment: Probability of the credit card payment done in full by the customer to the bank. If it is 0.8752, then it means that the customer has a chance of 87.52% to pay the entire credit card bill on an average per month.

4. current_balance: The balance amount left in the credit card account to make the future purchases (in 1000s). For example, if the current_balance is 6.675, then it means that the customer is left out with a credit card balance of (6.675*1000 = 6675) 6675 Rs which he can use for the future purchases.

5. credit_limit: Limit of the amount in credit card (10000s) sanctioned by the bank to the customer. For example, if the credit_limit is 3.763, it means that the customer has been sanctioned a credit card limit of (3.763*10000 = 37,630) 37630 Rs.

6. min_payment_amt : The average minimum amount paid by the customer while making payments for the credit card bill purchases made monthly (in 100s). For example, if the min_payment_amt is 3.252, it means that the customer has paid only (3.252*100 =

325.2) 325.2 Rs as the minimum payment instead of paying the entire credit card bill amount on an average per month.

7. max_spent_in_single_shopping: Maximum amount spent by the customer for a single transaction using the credit card (in 1000s). For example, if the max_spent_in_single_shopping is 6.55, it means that the customer has spent a maximum of (6.55*1000=6550) 6550 Rs for a single transaction using credit card on an average per month.

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

1.1) Checking the data

These are the first few rows of the dataset

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

Figure 1: First 5 rows of the dataset

These are the last few rows of the dataset

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
205	13.89	14.02	0.8880	5.439	3.199	3.986	4.738
206	16.77	15.62	0.8638	5.927	3.438	4.920	5.795
207	14.03	14.16	0.8796	5.438	3.201	1.717	5.001
208	16.12	15.00	0.9000	5.709	3.485	2.270	5.443
209	15.57	15.15	0.8527	5.920	3.231	2.640	5.879

Figure 2: Last 5 rows of the dataset

Observations:

- Data looks good based on initial records seen in top 5 and bottom 5.
- These are the top 5 rows of the data, with double digit values in Spending and advance_payments, single digit values in current_balance, credit_limit, min_payment_amt and max_spent_in_single_shopping, and point values in probability_of_full_payment.
- The shape of the data is `(210, 7)`

1.2) Information of the dataset and checking the null values

Here is some information about 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
```

Observations:

- There are a total of 210 records and 7 variables in the dataset.
- No missing record based on initial analysis.
- All the variables are numeric i.e. float.

Now, let's see whether there is any null values in the dataset or not.

```
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
```

Observation:

- From the above results, we can confidently say that there is no missing value present in the dataset.

1.3) Descriptive Analysis

Now we will see the Descriptive analysis of the dataset.

	count	mean	std	min	25%	50%	75%	90%	max
spending	210.0	14.847524	2.909699	10.5900	12.27000	14.35500	17.305000	18.9880	21.1800
advance_payments	210.0	14.559286	1.305959	12.4100	13.45000	14.32000	15.715000	16.4540	17.2500
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.85690	0.87345	0.887775	0.8993	0.9183
current_balance	210.0	5.628533	0.443063	4.8990	5.26225	5.52350	5.979750	6.2733	6.6750
credit_limit	210.0	3.258605	0.377714	2.6300	2.94400	3.23700	3.561750	3.7865	4.0330
min_payment_amt	210.0	3.700201	1.503557	0.7651	2.56150	3.59900	4.768750	5.5376	8.4560
max_spent_in_single_shopping	210.0	5.408071	0.491480	4.5190	5.04500	5.22300	5.877000	6.1850	6.5500

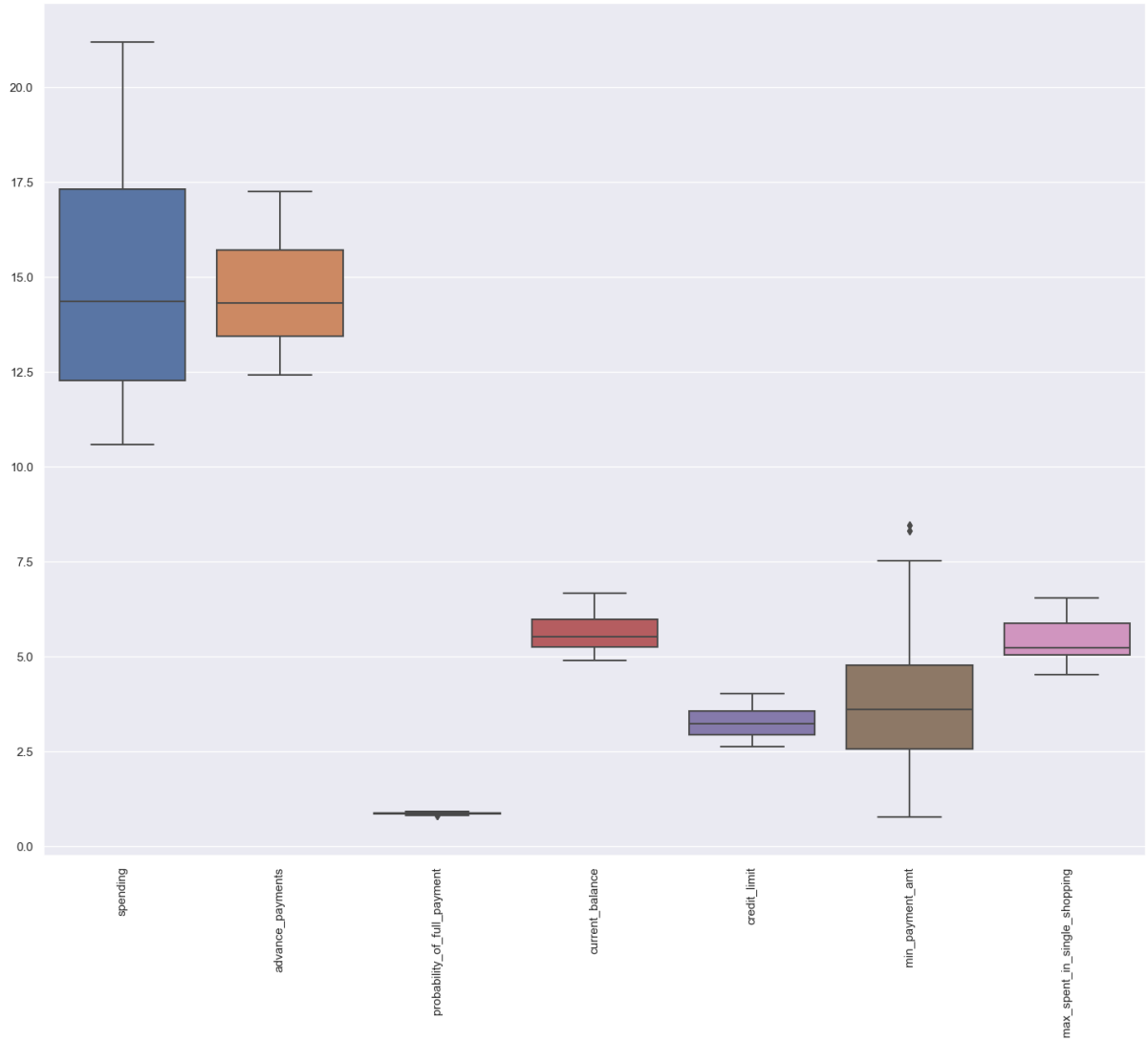
Observations:

- On an average, customers spend INR 14847.524 per month.
- advance_payments done by the customers ranges from INR 1241 to 1725.
- The average probability_of_full_payment made by the customer to the bank is 87.09%, the highest probability is 91.83% and the lowest probability is 80.81%.
- If we observe the values across the different features, we see in most of the cases the mean and median seem to be very near to each other, indicating that the shape of all numerical values seems to be more or less normally distributed.
- We have also included the 90% to see a variation and it looks evenly distributed.
- Standard Deviation is high for spending as compared to others.
- The difference between 25th percentile and min value of probability_of_full_payment is large, we can determine that there must be some outliers in this column. Also, the box plot will be right skewed as the range of probability_of_full_payment data has more weightage from its 25th percentile to median is more than the median to 75th percentile.
- Similarly, for min_payment_amt since the difference is quite large, we can determine that there will be outliers in this data column.

1.4) Outliers

As discussed earlier, we have determined that there must be some outliers in probability_of_full_payment an min_payment_amt.

```
No. of outliers in probability_of_full_payment: 3
No. of outliers in min_payment_amt: 2

No. of attributes with outliers: 2
```



We can see that in the above figure that almost all the box plots seem to be positively skewed except for probability_of_full_payment.

Though we did treated the outlier, we still see one as per the boxplot, it is okay, as it is no extreme and on lower band.

1.5) Univariate Analysis

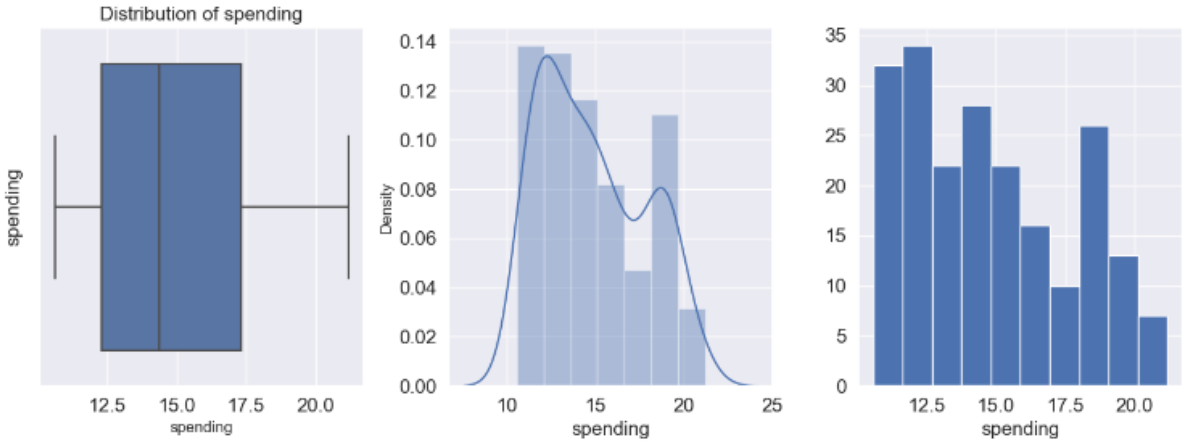
- Univariate Analysis is performed for all the numeric variables individually to display their statistical description.

- Visualized the variables using distplot and histogram or barchart to view the distribution and the boxplot to view 5-point summary and outliers if any.

1.5.1) Spending

```
Minimum spending:  10.59
Maximum spending:  21.18
Mean value:  14.847523809523807
Median value:  14.355
Standard deviation:  2.909699430687361
Null values:  False
```

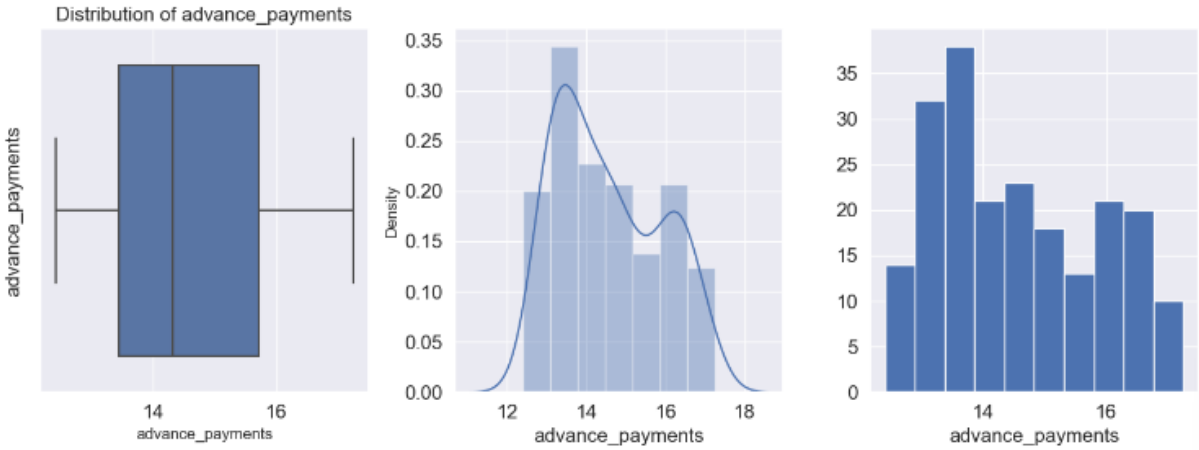
```
Number of outliers in spending upper :  0
Number of outliers in spending lower :  0
% of Outlier in spending upper:  0 %
% of Outlier in spending lower:  0 %
```



1.5.2) advance_payments

```
Minimum advance_payments:  12.41
Maximum advance_payments:  17.25
Mean value:  14.559285714285714
Median value:  14.32
Standard deviation:  1.305958726564022
Null values:  False
```

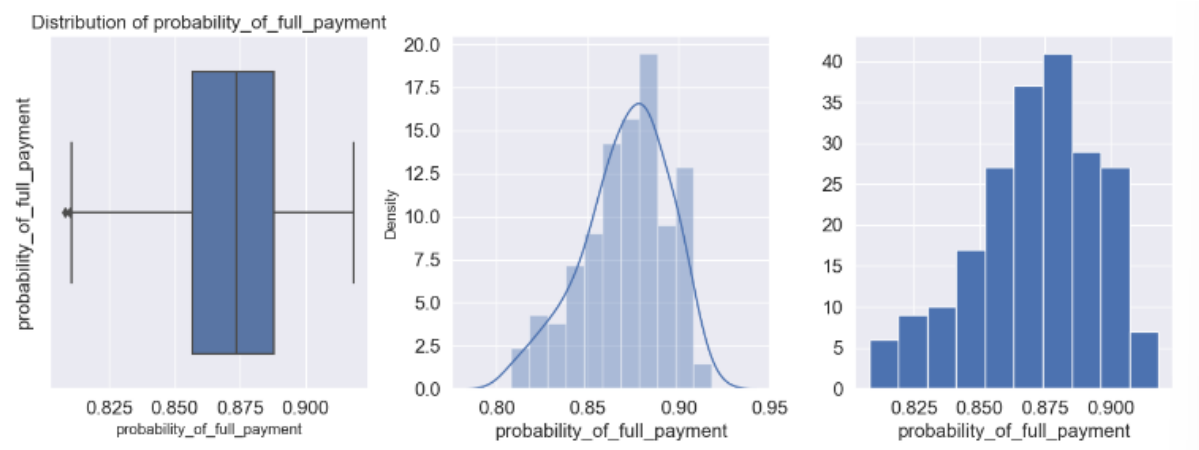
```
Number of outliers in advance_payments upper :  0
Number of outliers in advance_payments lower :  0
% of Outlier in advance_payments upper:  0 %
% of Outlier in advance_payments lower:  0 %
```



1.5.3) probability_of_full_payment

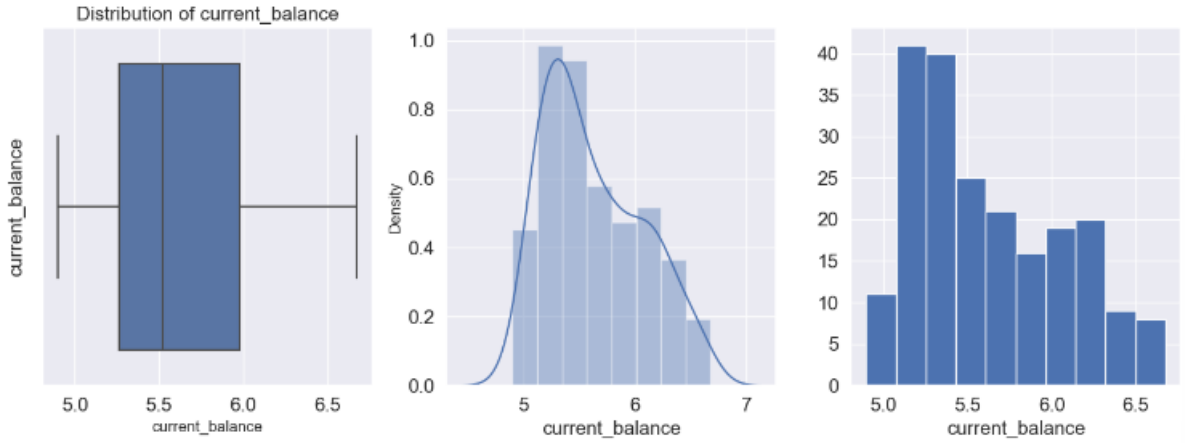
```
Minimum probability_of_full_payment  0.8081
Maximum probability_of_full_payment:  0.9183
Mean value:  0.8709985714285714
Median value:  0.8734500000000001
Standard deviation:  0.0236294165838465
Null values:  False
```

```
Number of outliers in probability_of_full_payment upper :  0
Number of outliers in probability_of_full_payment lower :  3
% of Outlier in probability_of_full_payment upper:  0 %
% of Outlier in probability_of_full_payment lower:  1 %
```



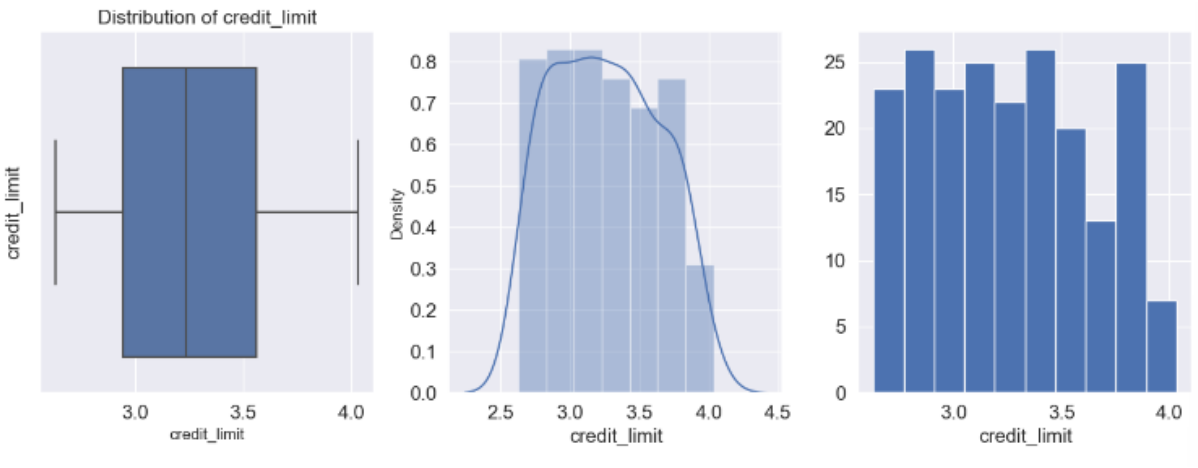
1.5.4) current_balance

```
Minimum current_balance:  4.899
Maximum current_balance:  6.675
Mean value:  5.628533333333333
Median value:  5.5235
Standard deviation:  0.44306347772644944
Null values:  False

Number of outliers in current_balance upper :  0
Number of outliers in current_balance lower :  0
% of Outlier in current_balance upper:  0 %
% of Outlier in current_balance lower:  0 %
```



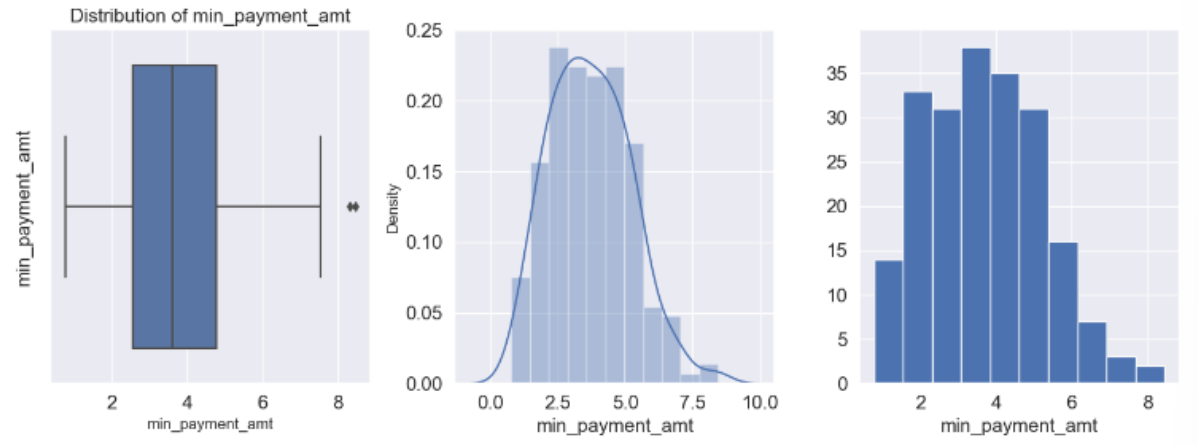
1.5.5) credit_limit

```
Minimum credit_limit:  2.63
Maximum credit_limit:  4.033
Mean value:  3.258604761904762
Median value:  3.237
Standard deviation:  0.37771444490658734
Null values:  False

Number of outliers in credit_limit upper :  0
Number of outliers in credit_limit lower :  0
% of Outlier in credit_limit upper:  0 %
% of Outlier in credit_limit lower:  0 %
```



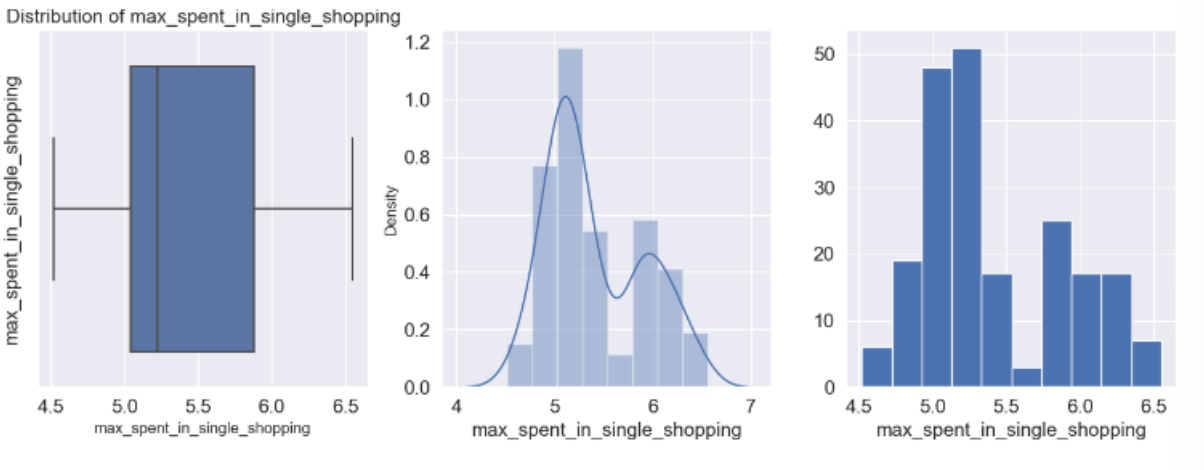
1.5.6) min_payment_amount

```
Minimum min_payment_amt:  0.7651
Maximum min_payment_amt:  8.456
Mean value:  3.7002009523809516
Median value:  3.599
Standard deviation:  1.5035571308217792
Null values:  False

Number of outliers in min_payment_amt upper :  2
Number of outliers in min_payment_amt lower :  0
% of Outlier in min_payment_amt upper:  1 %
% of Outlier in min_payment_amt lower:  0 %
```

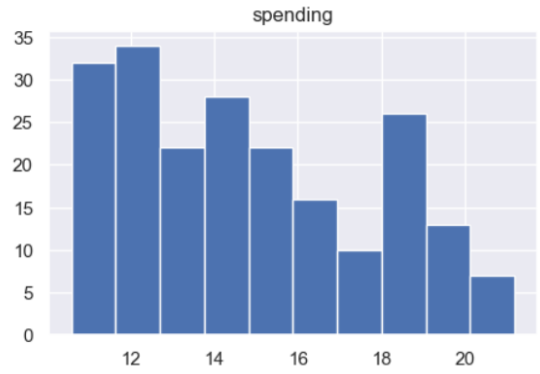


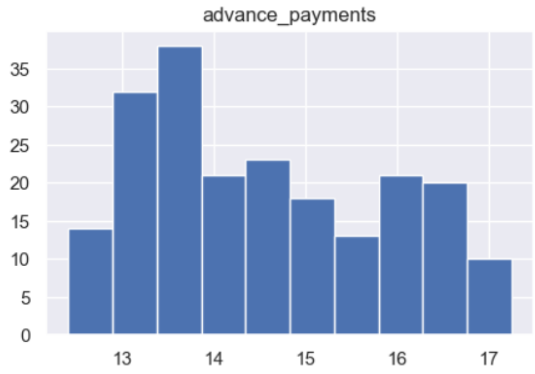
1.5.7) max_spent_in_single_shopping

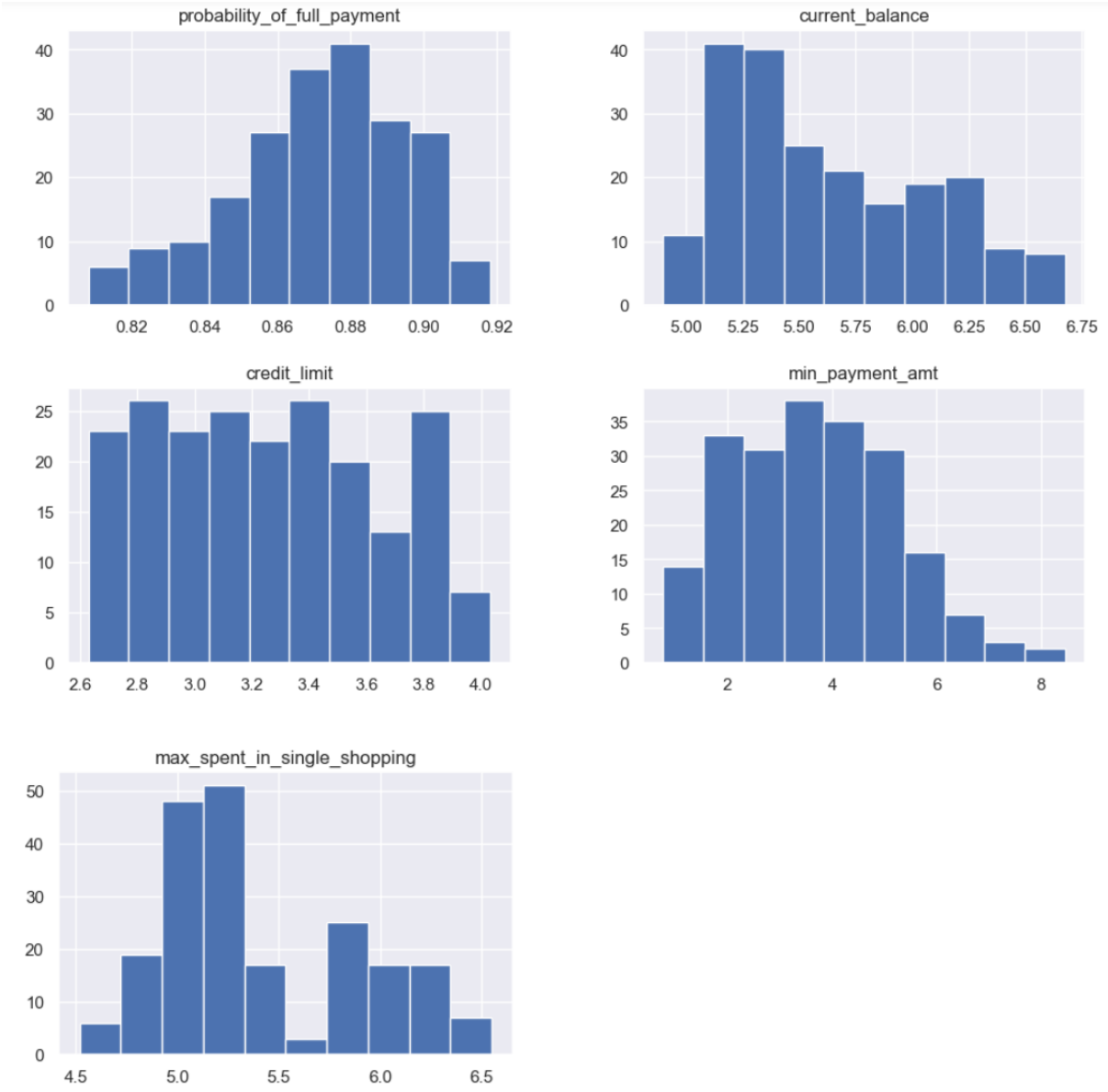
```
Minimum max_spent_in_single_shopping:  4.519
Maximum max_spent_in_single_shoppings:  6.55
Mean value:  5.4080714285714295
Median value:  5.223000000000001
Standard deviation:  0.49148049910240543
Null values:  False

Number of outliers in max_spent_in_single_shopping upper :  0
Number of outliers in max_spent_in_single_shopping lower :  0
% of Outlier in max_spent_in_single_shopping upper:  0 %
% of Outlier in max_spent_in_single_shopping lower:  0 %
```



1.6) Histograms of all the variables







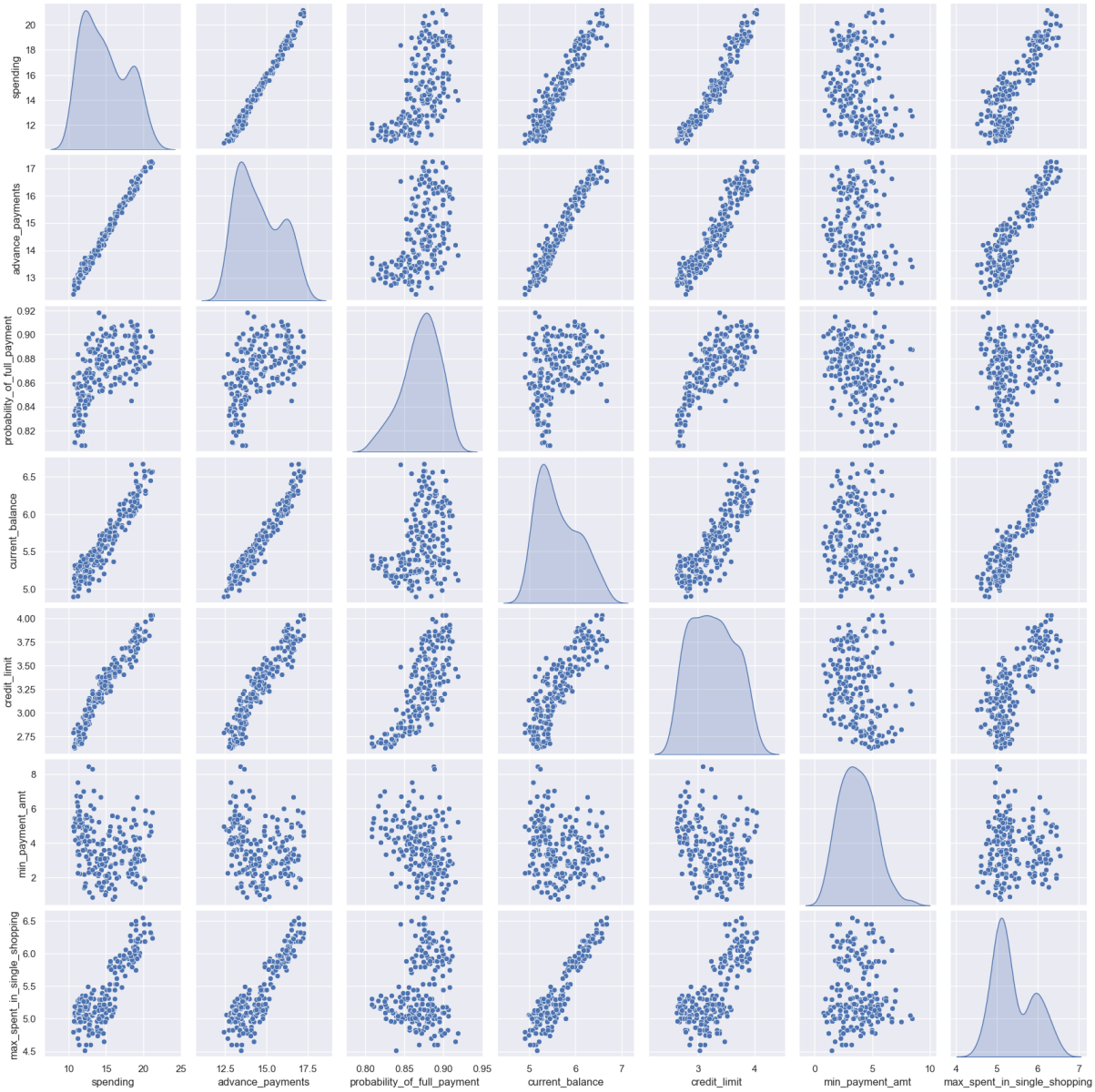
1.7) Skewness

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

In the above data, we can see that apart from probability_of_full_payment, all the other variables are positively skewed. Also, we can see that the range of Skewness values lies between -0.5 and 0.5 approximately, which conveys that the distribution is approximately symmetric.

Distribution is skewed to right tail for all the variable except probability_of_full_payment variable, which has left tail.

1.8) Multivariate Analysis



Observations:

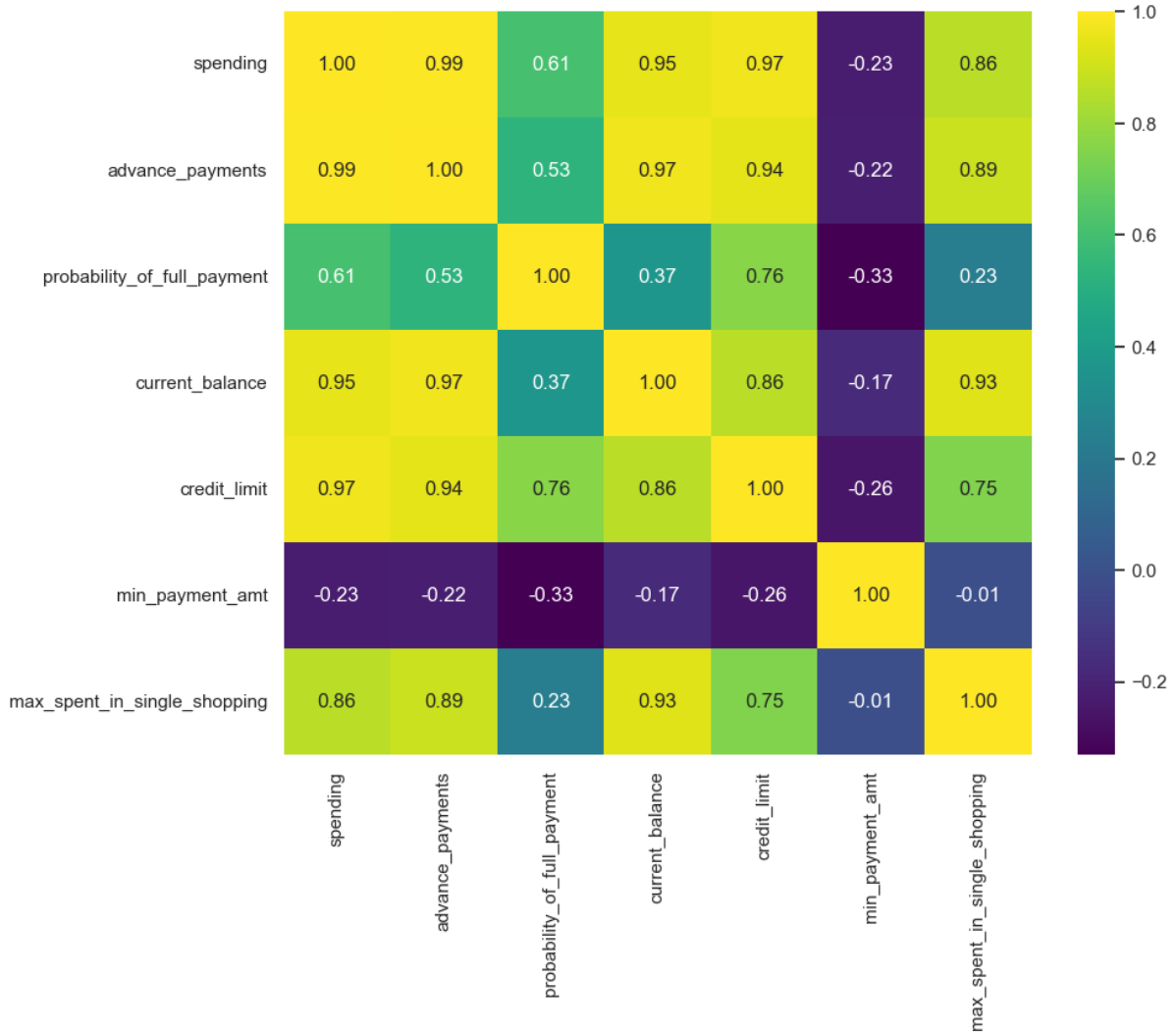
- Strong positive correlation between:-
  a) spending and advance_payments,
  b) advance_payments and current_balance,
  c) credit_limit and spending
  d) spending and current_balance
  e) credit_limit and advance_payments
  f) max_spent_in_single_shopping and current_balance
- Customers with higher spendings tends to make higher advance_payments. Looking at these factors, it explains the higher credit_limit they have been provided with.
- We can observe that as credit_limit increasing, ccurrent_balance is also increasing.
- Customers with the high probability_of_full_payment have been provided with higher credit_limit.

- Ccustomers with higher current_balance tend to make higher max_spent_in_single_shopping.
- Higher the credit_limit enables higher spending capacity of the customer.

We can see high correlations in a lot of variables here, let's have a look at the correlation table and the heatmap for more clarity.

1.9) Correlation

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
spending	1.000000	0.994341	0.608288	0.949985	0.970771	-0.229572	0.863693
advance_payments	0.994341	1.000000	0.529244	0.972422	0.944829	-0.217340	0.890784
probability_of_full_payment	0.608288	0.529244	1.000000	0.367915	0.761635	-0.331471	0.226825
current_balance	0.949985	0.972422	0.367915	1.000000	0.860415	-0.171562	0.932806
credit_limit	0.970771	0.944829	0.761635	0.860415	1.000000	-0.258037	0.749131
min_payment_amt	-0.229572	-0.217340	-0.331471	-0.171562	-0.258037	1.000000	-0.011079
max_spent_in_single_shopping	0.863693	0.890784	0.226825	0.932806	0.749131	-0.011079	1.000000



Observations:

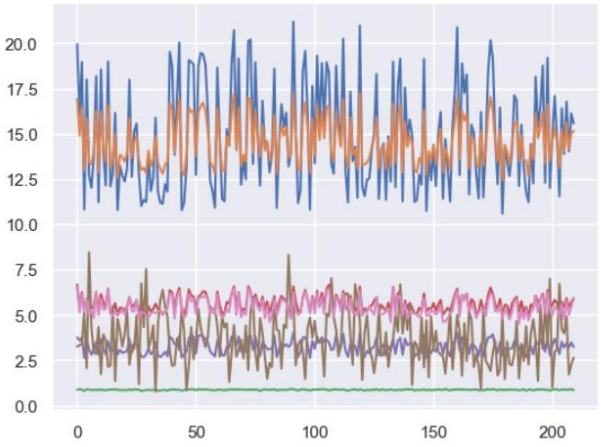
- spending is highly positively correlated with advance_payments, current_balance, credi_limit and max_spent_in_single_shopping. We can say that higher credit_limit increases customer's spending capacity using credit card, hence higher the max_spent_in_single_shopping. Higher credit_limit explains the higher current_balance remained in the credit card.
- advance_payments is also highly correlated with current_balance and credit_limit.
- probability_of_full_payment is moderately correlated with credit_limit. This explains that customers with higher probability of making full payment have been granted high credit limit, assuming that they won't default.
- min_payment_amt is negatively correlated with all the columns, but the correlation is not significant enough to derive any inferences.

Observing the correlations in the dataset , we infer that high multicollinearity exists. However, we will not be treating multi-collinearity, as it does not impact the clustering process.

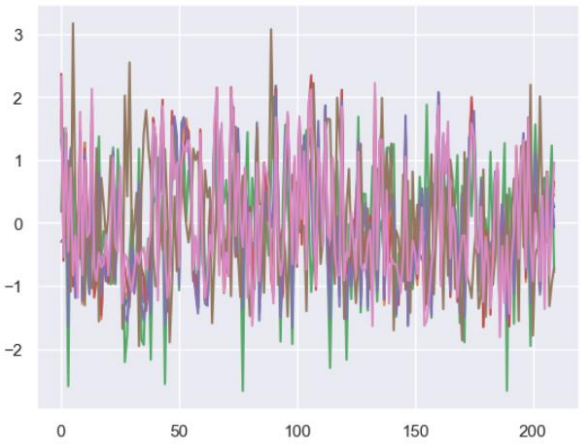
2) Do you think scaling is necessary for clustering in this case? Justify.

2.1) Scaled and Unscaled plots

Clean Dataset



Clean Dataset Scaled



2.2) Values after Scaling

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754355	1.811968	0.178230	2.367533	1.338579	-0.298806	2.328998
1	0.393582	0.253840	1.501773	-0.600744	0.858236	-0.242805	-0.538582
2	1.413300	1.428192	0.504874	1.401485	1.317348	-0.221471	1.509107
3	-1.384034	-1.227533	-2.591878	-0.793049	-1.639017	0.987884	-0.454961
4	1.082581	0.998364	1.196340	0.591544	1.155464	-1.088154	0.874813

2.3) Justification

Scaling of the data is necessary when the variables of the dataset are of different scales, i.e. one variable is in 1000s and others in only 10s.

There are certain variables which have values of different scales, like spending and credit_limit which have values in the multiples of 10,000s; advance_payments, current_balance and max_spent_in_single_shopping has values in multiples of thousand; and probability_of_full_payment have values less than 1. Since the data in these variables are of different scales and the standard deviation of each variable is also vary. It is tough to compare them. Hence, the scaling of the variables is necessary for clustering in this case.

Yes, I think that Scaling is necessary for clustering. In this case as standardizing the data prevents variables with larger scales from dominating the clustering process. We scaled the data using Standard scaler. I tried Standardization using Z score.

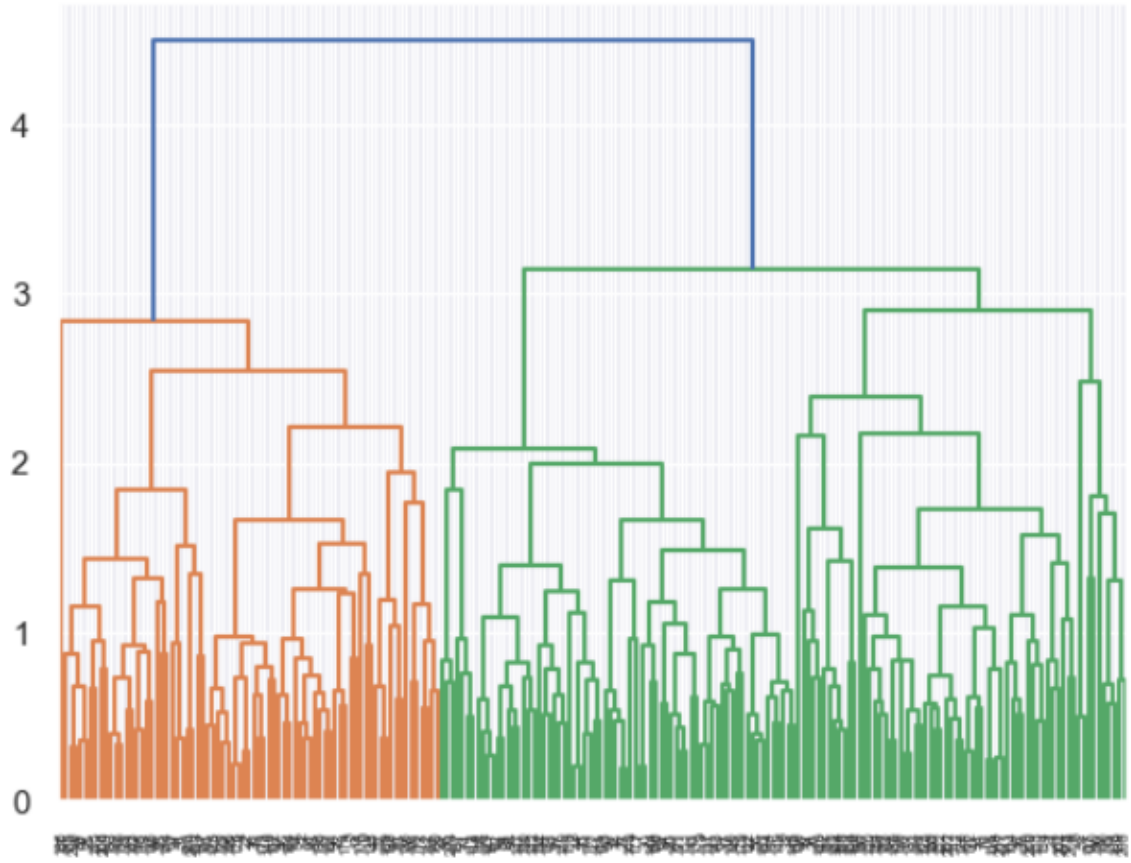
3) Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them.

For hierarchical clustering, the number of optimum clusters are obtained after the model is run, then we analyze the dendrogram to decide on how many clusters we need.

To perform hierarchical clustering, we are selecting dendrogram and linkage functions. Linkage function is used to compute the distances and merging the clusters and we are using ward and average linkage.

3.1) Creating the Dendrogram using average linkage method of Hierarchical Clustering

We have created a dendrogram using scaled data by average linkage method.

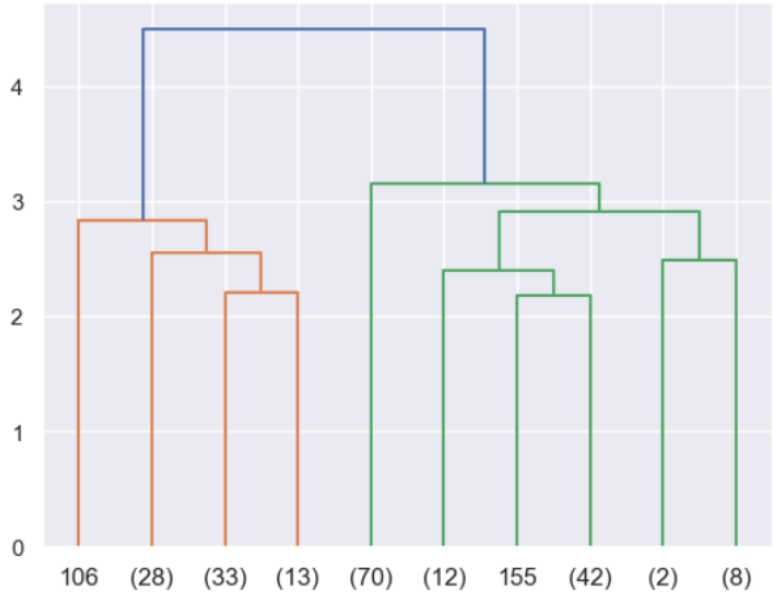


As the size of the dendrogram is very compact, but we can see that 2 clusters have been created.

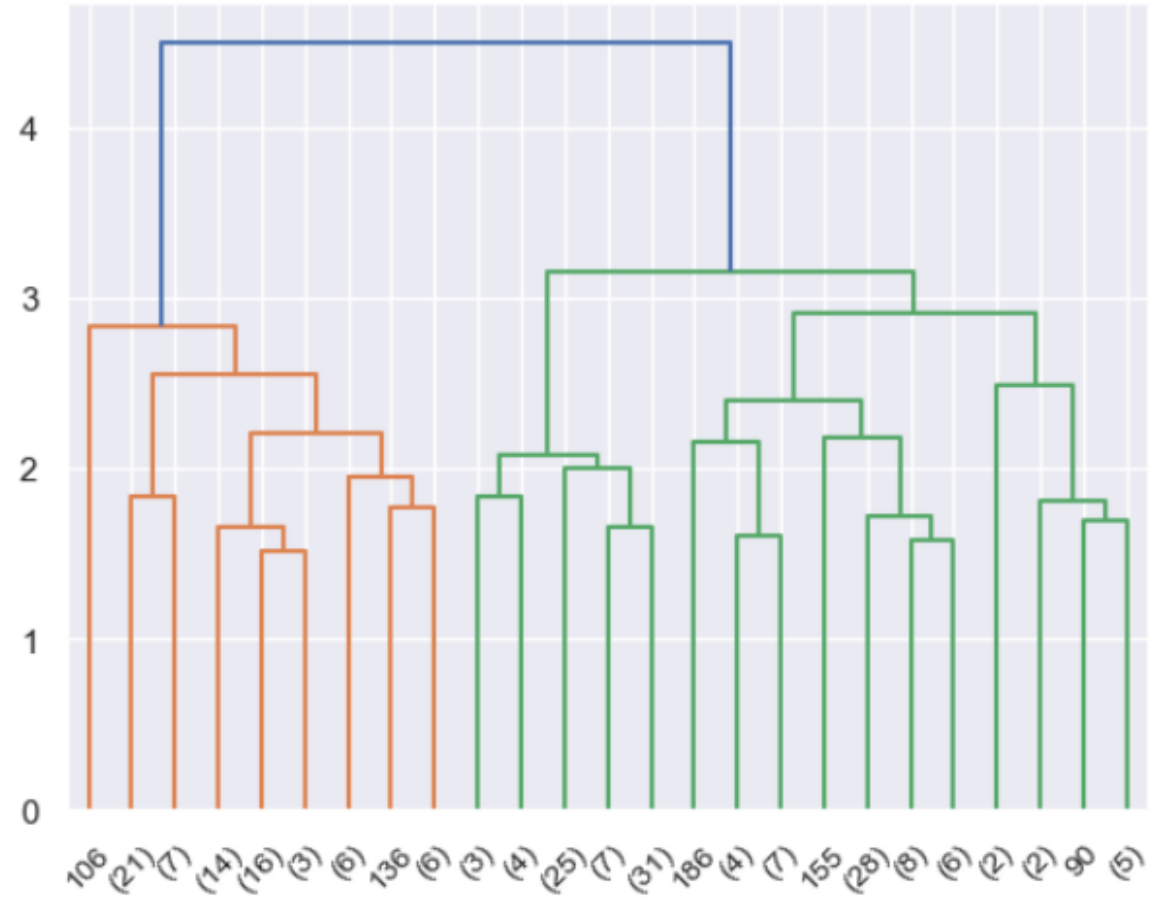
We truncated to get a clear visual to decide optimum no. of clusters.

3.1.1) Cutting the dendrogram with suitable leaf nodes:

With 10 leaf nodes to display:



With 25 leaf nodes to display:



By visualizing we decides to go with 3 clusters only.

3.1.2) Importing fcluster module to create clusters

- We are going to create clusters with 3 number of clusters.
- The criterion used to determine the number of cluster when the number of clusters is not explicitly specified is maxclust.

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

Here is how the data looks like:

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

It calculates the mean value for each numerical column within each group. Essentially, it computes the average values of each feature for each cluster in the dataset i.e. cluster profiles.

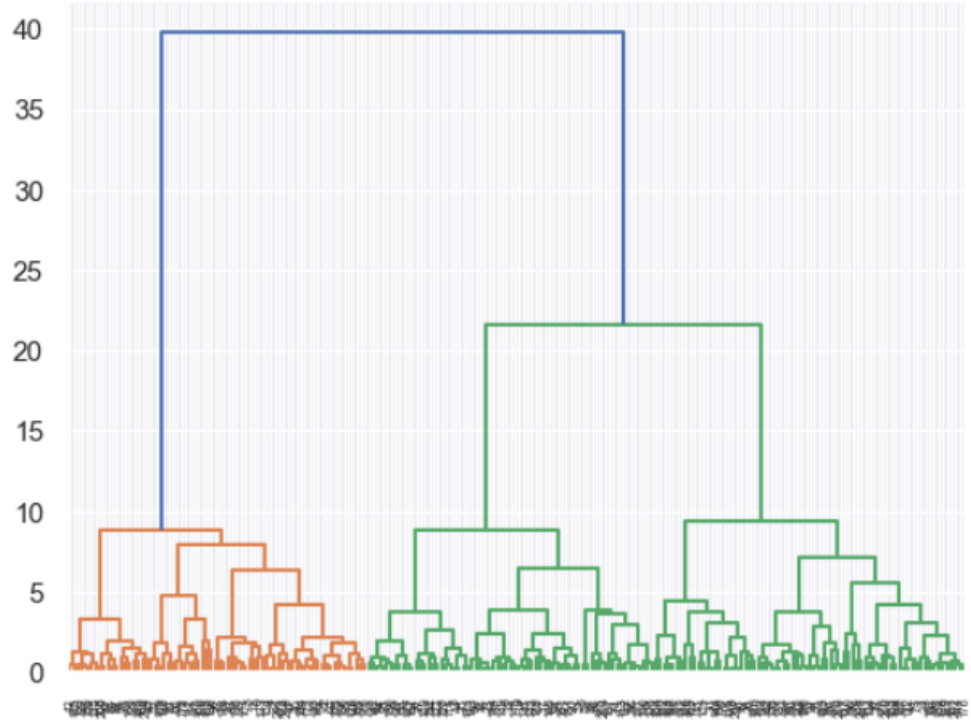
clusters-3	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
1	18.129200	16.058000	0.881595	6.135747	3.648120	3.650200	5.987040	75
2	11.916857	13.291000	0.846766	5.258300	2.846000	4.619000	5.115071	70
3	14.217077	14.195846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

And this is the cluster frequency:

```
clusters-3
1    75
2    70
3    65
Name: count, dtype: int64
```

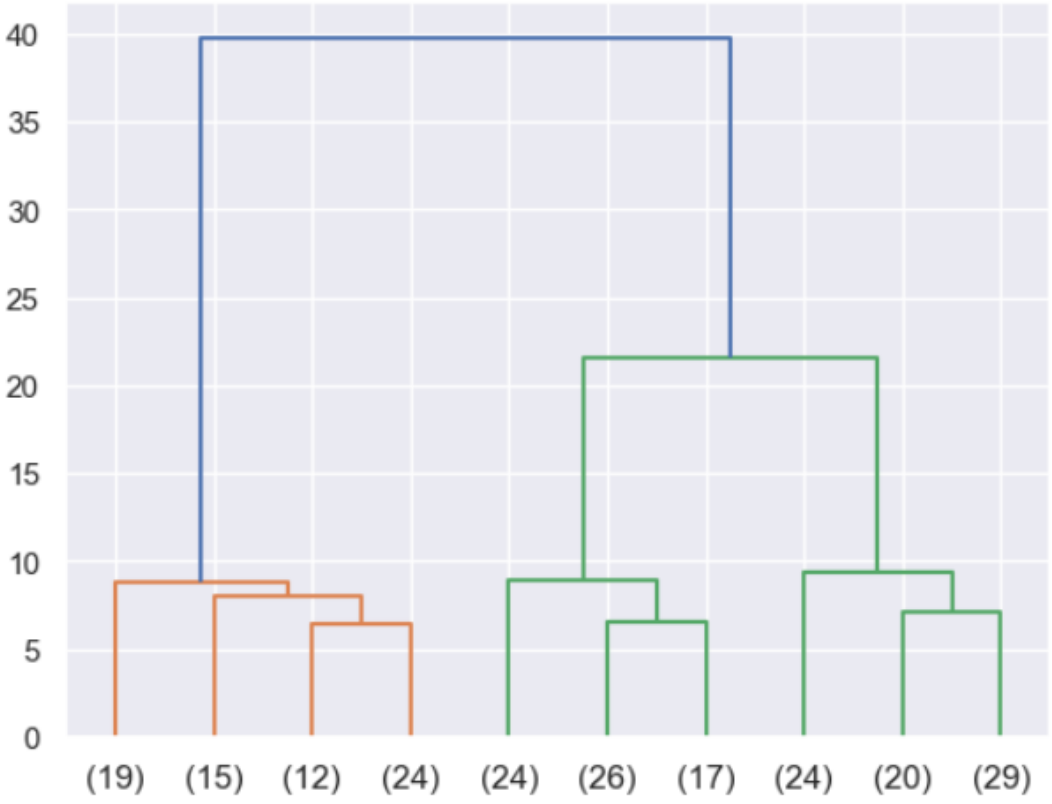
3.2) Creating the Dendrogram using ward linkage method of Hierarchical Clustering

We have created a dendrogram using clean dataset by ward linkage method.



3.2.1) Cutting the dendrogram with suitable leaf nodes:

With 10 leaf nodes to display:



3.2.2) Importing fcluster module to create clusters

- We are going to create clusters with 3 number of clusters.
- The criterion used to determine the number of cluster when the number of clusters is not explicitly specified is maxclust.

```
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)
```

Here is how the data looks like:

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

It calculates the mean value for each numerical column within each group. Essentially, it computes the average values of each feature for each cluster in the dataset i.e. cluster profiles.

clusters-3	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	0.848072	5.238940	2.848537	4.949433	5.122209	67
3	14.199041	14.233562	0.879190	5.478233	3.226452	2.612181	5.086178	73

And this is the cluster frequency:

```
clusters-3
1    70
2    67
3    73
Name: count, dtype: int64
```

Observations:

- Both the method are almost similar means , minor variation, which we know it occurs.
- We for cluster grouping based on the dendrogram, 3 or 4 looks good. Did the further analysis, and based on the dataset had gone for 3 group cluster solution based on the hierarchical clustering

- Also in real time, there could have been more variables value captured - tenure, BALANCE_FREQUENCY, balance, purchase, installment of purchase, others.
- And three group cluster solution gives a pattern based on high/medium/low spending with max_spent_in_single_shopping (high value item) and probability_of_full_payment(payment made).

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.

4.1) K-Means Clustering

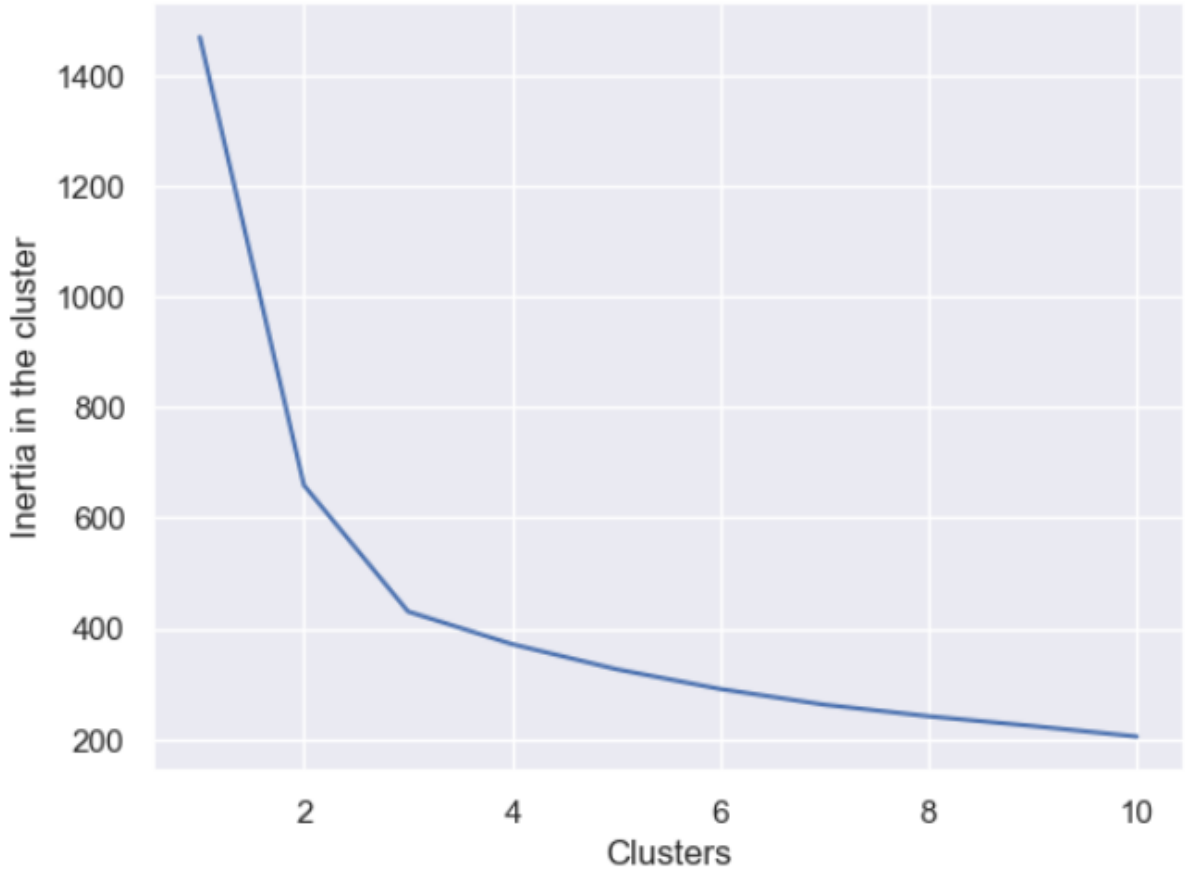
K-Means is a non-hierarchical approach to forming good clusters is to prespecify a desired number of clusters, k. The 'means' in the K-means refers to averaging of the data; that is, finding the centroid.

In this method the partitions are made such that non-overlapping groups having no hierarchical relationships between themselves.

We fit the scaled data, calculating the inertia and then total WSS.

Now with the help of curve, for a given number of clusters, the total WCSS is computed.

For K-Means clustering, we need to know the optimum number of clusters we require, before the model is run. In order to decide the optimum number of clusters that we require, a WSS plot is created and that value of k is chosen to be optimum, where addition of one more cluster does not lower the value of total WSS appreciably:

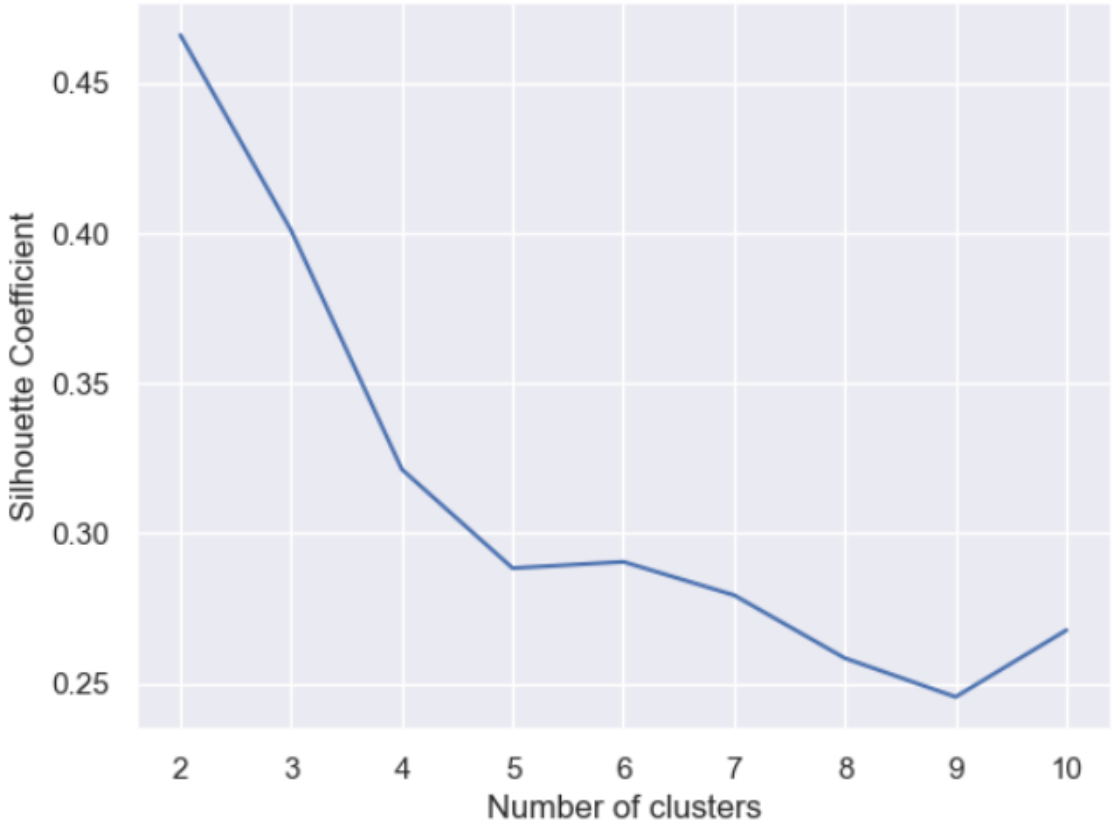


We go with 3 cluster segmentation as per our business recommendation as we see an elbow at cluster number = 2 and 3, but we are going with cluster number = 3, after which the plot seems redundant.

The optimum number of clusters can also be verified using the Silhouette Score. Silhouette Score shows if the sample is enough far away from the neighboring clusters. The Silhouette Score value close to +1 indicates clusters are well- separated, close to 0 indicates clusters are not separated well enough and close to -1 indicates clustering is not done properly.

In our case Silhouette Score comes out to be 0.3291966792017613.

To check if all the records are mapped correctly, we calculated Silhouette Samples for each record. The minimum value of Silhouette Sample is 0.002, which means that rest all the values are positive. We can say that there are no records mapped incorrectly to any cluster.



From SC Score, the number of optimal clusters could be 3 or 4.

4.2) Solution of the clusters

Here are the few rows of the dataset:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Clus_kmeans	sil_width
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	0	0.432658
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	3	0.099543
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	0	0.425893
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	1	0.529852
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837	0	0.082791

4.2.1) 3 Cluster Solution

Cluster 3 labels are as follows:

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

Counts for the labels is:

```
0    72
2    71
1    67
Name: count, dtype: int64
```

Here are some of the Cluster information

cluster	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
1	18.5	16.2	0.9	6.2	3.7	3.6	6.0
2	11.9	13.2	0.8	5.2	2.8	4.7	5.1
3	14.4	14.3	0.9	5.5	3.3	2.7	5.1

cluster	Cluster_Size	Cluster_Percentage
1	67	31.90
2	72	34.29
3	71	33.81

Mean of different clusters and individual variable.

cluster	1	2	3
spending	18.5	11.9	14.4
advance_payments	16.2	13.2	14.3
probability_of_full_payment	0.9	0.8	0.9
current_balance	6.2	5.2	5.5
credit_limit	3.7	2.8	3.3
min_payment_amt	3.6	4.7	2.7
max_spent_in_single_shopping	6.0	5.1	5.1

4.2.2) 4 Cluster Solution

Cluster 4 labels are as follows:

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

Counts for the labels is:

```
2    68
3    61
1    51
0    30
```

Here are some of the Cluster information

cluster	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
1	11.8	13.2	0.8	5.2	2.8	5.0	5.1
2	16.3	15.3	0.9	5.9	3.4	3.9	5.7
3	14.0	14.1	0.9	5.4	3.2	2.6	5.0
4	19.1	16.5	0.9	6.3	3.8	3.5	6.1

cluster	Cluster_Size	Cluster_Percentage
1	64	30.48
2	30	14.29
3	67	31.90
4	49	23.33

Mean of different clusters and individual variable.

cluster	1	2	3	4
spending	11.8	16.3	14.0	19.1
advance_payments	13.2	15.3	14.1	16.5
probability_of_full_payment	0.8	0.9	0.9	0.9
current_balance	5.2	5.9	5.4	6.3
credit_limit	2.8	3.4	3.2	3.8
min_payment_amt	5.0	3.9	2.6	3.5
max_spent_in_single_shopping	5.1	5.7	5.0	6.1

4.2.3) 5 Cluster Solution

Here are some of the Cluster information

cluster	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
1	19.2	16.5	0.9	6.3	3.8	3.5	6.1
2	14.2	14.2	0.9	5.5	3.2	2.4	5.1
3	11.6	13.2	0.8	5.2	2.8	4.4	5.1
4	16.4	15.3	0.9	5.9	3.4	3.9	5.7
5	12.5	13.4	0.9	5.2	3.0	5.4	5.1

cluster	Cluster_Size	Cluster_Percentage
1	48	22.86
2	59	28.10
3	45	21.43
4	29	13.81
5	29	13.81

Mean of different clusters and individual variable.

cluster	1	2	3	4	5
spending	19.2	14.2	11.6	16.4	12.5
advance_payments	16.5	14.2	13.2	15.3	13.4
probability_of_full_payment	0.9	0.9	0.8	0.9	0.9
current_balance	6.3	5.5	5.2	5.9	5.2
credit_limit	3.8	3.2	2.8	3.4	3.0
min_payment_amt	3.5	2.4	4.4	3.9	5.4
max_spent_in_single_shopping	6.1	5.1	5.1	5.7	5.1

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

3 Group Clustering via K-Means

cluster	1	2	3
spending	18.5	11.9	14.4
advance_payments	16.2	13.2	14.3
probability_of_full_payment	0.9	0.8	0.9
current_balance	6.2	5.2	5.5
credit_limit	3.7	2.8	3.3
min_payment_amt	3.6	4.7	2.7
max_spent_in_single_shopping	6.0	5.1	5.1

3 Group Clustering via Hierarchical

clusters-3	1	2	3
spending	18.371429	11.872388	14.199041
advance_payments	16.145429	13.257015	14.233562
probability_of_full_payment	0.884400	0.848072	0.879190
current_balance	6.158171	5.238940	5.478233
credit_limit	3.684629	2.848537	3.226452
min_payment_amt	3.639157	4.949433	2.612181
max_spent_in_single_shopping	6.017371	5.122209	5.086178
Freq	70.000000	67.000000	73.000000

As we can see that in hierarchical clustering, customers under cluster 1 are higher spenders, cluster 3 mediocre and cluster 2 lowest.

In K-Means clustering, customers under cluster 2 are higher spenders, cluster 1 mediocre and cluster 2 lowest.

Business Recommendations:

Group 1 : High Spending Group

- The offerings such as, higher reward points on a higher probability can increase their spending capacity.
- Adding an option of no cost EMI as a promotional scheme with bank's tied up brands, can be a great motivator for this group.

- The segmentation of maximum max_spent_in_single_shopping is the highest of this group, hence, the discounts offered or attractive offers on the next transactions with full payments upfront.
- Periodic assessment and increase of credit limits
- The preferential customer treatment which might lead to higher spending habits
- Since there is a clear indication that the customers of this category are financially stable, interesting loan schemes exclusively for them could be planned.
- Collaborations with high end luxury brands and accessories would lead to higher one-time maximum spending.

Group 2 : Low Spending Group

- We can spend some time analying the brands and utilities this segment spends its most amount on and provide discounts and offers on the credit card usage accordingly.
- Customers of this segment will have to be given timely reminders on payments so that the due dates of the billing cycles are not missed.
- Small-scale campaigns could be run providing the customers of this segment attractive offers for early payments which would improve the rate of payment received and result in lesser default rates.

Group 3 : Medium Spending Group

- The customers of this segmentation cluster are suggested to be the target customers with highest potential as there is consistent maintenance of a higher credit score which results in timely payments of their bills.
- The customers of this category can have an increased credit limit raised and monitored periodically and have significantly marginalized interest rates keeping RBI guidelines in mind.
- The advertisement and promotion of premium cards or loyalty cards of specific brand collaborated partnerships would lead to increase in the transactional values over an extended period.
- Once the above-mentioned credit limits are enhanced, the result would be an automatic increase in spending habits across the premium partners in e-commerce, travel portals, airlines & hotels.

PROBLEM STATEMENT 2 - DECISION TREE, RANDOM FOREST, AND ARTIFICIAL NEURAL NETWORK

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.

Data Dictionary:

1. Target: Claim Status (Claimed)

2. Code of tour firm (Agency_Code)

3. Type of tour insurance firms (Type)

4. Distribution channel of tour insurance agencies (Channel)

5. Name of the tour insurance products (Product)

6. Duration of the tour (Duration in days)

7. Destination of the tour (Destination)

8. Amount worth of sales per customer in procuring tour insurance policies in rupees (in 100’s)

9. The commission received for tour insurance firm (Commission is in percentage of sales)

10.Age of insured (Age)

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

1.1) Checking the data

Here is the first few rows of the dataset:

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

Last few rows:

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
2990	51	EPX	Travel Agency	No	0.00	Online	2	20.00	Customised Plan	ASIA
2991	29	C2B	Airlines	Yes	48.30	Online	381	193.20	Silver Plan	ASIA
2992	28	CWT	Travel Agency	No	11.88	Online	389	19.80	Customised Plan	ASIA
2993	36	EPX	Travel Agency	No	0.00	Online	234	10.00	Cancellation Plan	ASIA
2994	27	C2B	Airlines	Yes	71.85	Online	416	287.40	Gold Plan	ASIA
2995	28	CWT	Travel Agency	Yes	166.53	Online	364	256.20	Gold Plan	Americas
2996	35	C2B	Airlines	No	13.50	Online	5	54.00	Gold Plan	ASIA
2997	36	EPX	Travel Agency	No	0.00	Online	54	28.00	Customised Plan	ASIA
2998	34	C2B	Airlines	Yes	7.64	Online	39	30.55	Bronze Plan	ASIA
2999	47	JZI	Airlines	No	11.55	Online	15	33.00	Bronze Plan	ASIA

Data looks good at the first glance

1.2) Information 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
```

Observations:

- There are a total of 10 variables in the dataset.
- Age, Commision, Duration, Sales are the numeric variable and rest all are categorial variables.
- There is a total of 3000 records and no missing values as of initial analysis.
- There is a total of nine independent variable and one target variable wiz. Claimed.
- We can see that there are no null values in the data. 2 of the 10 variables are of Data type Float, 2 variables are of Data type integer and the remaining 6 are of Object Data type.

1.3) Check for any missing value in any column

There is no missing value present in the dataset.

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

1.4) Descriptive Statistics Summary

	count	mean	std	min	25%	50%	75%	90%	max
Age	3000.0	38.091000	10.463518	8.0	32.0	36.00	42.000	53.000	84.00
Commision	3000.0	14.529203	25.481455	0.0	0.0	4.63	17.235	48.300	210.21
Duration	3000.0	70.001333	134.053313	-1.0	11.0	26.50	63.000	224.200	4580.00
Sales	3000.0	60.249913	70.733954	0.0	20.0	33.00	69.000	172.025	539.00

Duration has a negative value wiz -1 , it is not possible. That means that the entry is a wrong entry.

Commision and Sales varies significantly in mean and median.

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

For Object data type variables like, Agency_Code, Type, Claimed, Channel, Product Name, and Destination, there are very less unique values and 50% is the median.

The topmost frequent value of:

- Agency_code is EPX with a frequency of 1365
- Type is Travel Agency with a frequency of 1837
- Claimed is No with a frequency of 2076
- Channel is Online with a frequency of 2954
- Product Name is Customised Plan with a frequency of 1136
- Destination is ASIA with a frequency of 2465

For the float and integers data type values like: Age, Commision, Duration and Sales the difference between its 75th percentile and Max value is very large, indicating there will be large number of outliers in the data.

Age of customers ranges from 8 to 84 who are insured, with an average of 39.

Commission and Sales variables have 0 as minimum values.

The shape of the data is (3000, 10).

1.5) Getting unique counts of all Nominal Variables

```
AGENCY_CODE :  4
Agency_Code
JZI     239
CWT     472
C2B     924
EPX    1365
Name: count, dtype: int64


TYPE :  2
Type
Airlines         1163
Travel Agency    1837
Name: count, dtype: int64


  CLAIMED :  2
  Claimed
  Yes     924
  No     2076
  Name: count, dtype: int64


  CHANNEL :  2
  Channel
  Offline      46
  Online     2954
  Name: count, dtype: int64


  PRODUCT NAME :  5
  Product Name
  Gold Plan             109
  Silver Plan           427
  Bronze Plan           650
  Cancellation Plan     678
  Customised Plan      1136
  Name: count, dtype: int64


  DESTINATION :  3
  Destination
  EUROPE       215
  Americas     320
  ASIA        2465
  Name: count, dtype: int64
```

We analyzed the Object data type variables for their count of unique values.

1.6) Check for duplicate data

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
63	30	C2B	Airlines	Yes	15.0	Online	27	60.0	Bronze Plan	ASIA
329	36	EPX	Travel Agency	No	0.0	Online	5	20.0	Customised Plan	ASIA
407	36	EPX	Travel Agency	No	0.0	Online	11	19.0	Cancellation Plan	ASIA
411	35	EPX	Travel Agency	No	0.0	Online	2	20.0	Customised Plan	ASIA
422	36	EPX	Travel Agency	No	0.0	Online	5	20.0	Customised Plan	ASIA
...	...	...	...	...	...	...	...	...	...	...
2940	36	EPX	Travel Agency	No	0.0	Online	8	10.0	Cancellation Plan	ASIA
2947	36	EPX	Travel Agency	No	0.0	Online	10	28.0	Customised Plan	ASIA
2952	36	EPX	Travel Agency	No	0.0	Online	2	10.0	Cancellation Plan	ASIA
2962	36	EPX	Travel Agency	No	0.0	Online	4	20.0	Customised Plan	ASIA
2984	36	EPX	Travel Agency	No	0.0	Online	1	20.0	Customised Plan	ASIA

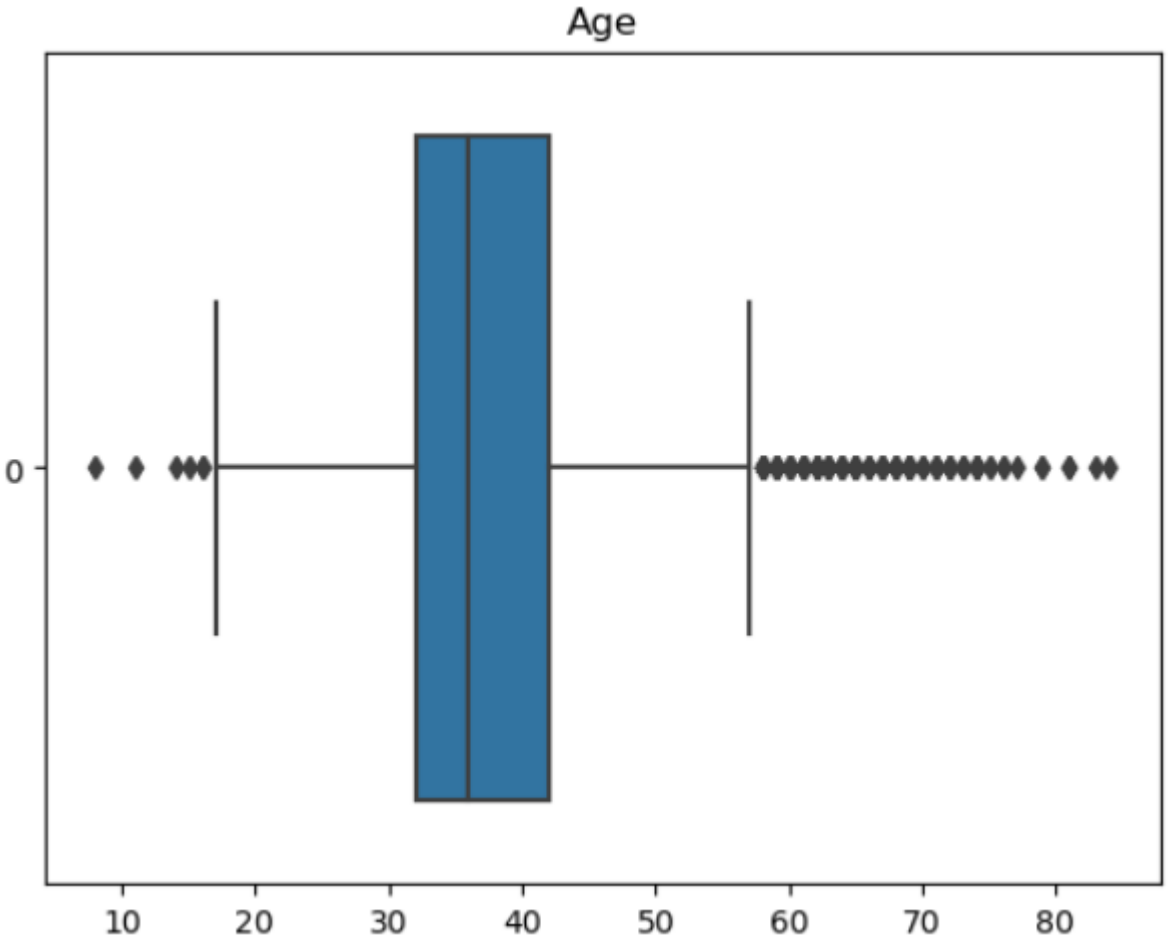
Total number of duplicate rows are 139.

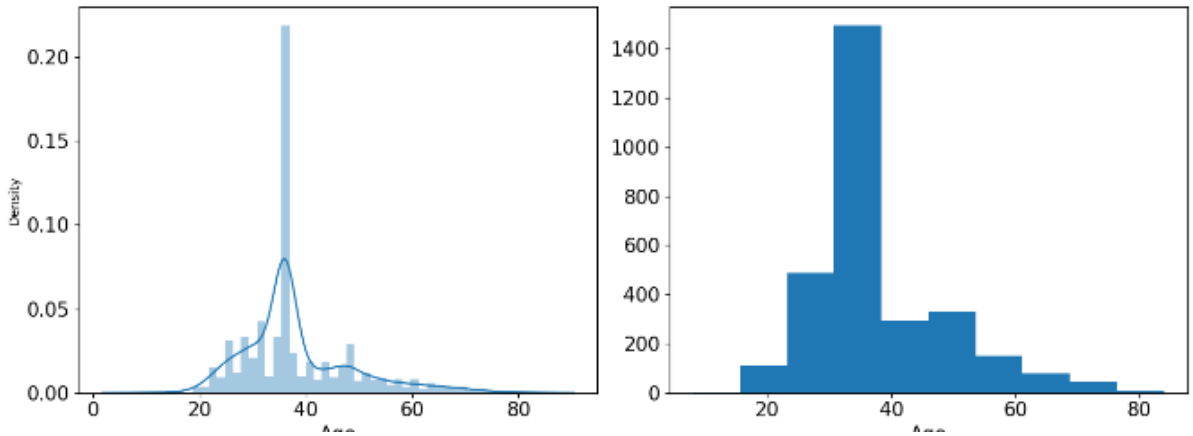
Though it shows there are 139 records, but it can be of different customers, there is no customer ID or any other identifier, so I am not dropping them off.

1.7) Univariate Analysis – Numerical Variables

1.7.1) Age

```
Minimum Age:  8
Maximum Age:  84
Mean value:  38.091
Median value:  36.0
Standard deviation:  10.463518245377944
Null values:  False


Number of outliers in Age upper :  198
Number of outliers in Age lower :  6
% of Outlier in Age upper:  7 %
% of Outlier in Age lower:  0 %
```

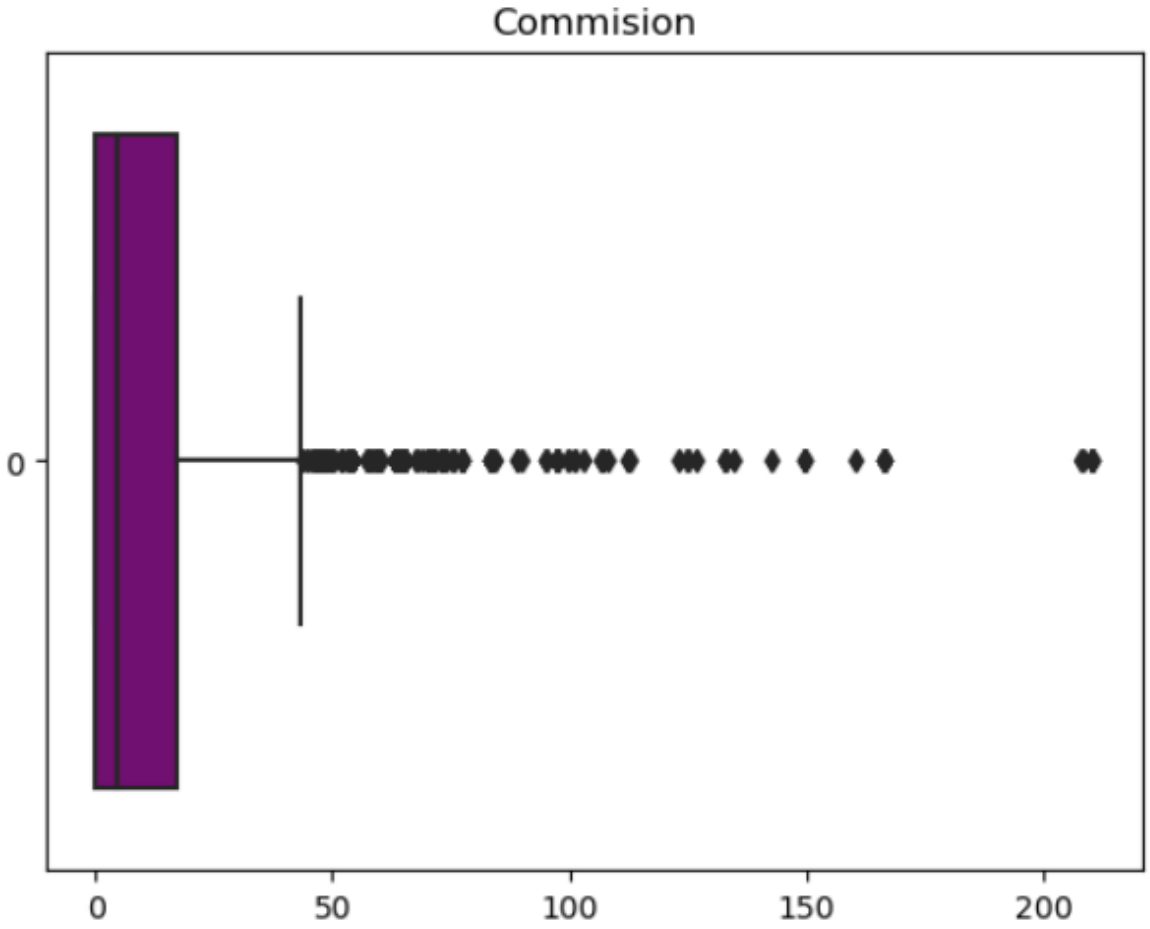


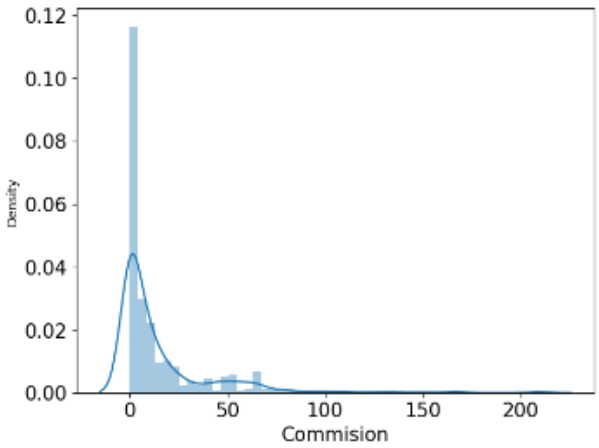


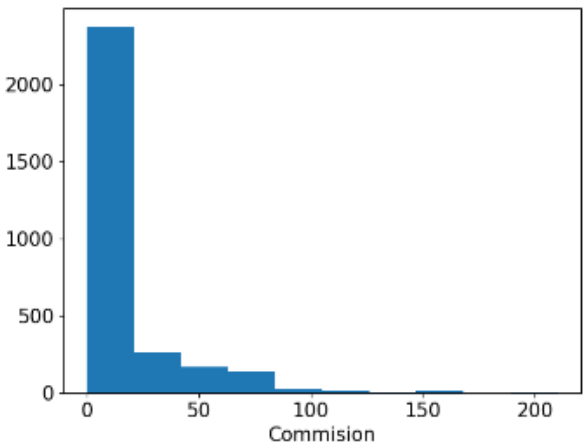
1.7.2) Commision

```
Minimum Commision:  0.0
Maximum Commision:  210.21
Mean value:  14.529203333333333
Median value:  4.63
Standard deviation:  25.48145450662553
Null values:  False
```

```
Number of outliers in Commision upper :  362
Number of outliers in Commision lower :  0
% of Outlier in Commision upper:  12 %
% of Outlier in Commision lower:  0 %
```

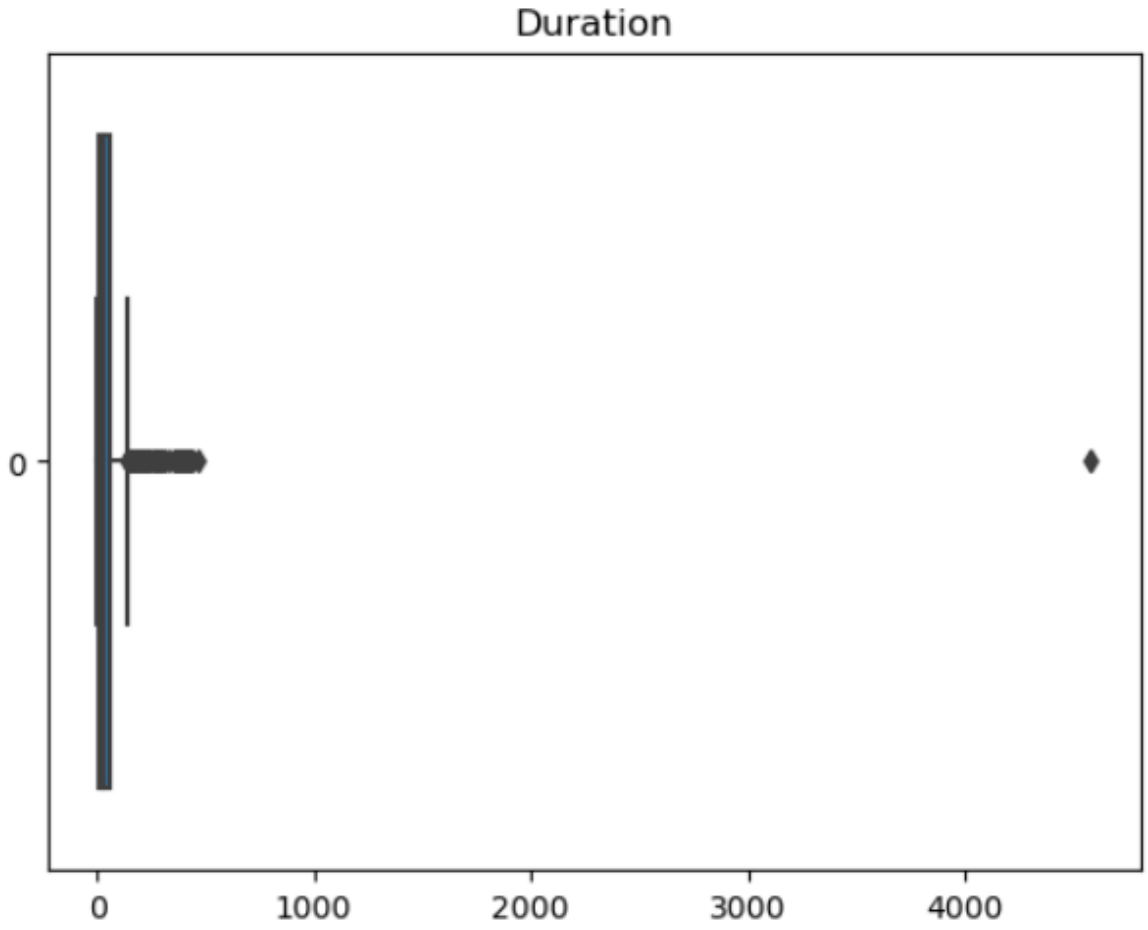


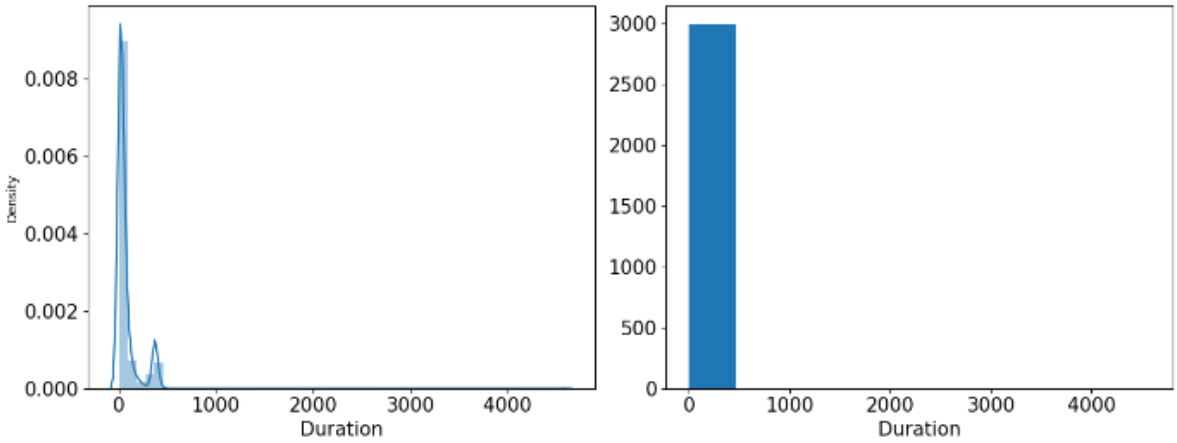




1.7.3) Duration

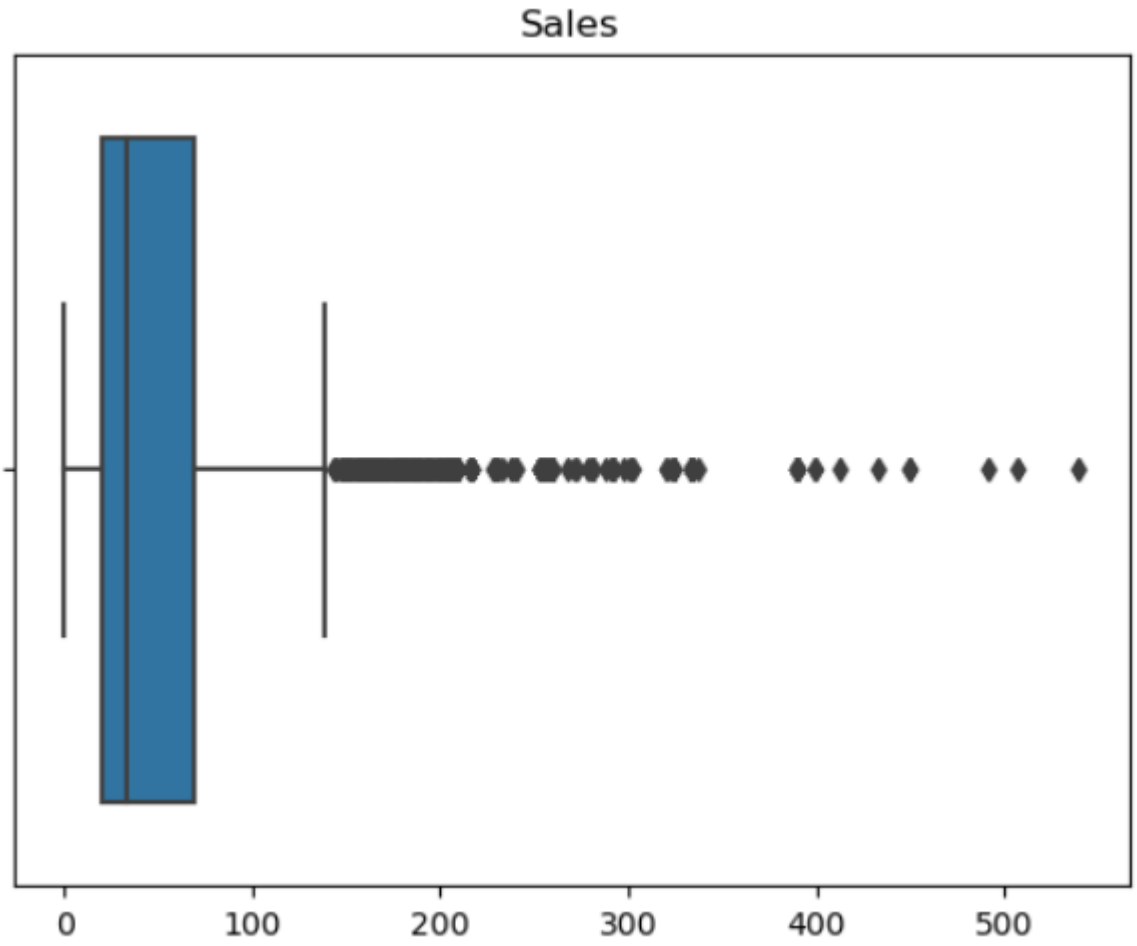
```
Minimum Duration:  -1
Maximum Duration:  4580
Mean value:  70.00133333333333
Median value:  26.5
Standard deviation:  134.05331313253495
Null values:  False

Number of outliers in Duration upper :  382
Number of outliers in Duration lower :  0
% of Outlier in Duration upper:  13 %
% of Outlier in Duration lower:  0 %
```

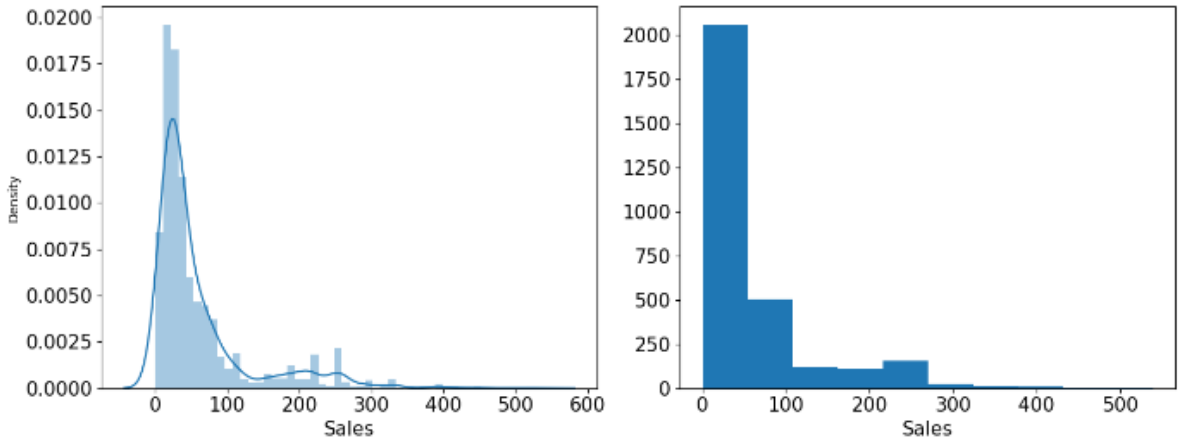




1.7.4) Sales

```
Minimum Sales:  0.0
Maximum Sales:  539.0
Mean value:  60.24991333333333
Median value:  33.0
Standard deviation:  70.73395353143047
Null values:  False

Number of outliers in Sales upper :  353
Number of outliers in Sales lower :  0
% of Outlier in Sales upper:  12 %
% of Outlier in Sales lower:  0 %
```





Observations for all:

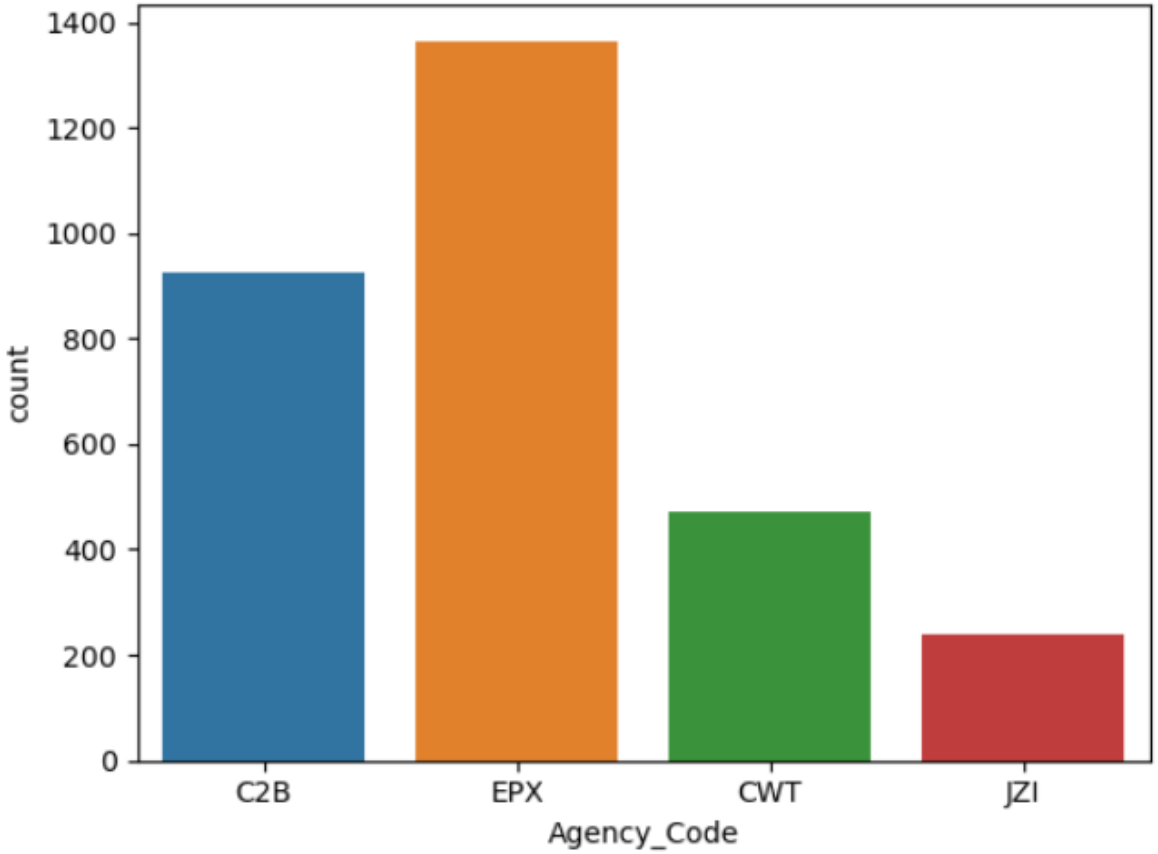
From the boxplot we can see that there are outliers present in the dataset, but there is no need to treat them since they are not going to affect the prediction models.

From the boxplot, distribution for all the variables is positively skewed.

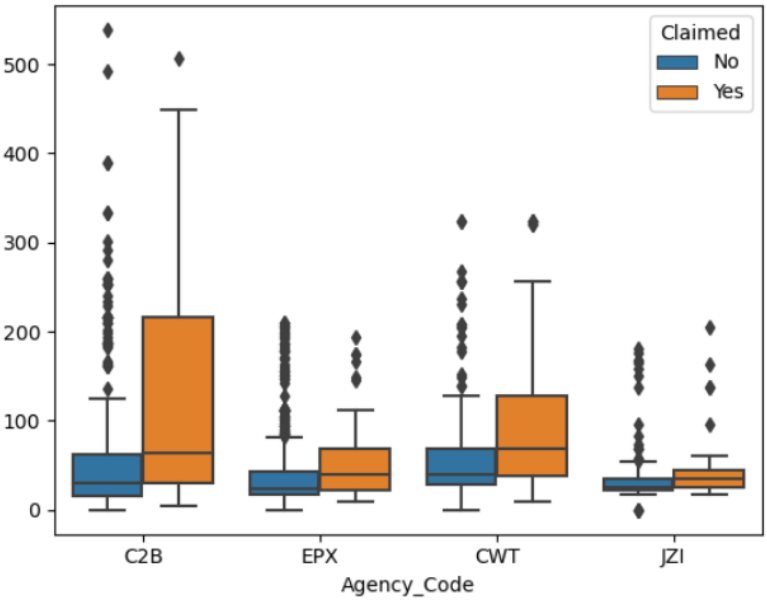
1.8) Univariate Analysis – Categorical Variables

1.8.1) Agency_Code

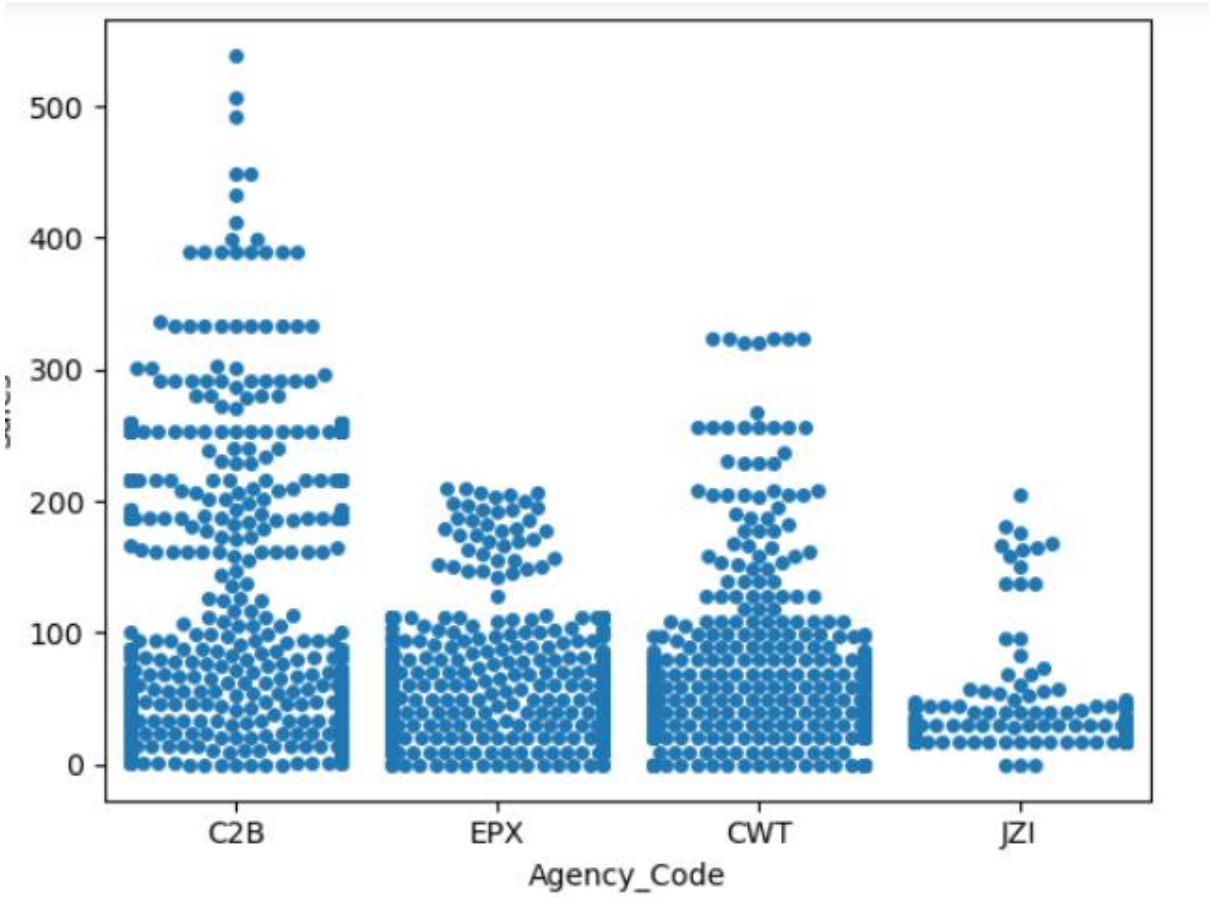
Count plot



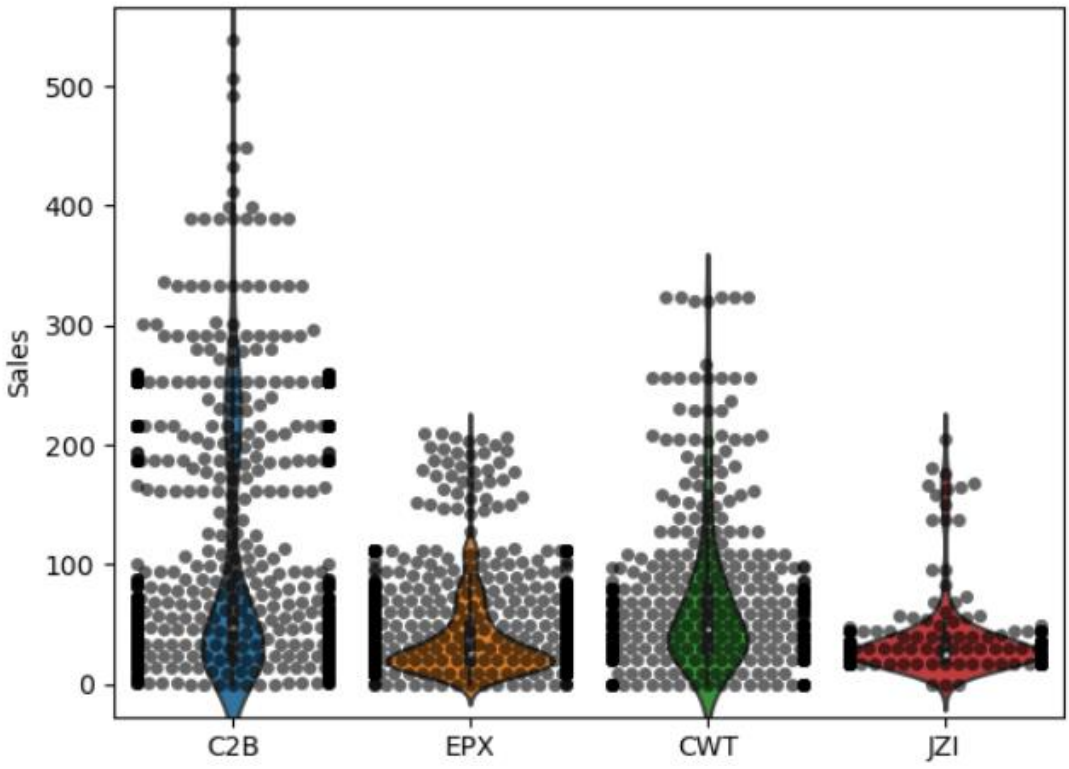
Box plot



Swarmplot

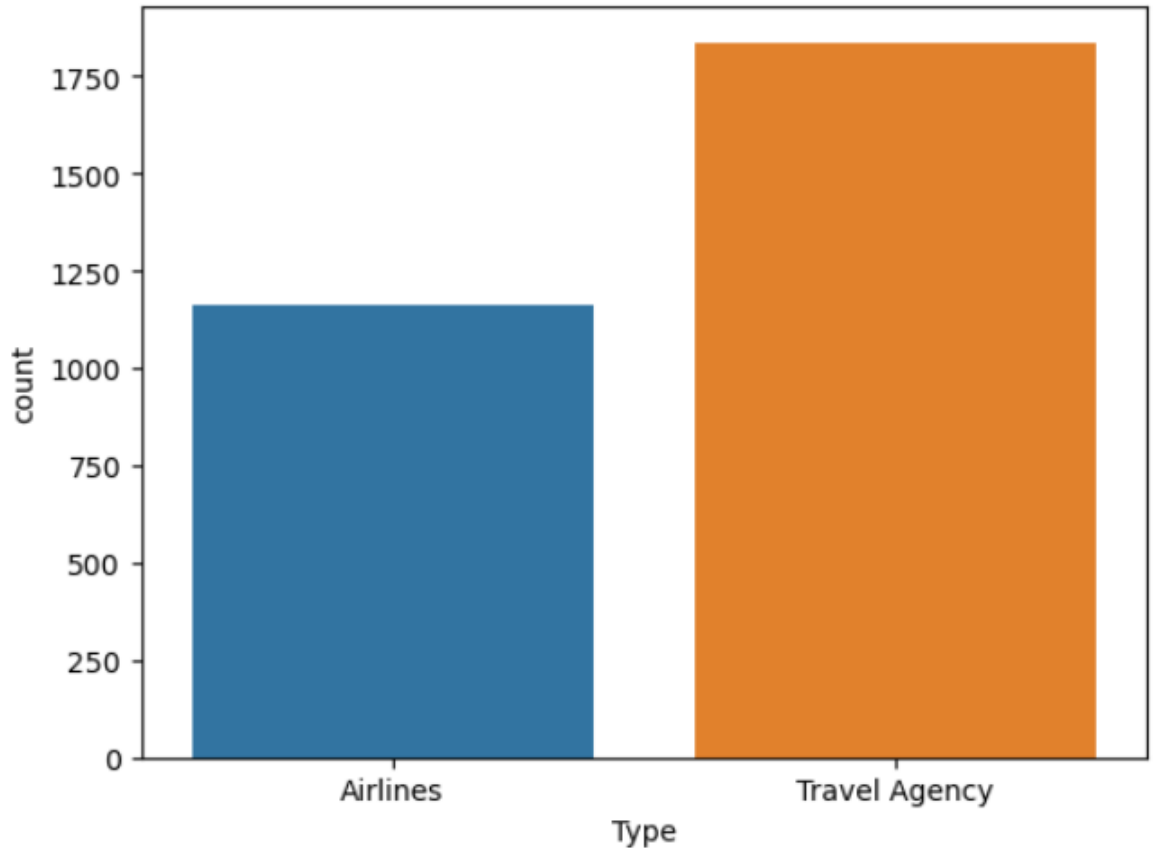


Violin and Swarm plot

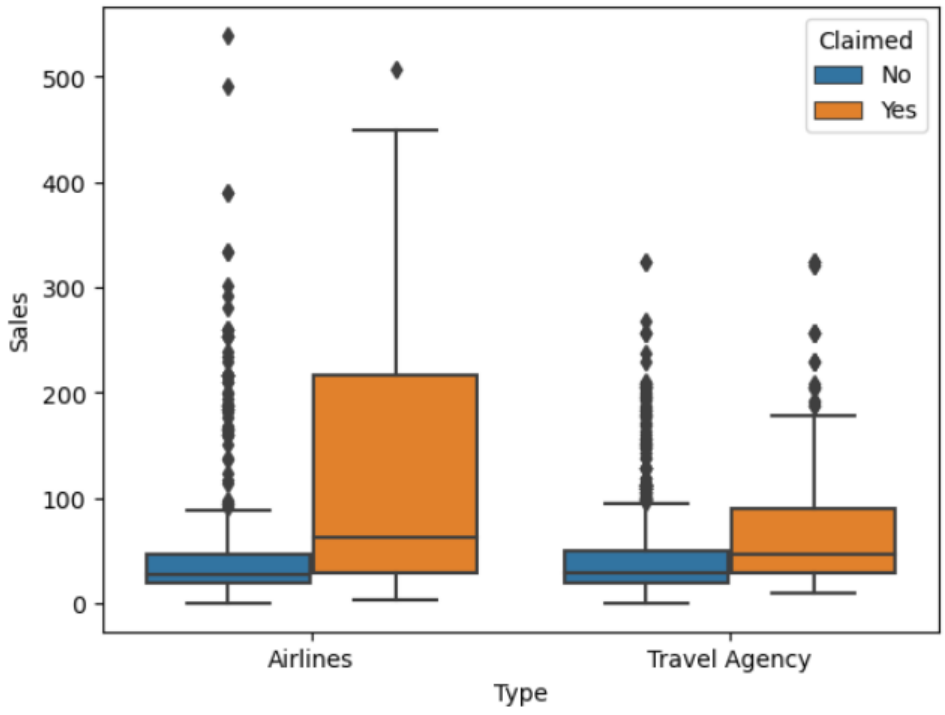


1.8.2) Type

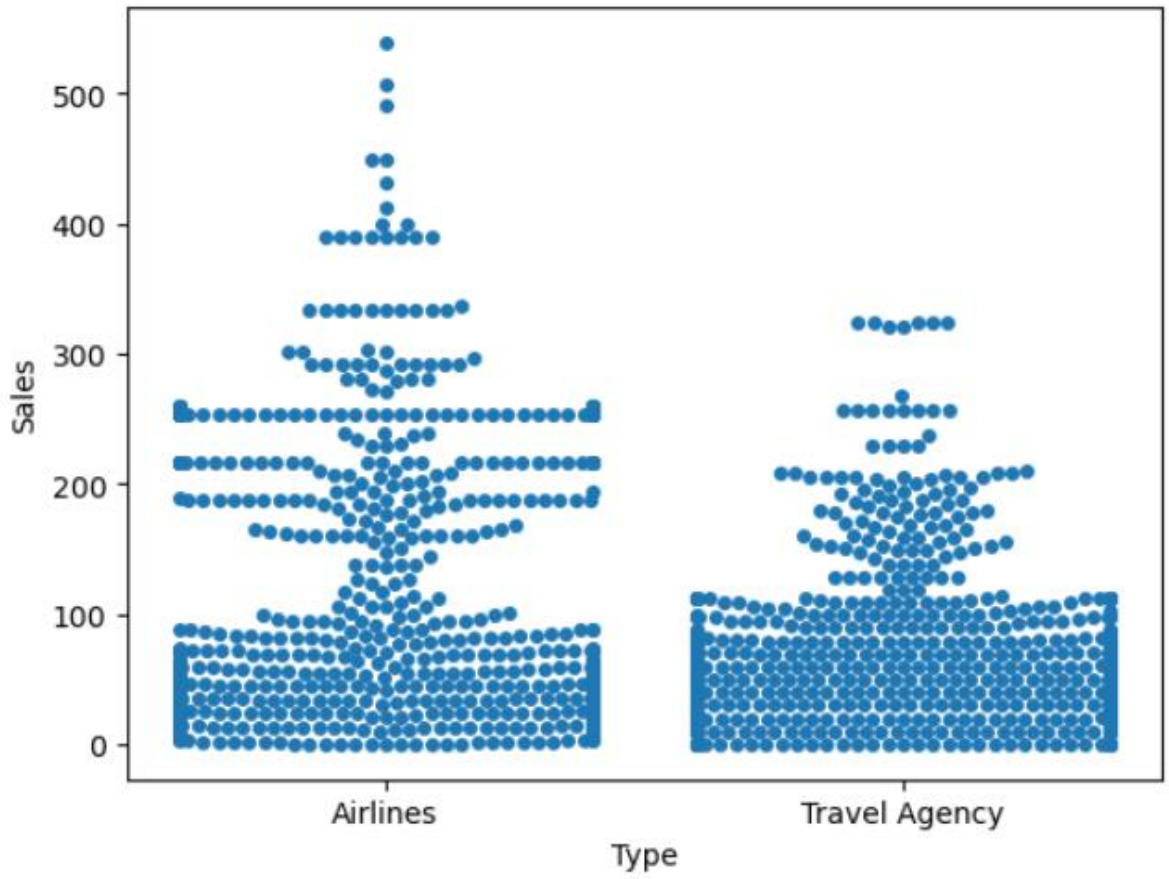
Count plot



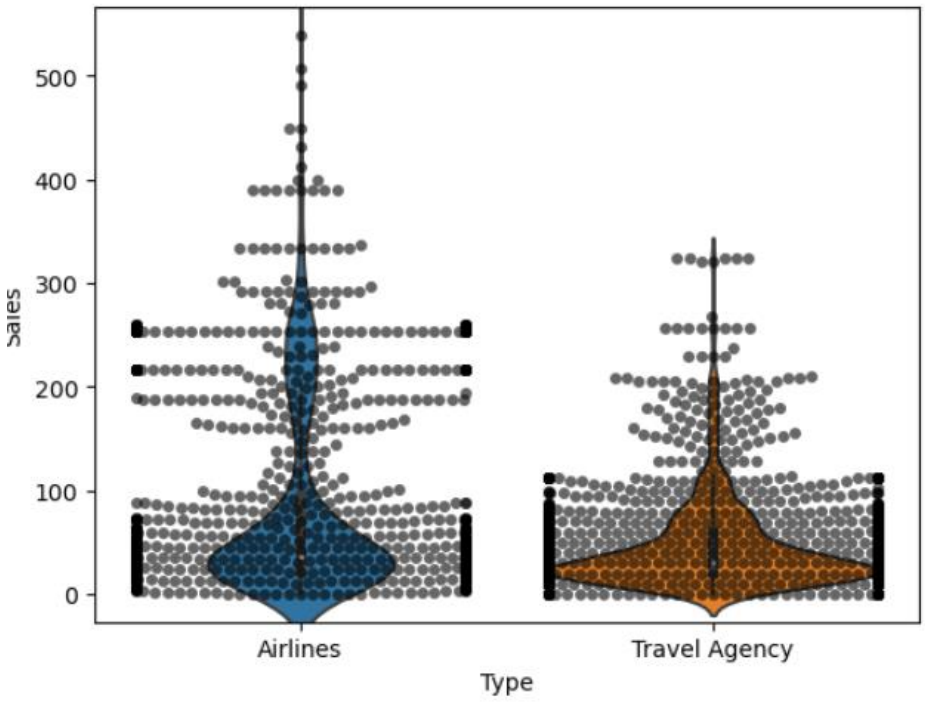
Box plot



Swarm plot

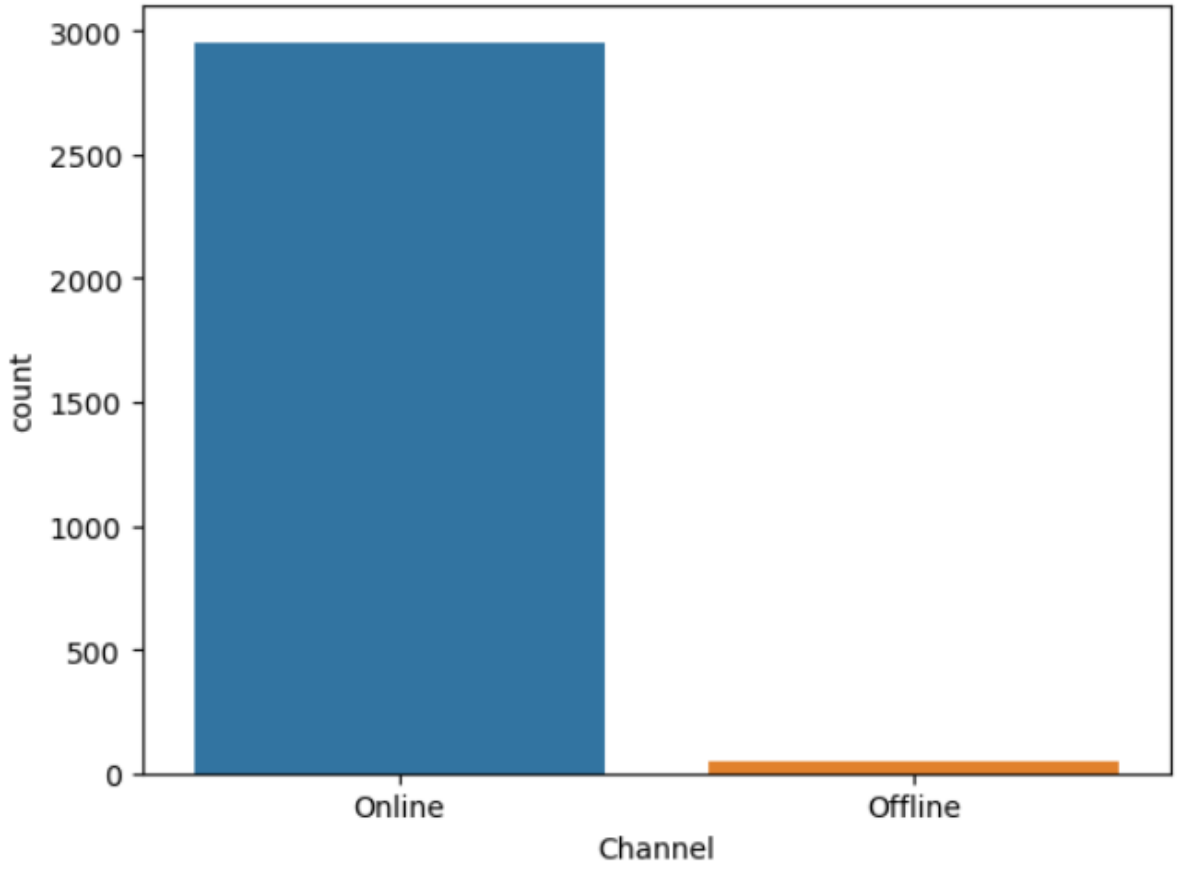


Violin and Swarm plot

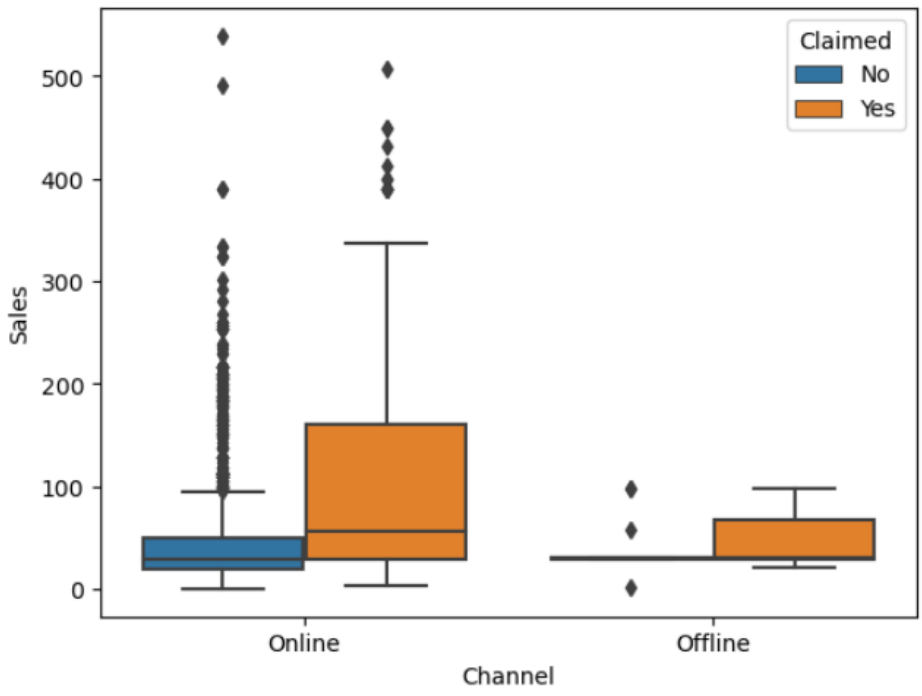


1.8.3) Channel

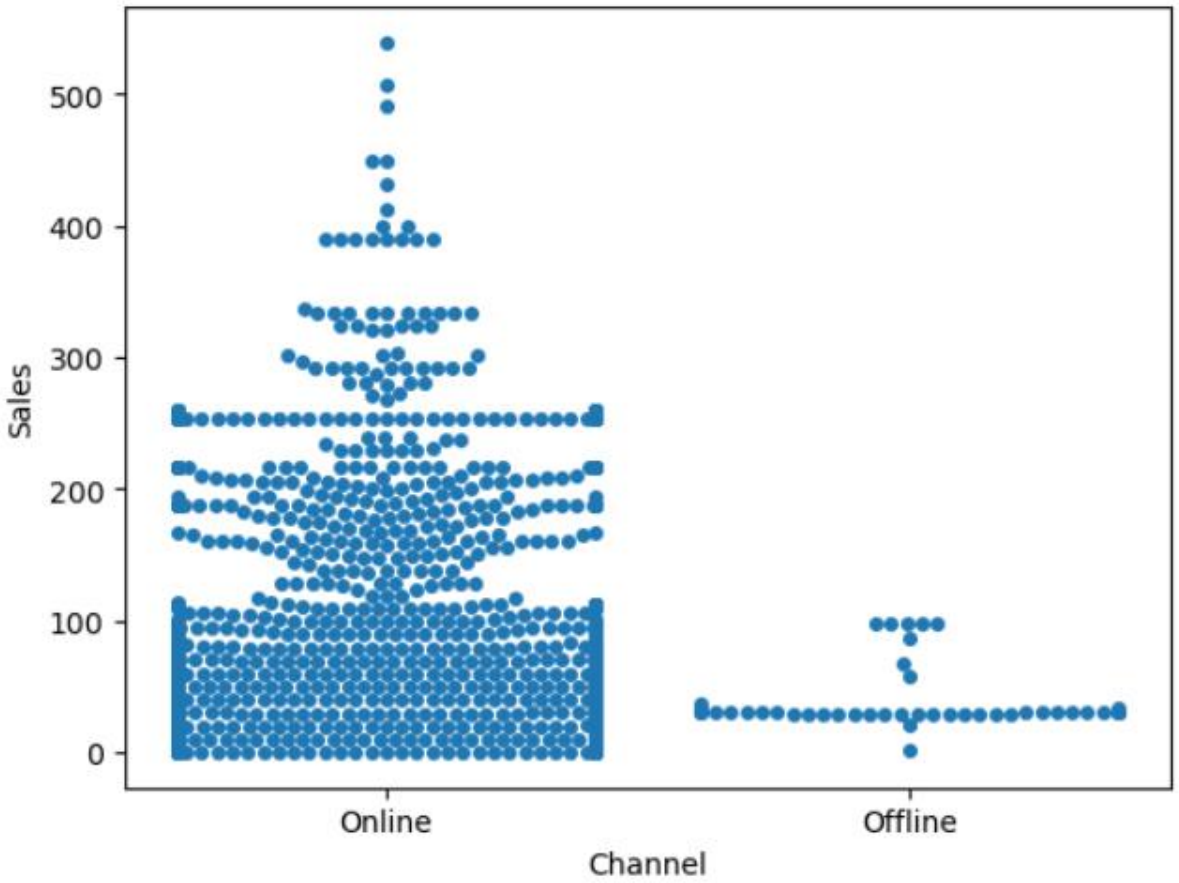
Count plot



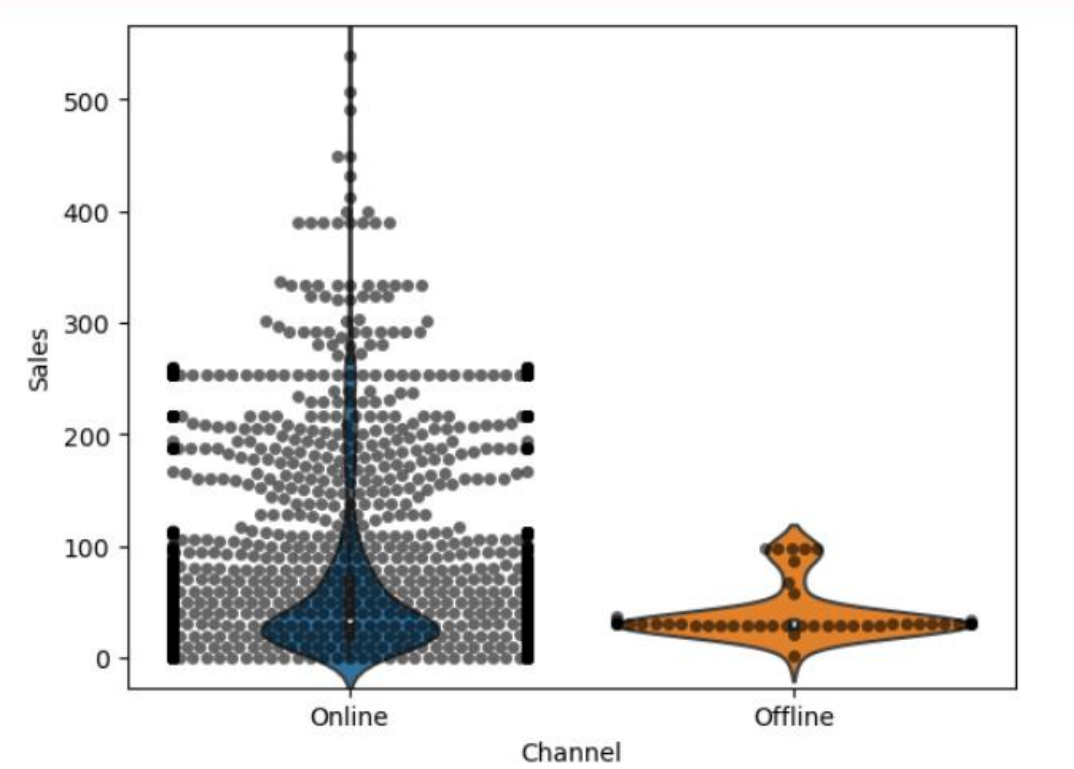
Box plot



Swarm plot

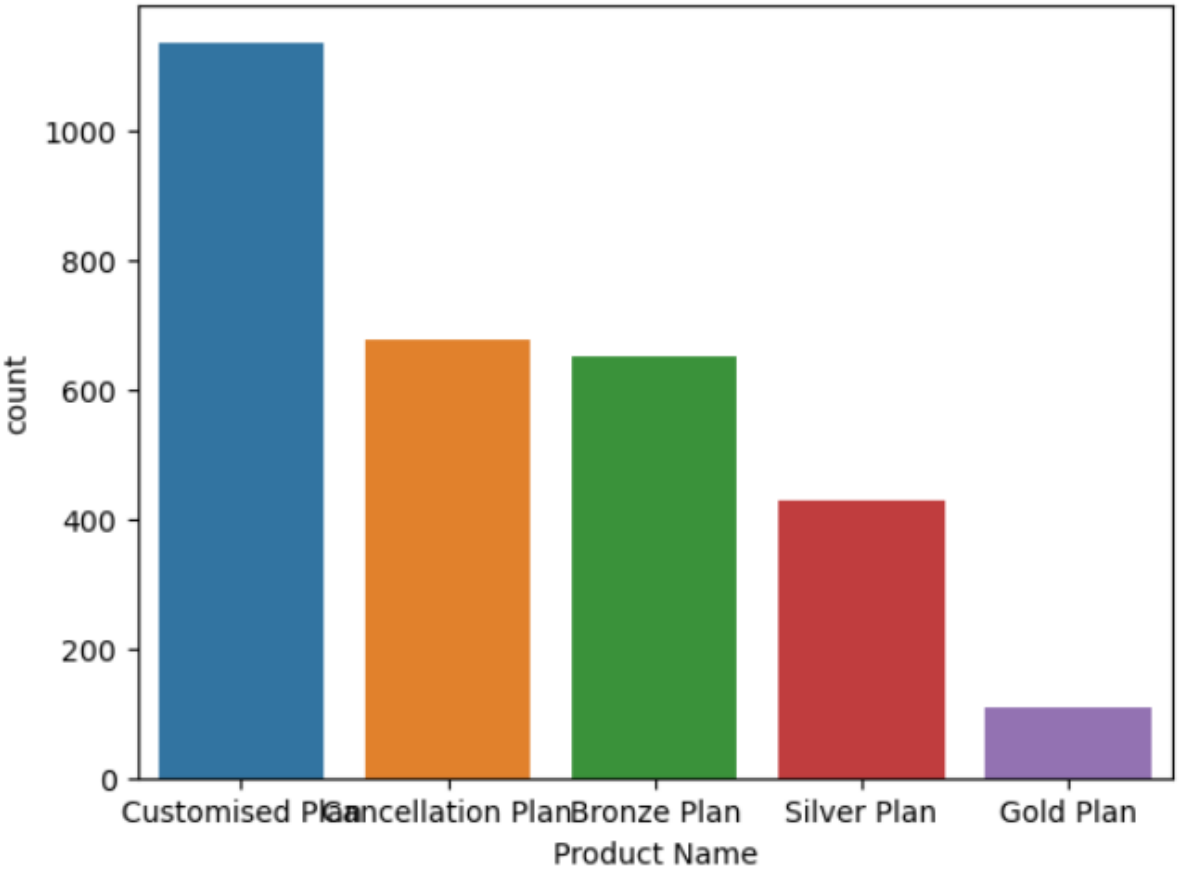


Violin and Swarm plot

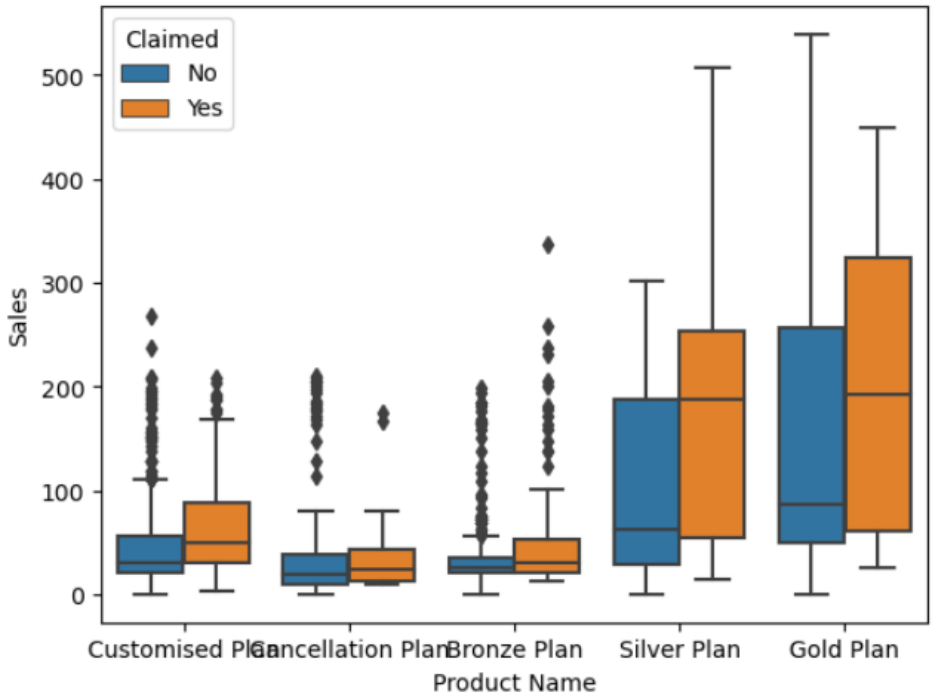


1.8.4) Product Name

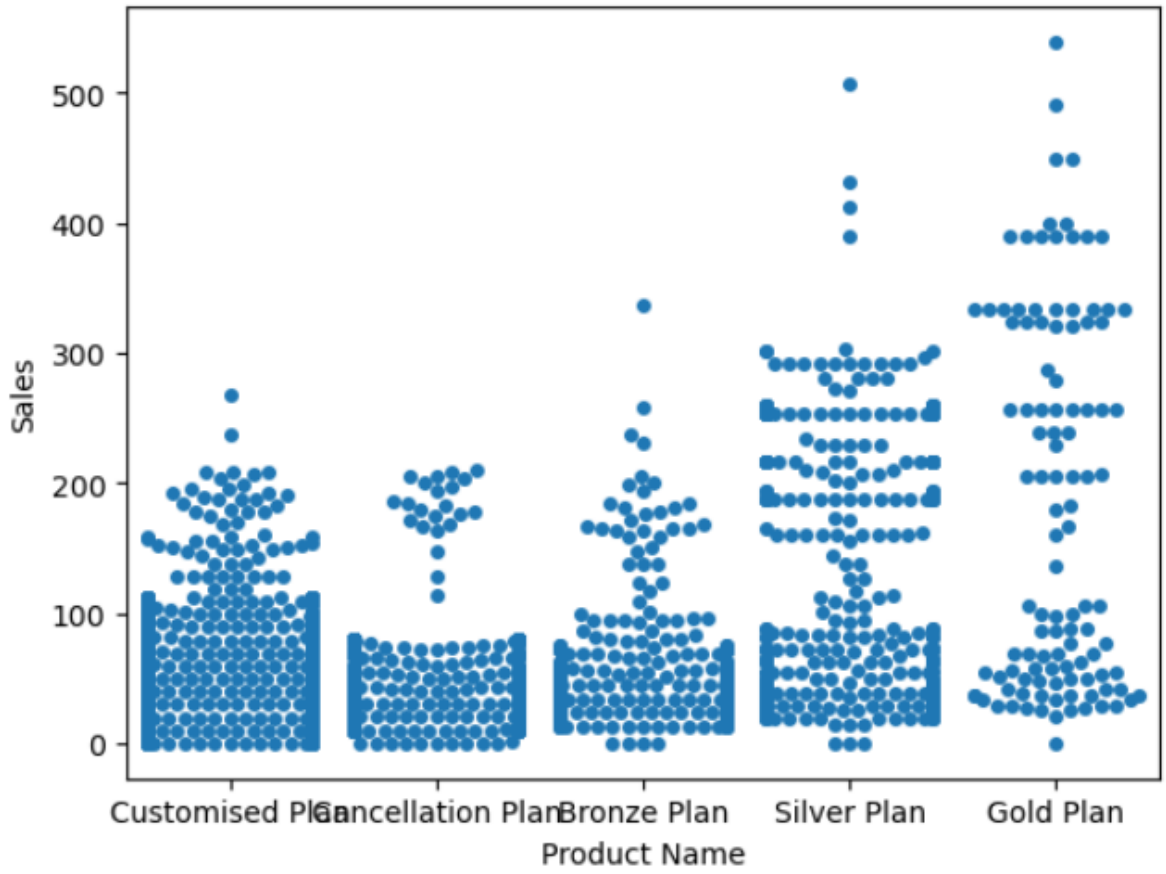
Count plot



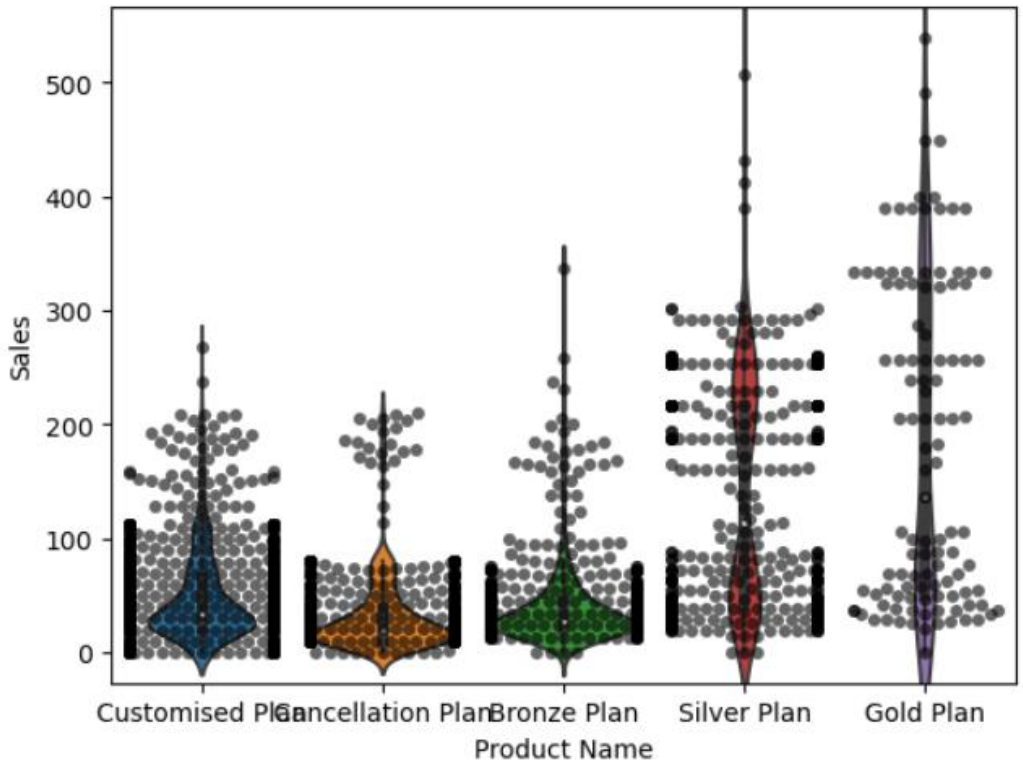
Box plot



Swarm plot

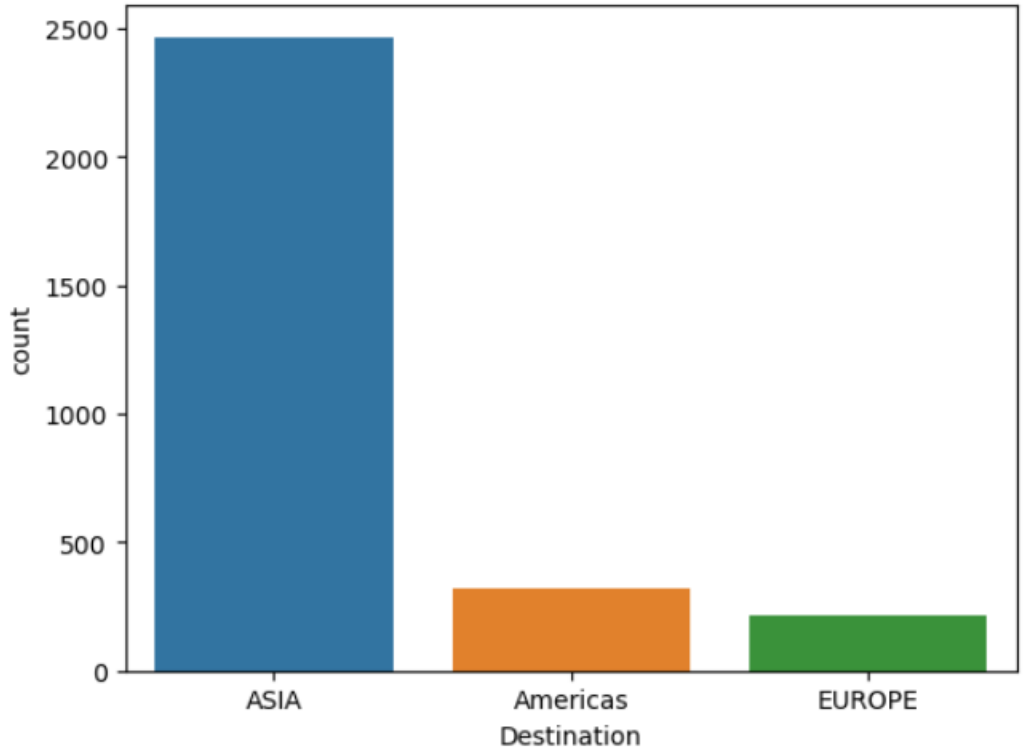


Violin and Swarm plot

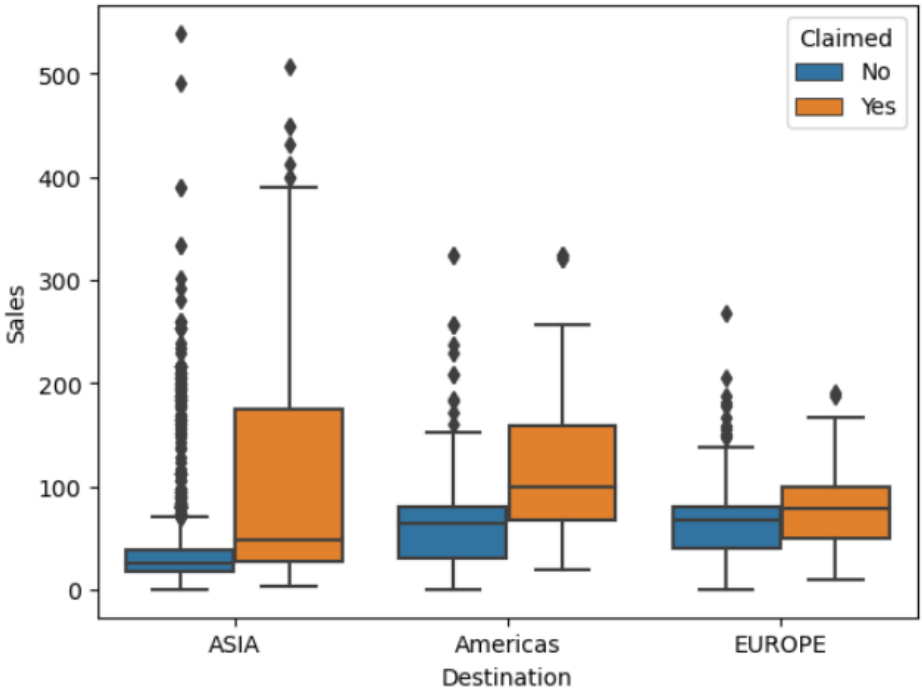


1.8.5) Destination

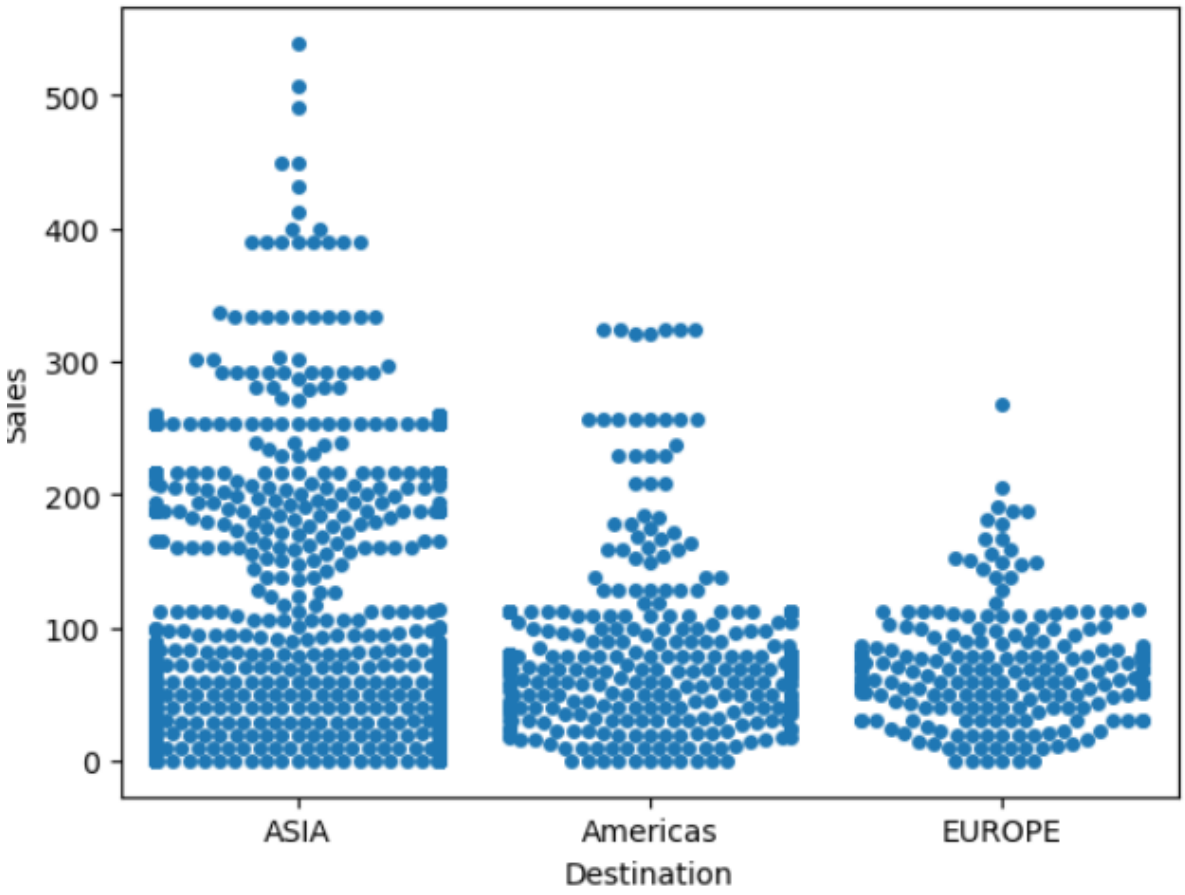
Count plot



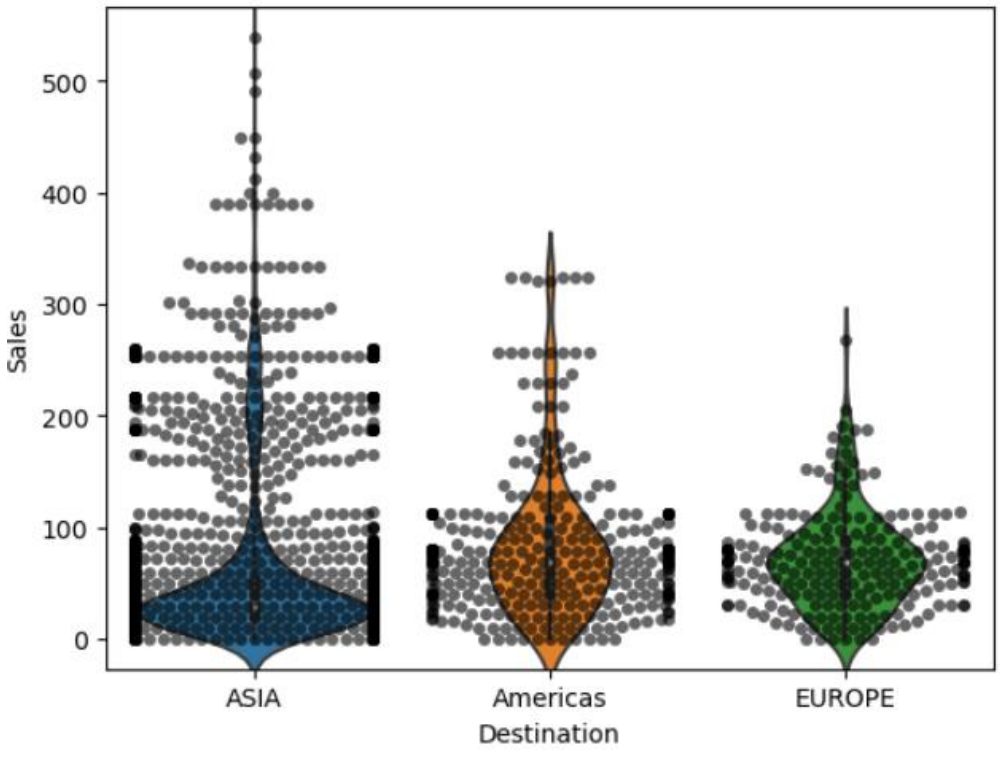
Box plot



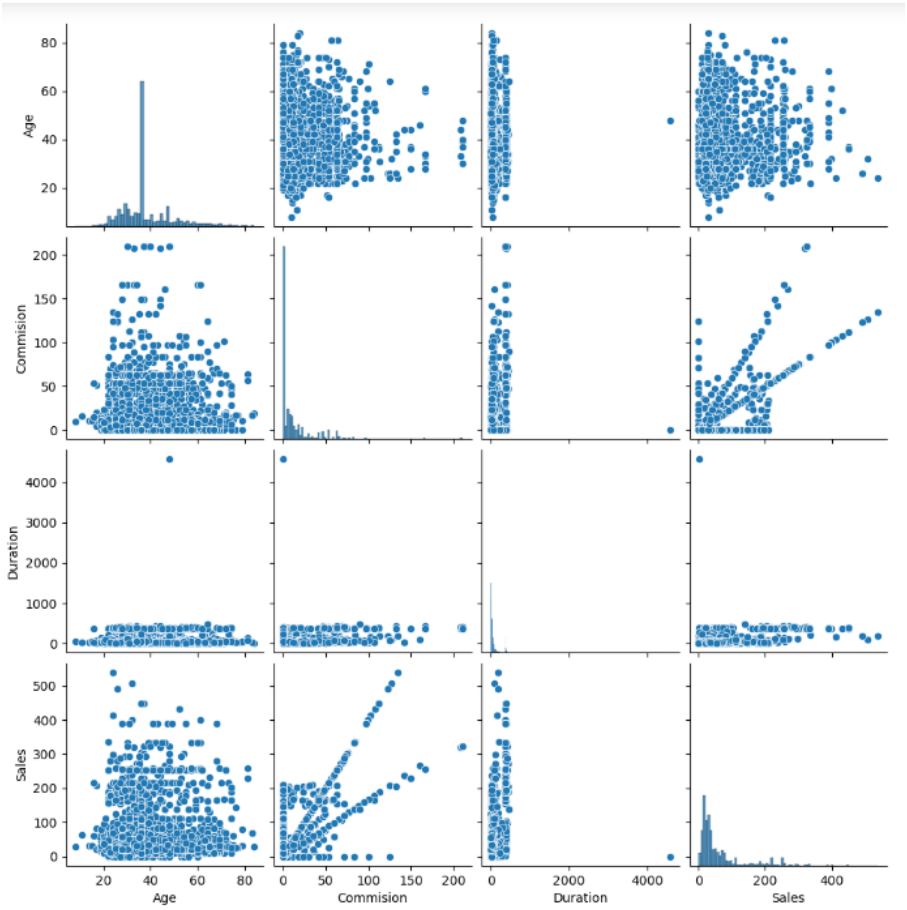
Swarm plot



Violin and Swarm plot



1.9) Multivariate Analysis

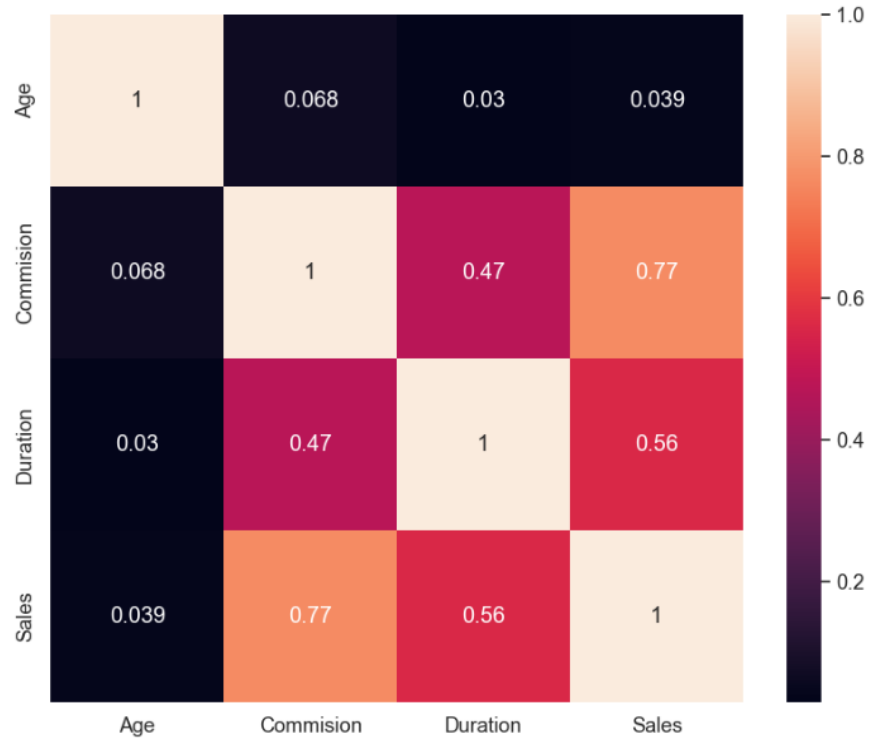


We can only find an interpretable relationship between Sales and Commision. Commision is increasing with the increase in Sales.

Rest of the variables don't seem to have definite patterns between them to make inferences on.

1.10) Correlation

There is a moderately good correlation among Duration, Sales and Commision. We can infer that as the travel duration increases the sales amount of insurance policies also increases, hence the higher percentage of commission per sale.



1.11) Converting all objects to categorical codes

For prediction models the data to pass should be in numeric/ categorical format only. The object variables in our dataset need to be converted to integer format.

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


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

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


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

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


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

Here is the information of the dataset after converting all objects to categorical codes

```
<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   int8   
 2   Type           3000 non-null   int8   
 3   Claimed        3000 non-null   int8   
 4   Commision      3000 non-null   float64
 5   Channel        3000 non-null   int8   
 6   Duration       3000 non-null   int64  
 7   Sales          3000 non-null   float64
 8   Product Name   3000 non-null   int8   
 9   Destination    3000 non-null   int8   
dtypes: float64(2), int64(2), int8(6)
memory usage: 111.5 KB
```

All the object variables got separated into different variables with datatype int8 and the number of variables remains the same in the dataset.

Here is the dataset after modification:

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

The dependent variable ‘Claimed’ has 2 categorical values ‘No’ (69.2%) and ‘Yes’ (30.8%). The data seems to be well balanced.

```
Claimed
0    0.692
1    0.308
Name: proportion, dtype: float64
```

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

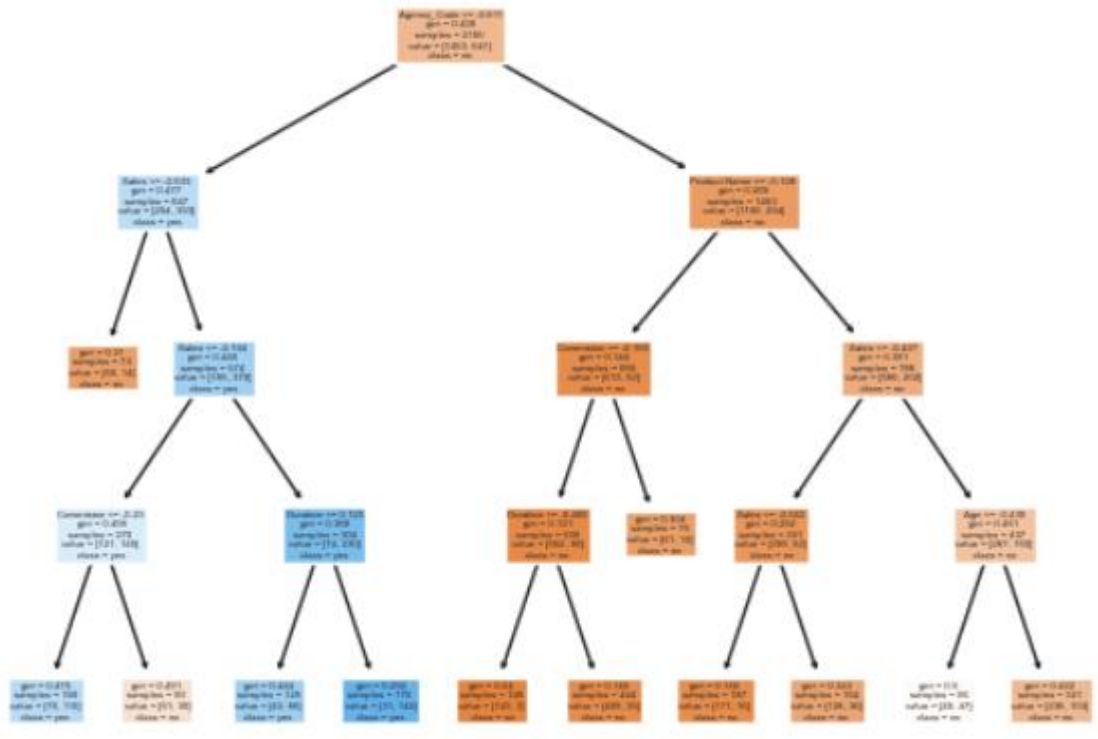
Firstly, splitting the data into Train and Test data.

Below is the data shape after dropping the column Claimed:

```
X_train (2100, 9)
X_test (900, 9)
train_labels (2100,)
test_labels (900,)
```

2.1) Decision Tree Classifier

CART is a Binary Decision Tree model. I have used Gini Index as its Criteria. It is an attribute that Maximizes the reduction in impurity is chosen as the Splitting Attribute.



In the first instance, we will allow the decision tree to be completely built using default parameters; Criterion = 'gini'.

We have performed Gridsearch crossvalidation for this model, by pasing multiple combination of values for the parameters, to find out the best parameters to build a model that performs well.

max_depth- The maximum depth of the tree.

min_samples_split- The minimum number of samples required to split an internal node.

min_samples_leaf- The minimum number of samples required to be at a leaf node

criterion- The function to measure he quality of a split.

The best max_depth is 4, min_samples_leaf is 44 and min_samples_split is 26

Here is the Importance of each variable:

	Imp
Agency_Code	0.634112
Sales	0.220899
Product Name	0.086632
Commision	0.021881
Age	0.019940
Duration	0.016536
Type	0.000000
Channel	0.000000
Destination	0.000000

Getting the predicted classes and probs

	0	1
0	0.697947	0.302053
1	0.979452	0.020548
2	0.921171	0.078829
3	0.510417	0.489583
4	0.921171	0.078829

2.2) Random Forest Classifier

We performed GridSearch crossvalidation for this model, by passing multiple combination of values for the parameters, to find out the best parameters to build a model that performs well.

n_estimators- The number of trees in the forest.

criterion- The function to measure the quality of a split

max_depth – the maximum depth of the tree

min_samples_split – The minimum number of samples required to split an internal node.

min_samples_leaf - The minimum number of samples required to be at a leaf node.

max_features – The number of features to consider when looking for the best split

The best max_depth is 6, min_samples_leaf is 8, n_estimators is 350, max_features is 3 and min_samples_split is 46.

Here is the importance of each variable:

	Imp
Agency_Code	0.276015
Product Name	0.235583
Sales	0.152733
Commision	0.135997
Duration	0.077475
Type	0.071019
Age	0.039503
Destination	0.008971
Channel	0.002705

Getting the predicted classes and probs

	0	1
0	0.778010	0.221990
1	0.971910	0.028090
2	0.904401	0.095599
3	0.651398	0.348602
4	0.868406	0.131594

2.3) Artificial Neural Network

We performed GridSearch crossvalidation for this model.

hidden_layer_sizes – The ith element represents the number of neurons in the ith hidden layer

activation – Activation function for the hidden layer.

solver – The solver for weight optimization

max_iter – Maximum number of iterations. The solver iterates until convergence or this number of iterations. For stochastic solver, note that this determines the number of epochs, not the number of gradient steps.

tol – Tolerance for the optimization

The best hidden_layer_sizes is 200, max_iter is 2500, random_state is 1, tol is 0.01.

Getting the predicted classes and probs

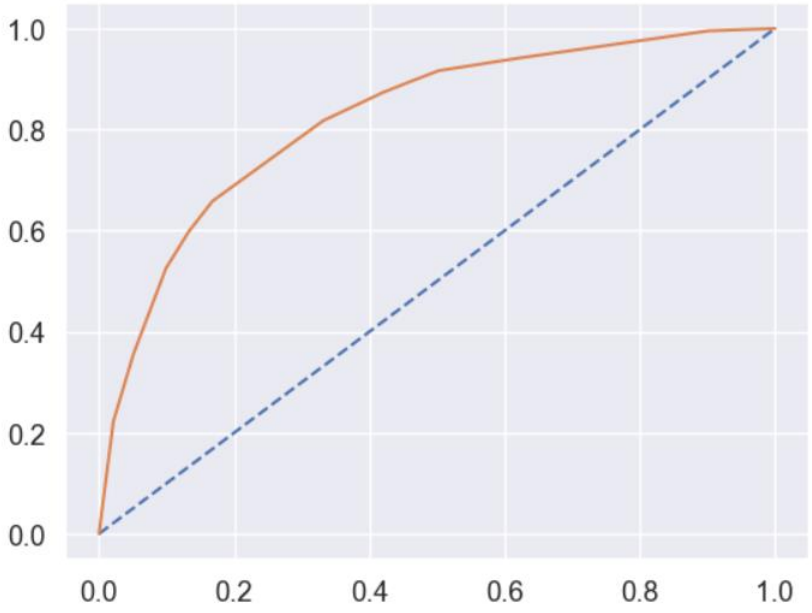
	0	1
0	0.822676	0.177324
1	0.933407	0.066593
2	0.918772	0.081228
3	0.688933	0.311067
4	0.913425	0.086575

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.

3.1) CART

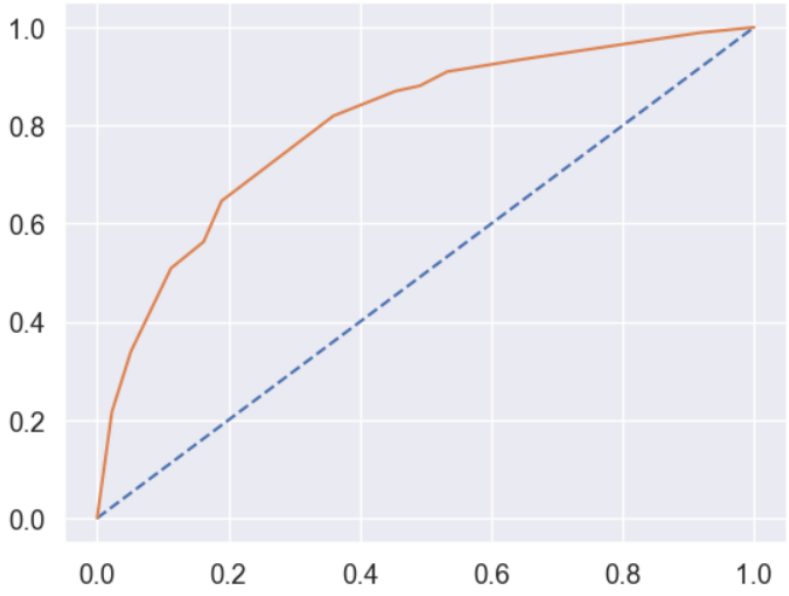
3.1.1) AUC and ROC for the training data

AUC is 0.823



3.1.2) AUC and ROC for the test data

AUC is 0.801



3.1.3) Confusion Matrix and Classification Report for the training data

Confusion matrix:

```
array([[1309,  144],
       [ 307,  340]], dtype=int64)
```

Classification Report:

	precision	recall	f1-score	support
0	0.81	0.90	0.85	1453
1	0.70	0.53	0.60	647
accuracy			0.79	2100
macro avg	0.76	0.71	0.73	2100
weighted avg	0.78	0.79	0.78	2100

3.1.4) Confusion Matrix and Classification Report for the testing data

Confusion Matrix:

```
array([[553,  70],
       [136, 141]], dtype=int64)
```

Classification Report:

	precision	recall	f1-score	support
0	0.80	0.89	0.84	623
1	0.67	0.51	0.58	277
accuracy			0.77	900
macro avg	0.74	0.70	0.71	900
weighted avg	0.76	0.77	0.76	900

3.1.5) CART Conclustion

Train Data:

- AUC – 82%
- Accuracy – 79%
- Precision – 70%
- f1-score – 60%

Test Data:

- AUC – 80%
- Accuracy – 77%
- Precision – 80%
- f1-score – 84%

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

3.2) Random Forest

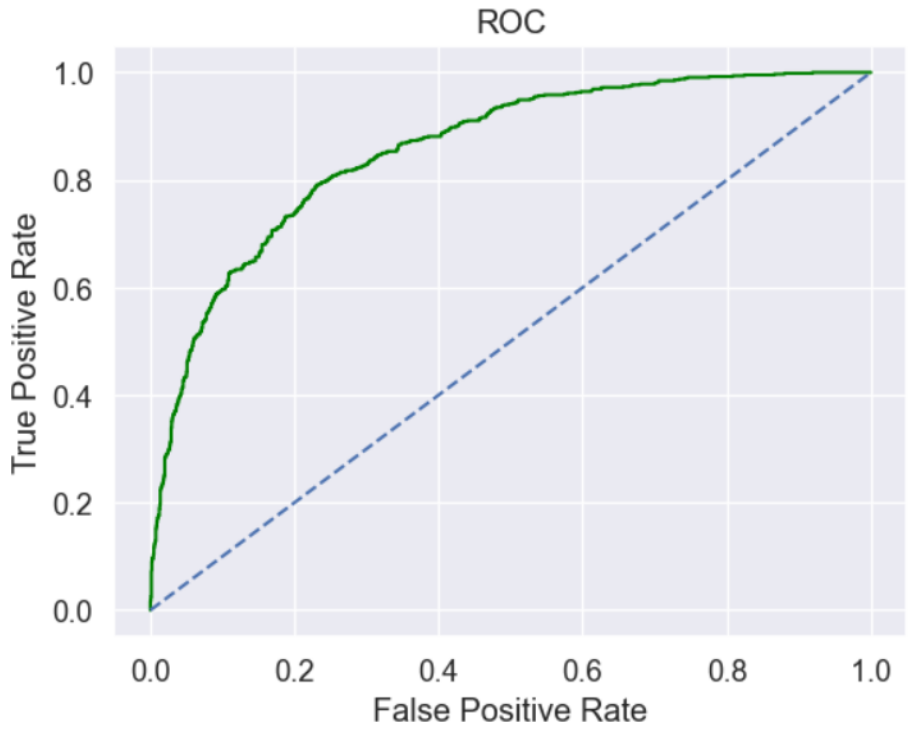
3.2.1) Performance Evaluation on Training data

Classification Report:

	precision	recall	f1-score	support
0	0.84	0.89	0.86	1453
1	0.72	0.61	0.66	647
accuracy			0.80	2100
macro avg	0.78	0.75	0.76	2100
weighted avg	0.80	0.80	0.80	2100

ROC:

Area under the curve is 0. 8563713512840778.



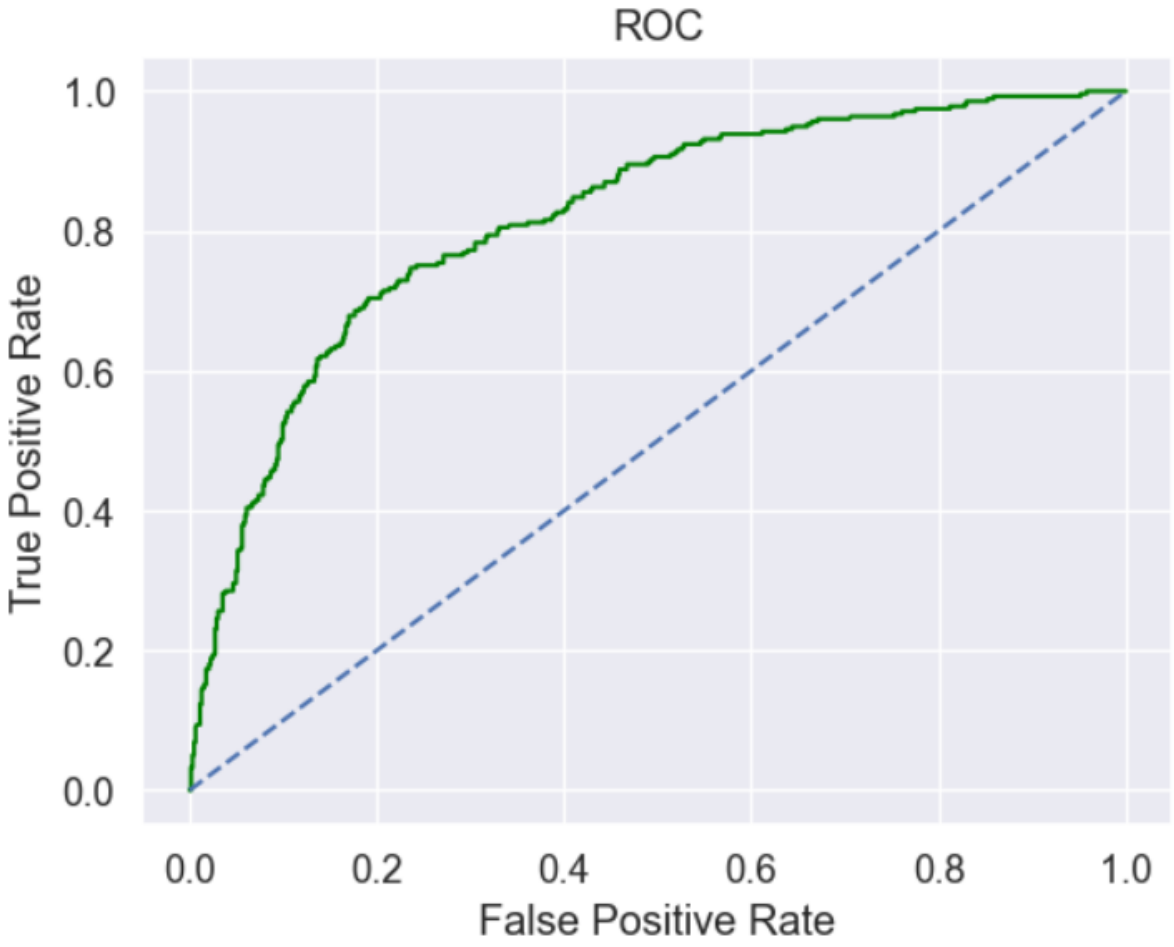
3.2.2) Performance Evaluation on Test data

Classification Report:

	precision	recall	f1-score	support
0	0.82	0.88	0.85	623
1	0.68	0.56	0.62	277
accuracy			0.78	900
macro avg	0.75	0.72	0.73	900
weighted avg	0.78	0.78	0.78	900

ROC:

Area under the curve is 0. 8181994657271499.



3.2.3) Random Forest Conclusion

Train Data:

- AUC – 86%
- Accuracy – 80%
- Precision – 72%
- f1-score – 66%

Test Data:

- AUC – 82%
- Accuracy – 78%
- Precision – 68%
- f1-score – 62%

Training and test set result are almost similar, and with the overall measures high, the model is a good model.

3.3) Artificial Neural Network

3.3.1) Performance Evaluation on Training data

Confusion Matrix:

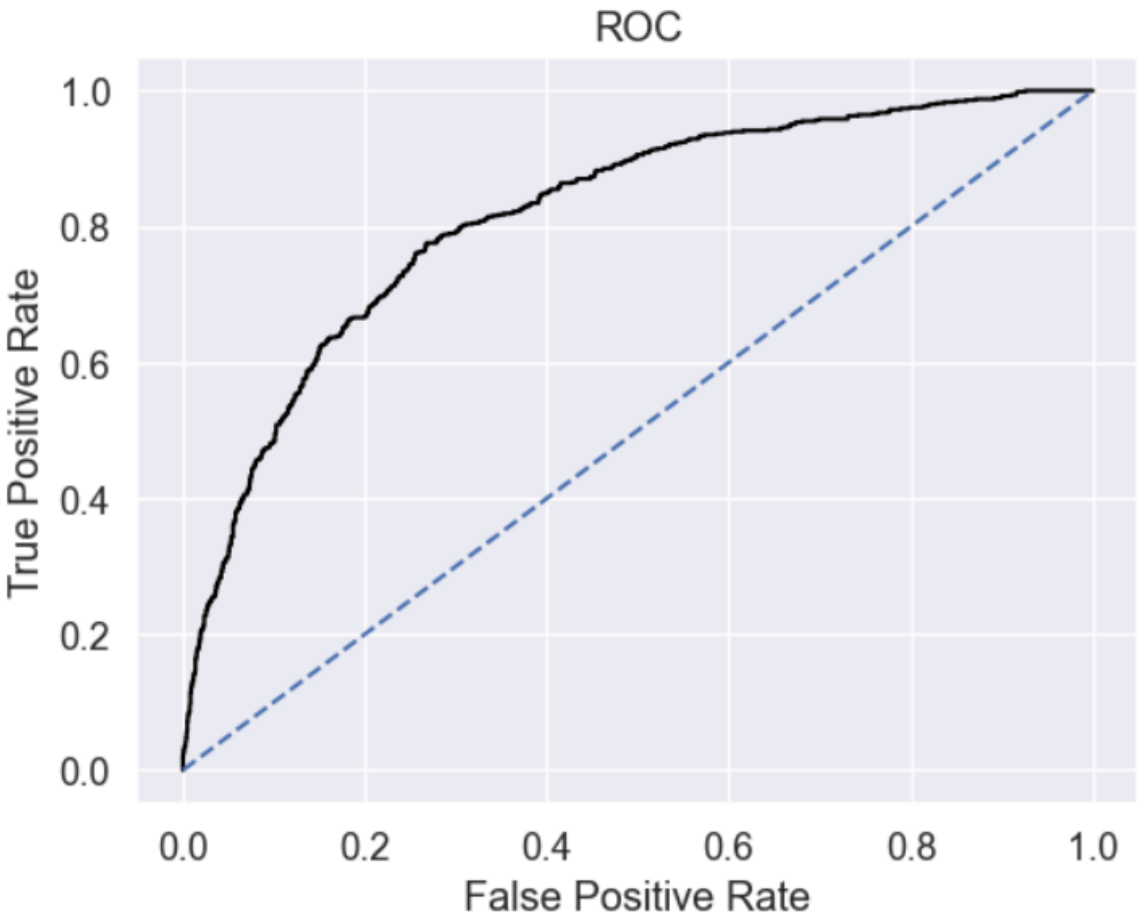
```
array([[1298,  155],
       [ 315,  332]], dtype=int64)
```

Classification Report:

	precision	recall	f1-score	support
0	0.80	0.89	0.85	1453
1	0.68	0.51	0.59	647
accuracy			0.78	2100
macro avg	0.74	0.70	0.72	2100
weighted avg	0.77	0.78	0.77	2100

ROC:

Area under the curve is 0.8166837040243976



3.3.2) Performance Evaluation on Test data

Confusion Matrix:

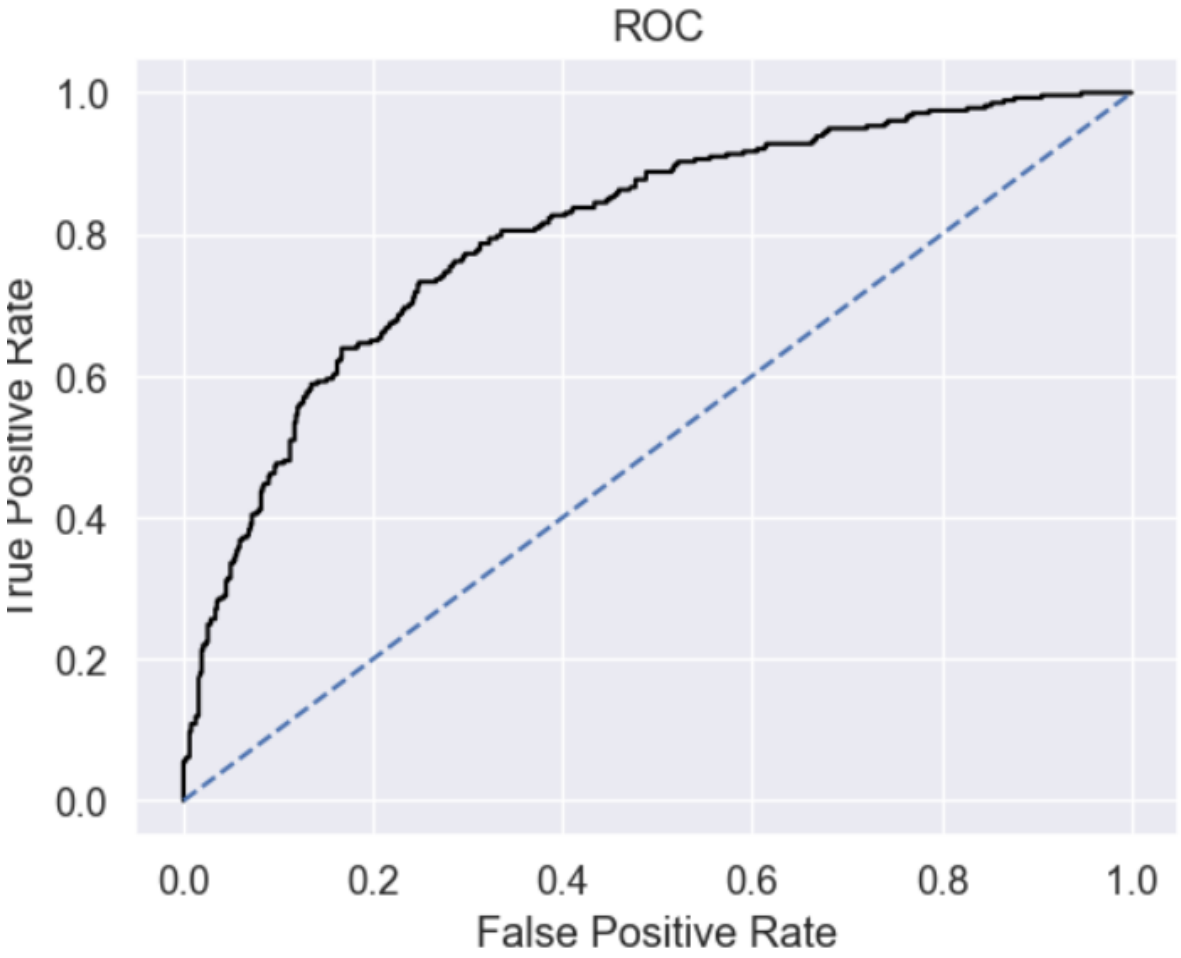
```
array([[553,  70],
       [138, 139]], dtype=int64)
```

Classification Report:

	precision	recall	f1-score	support
0	0.80	0.89	0.84	623
1	0.67	0.50	0.57	277
accuracy			0.77	900
macro avg	0.73	0.69	0.71	900
weighted avg	0.76	0.77	0.76	900

ROC:

Area under the curve is 0.8044225275393896



3.3.3) Artificial Neural Network Conclusion:

Train Data:

- AUC – 82%
- Accuracy – 78%
- Precision – 68%
- f1-score – 59%

Test Data:

- AUC – 80%
- Accuracy – 77%
- Precision – 67%
- f1-score – 57%

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

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

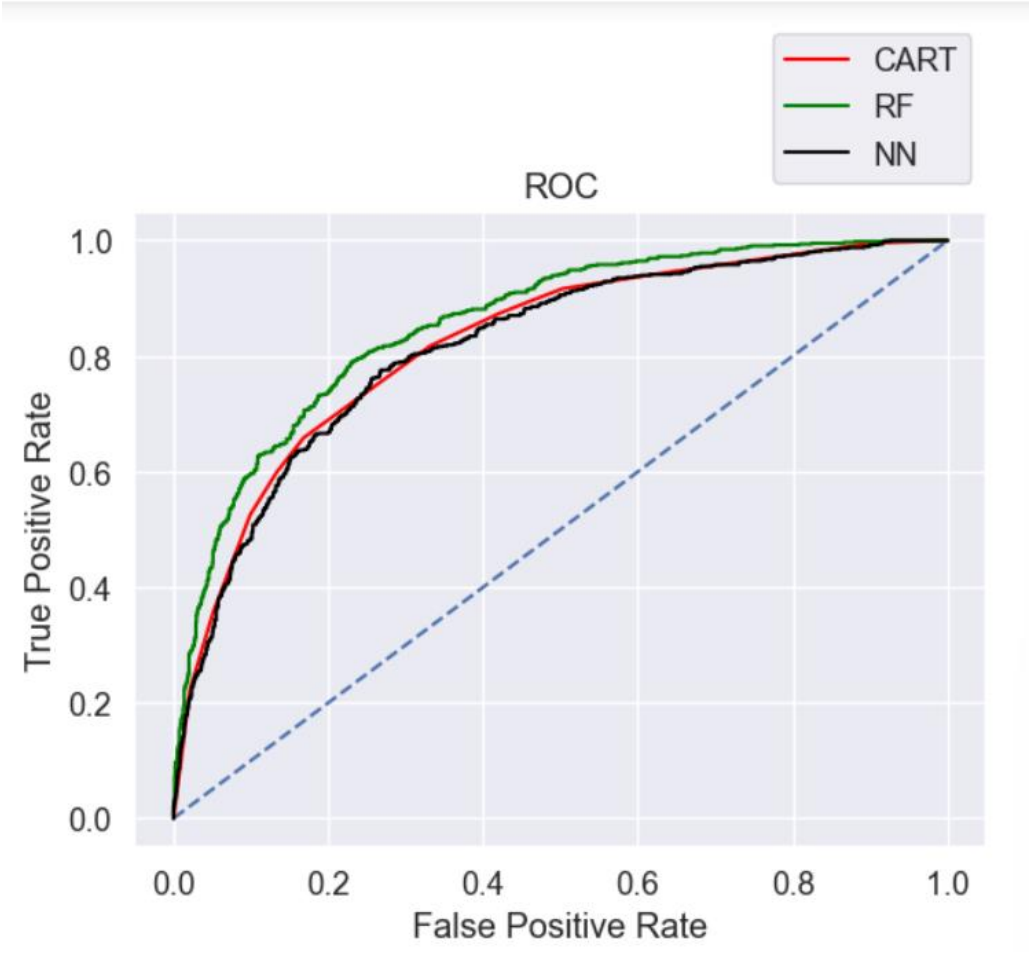
Below we are comparing Accuracy, AUC, Recall, Precision and F1 score of all the models, where Target is 0, i.e. the claimed as NO.

The logic to choose Claimed as NO is that the model is calculating Claimed as No more accurately than Claimed as Yes. Also, this way we will be able to identify using the attributes that which policy will not be claimed with more than approx. 75% accuracy.

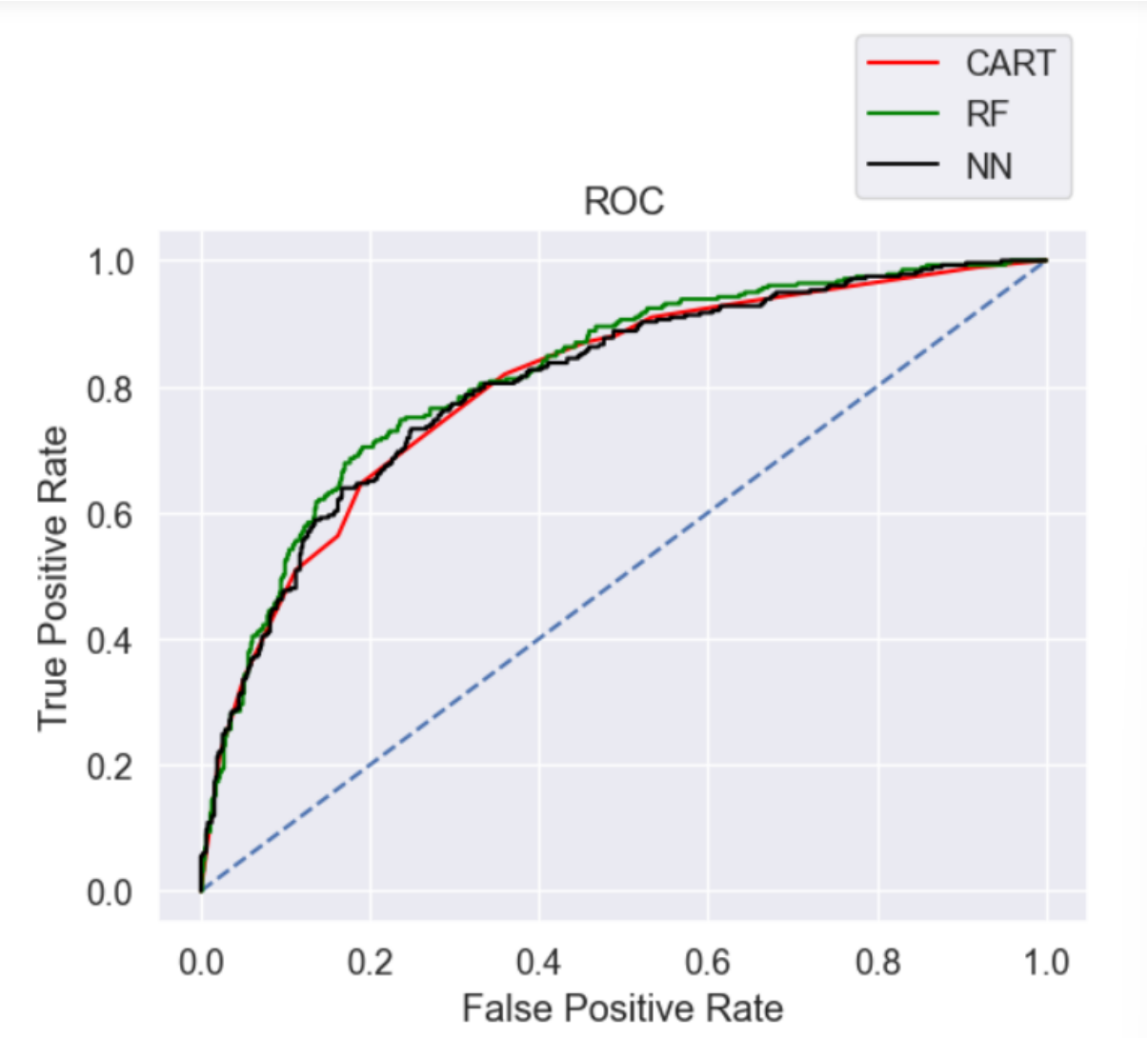
4.1) Comparison of the performance metrics from the 3 models

	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.79	0.77	0.80	0.78	0.78	0.77
AUC	0.82	0.80	0.86	0.82	0.82	0.80
Recall	0.53	0.51	0.61	0.56	0.51	0.50
Precision	0.70	0.67	0.72	0.68	0.68	0.67
F1 Score	0.60	0.58	0.66	0.62	0.59	0.57

4.2) ROC curve for the 3 models on the training data



4.3) ROC curve for the 3 models on the test data



4.4) Conclusion

Out of the 3 models, Random Forest has slightly better performance than the Cart and Neural network model.

Overall, all the 3 models are reasonably stable enough to be used for making any future predictions. From Random Forest Model, the variable change is found to be the most useful feature amongst all other features for predicting if a person will claim or not. If change is NO, then those policies have more chances of getting claimed.

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

- It seems that all the models show high accuracy in predicting the customers who will not claim for tour insurance after performing the supervised learning algorithm.
- As we know this Insurance firm was facing higher claim frequency and this model would certainly help in reducing the ratio.
- Since the variable Agency code seem to be the most important factor in deriving the model, therefore I recommend that the insurance company tie up with more Agencies to expand its business.
- To attain the less frequency of claims, they should add certain steps to their policy's terms and conditions that would benefit both customers and company.
- Using this model and customer data this insurance firm can easily pick their profitable customers.
- Team can easily target the customers who will not claim for tour insurance. Once Team receives customer data who falls under NO claim status as per the model, then team needs to build strong relationship with those customers because you only get profit when repeated customer sees loyalty and trust in an organization.
- I believe that the tour insurance company should also increase its varieties for Product Name. For now, they are having Bronze, Cancellation, Customized, Gold and Silver plans, but adding few more to the list will encourage customers to choose the optimum plan which proves to be the successful for them and in return, would lead to less frequency of claims for the company. The same would also result in more sales for the tour insurance company.
- Product plan which has higher commission rate can be recommended to the set customers who will fall under NO claim status.