SMDM Project: Advanced Statistics

CLUSTERING, CART, RANDOME FOREST AND ARTIFICIAL NEURAL NETWORK

Student's Name: THILAK RAJ | Batch: 18 FEB 2022

Executive Summary (PROBLEM-1)

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

Data Dictionary for Market Segmentation:

- Spending: Amount spent by the customer per month (in 1000s)
- Advance_payments: Amount paid by the customer in advance by cash (in 100s)
- Probability_of_full_payment: Probability of payment done in full by the customer to the bank
- Current_balance: Balance amount left in the account to make purchases (in 1000s)
- Credit_limit: Limit of the amount in credit card (10000s)
- Min_payment_amt : minimum paid by the customer while making payments for purchases made monthly (in 100s)
- Max_spent_in_single_shopping: Maximum amount spent in one purchase (in 1000s)

1.1 Read the data and do exploratory data analysis. Describe the data briefly.

Sample of 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

OBSERVATION:

- Dataset consists of 7 columns.
- Spending which the amount spent by the customer per month in 1000s.
- Advance_payments is the amount paid by the customer in advance by cash in 100s.
- Probability_of_full_payment is the Probability of payment done in full by the customer to the bank.
- Current_balance is the Balance amount left in the account to make purchases given in 1000s.
- Credit_limit is the Limit of the amount in credit card given in 10000s.
- Min_payment_amt is the minimum amount paid by the customer while making payments for purchases made monthly which is provided in 100s.

- Max_spent_in_single_shopping is the maximum amount spent in one purchase which is given in 1000s.
- Dataset is consisting of 210 individual's data.

Exploratory Data Analysis:

Let us check the types of variables and Missing Values in the data frame.

```
<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
```

Observation:

- **7 Columns and 210 non-null records.**
- **There is no missing record based on initial analysis.**
- **All the variables numeric type**
- **Found no null values.**
- **Found no duplicate data.**

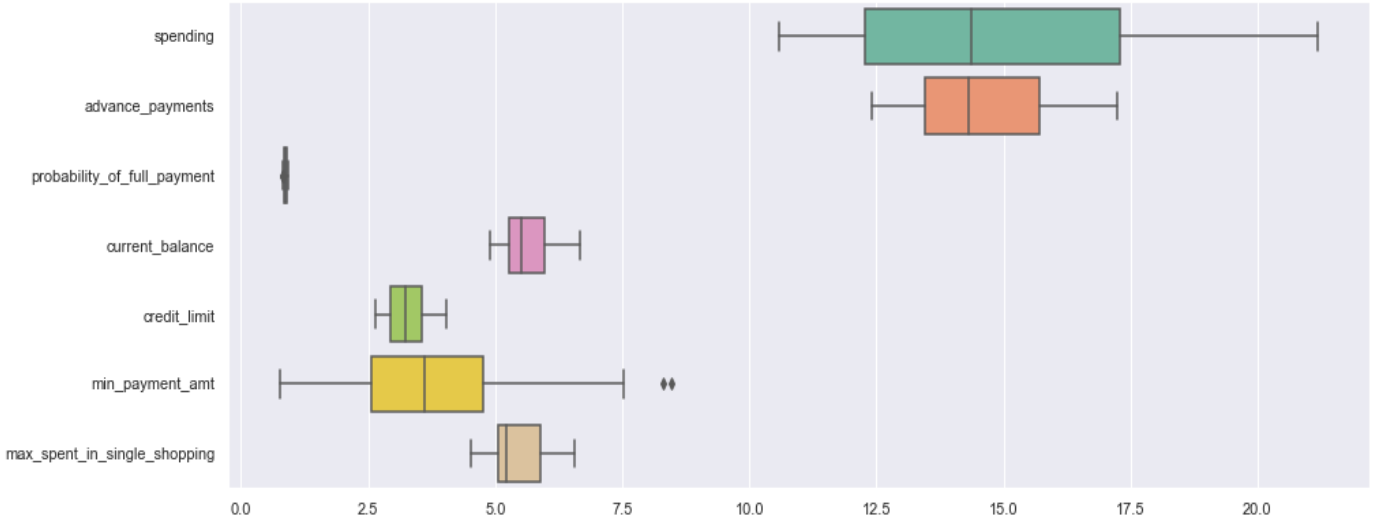
UNIVARIENT ANALAYSIS:

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

OBSERVATIONS:

- **As per the above summary, the data looks good.**
- **Mean of spending and advanced payments are almost equal.**
- **Mean of other variables are nearly equal.**
- **Standard Deviation is high for spending.**
- **We can standardize the data for further analysis.**

OUTLIER VALIDATION:



From the above graph we can see that, there are outliers for existingprobability_of_full_payment and min_payment_amt variables.

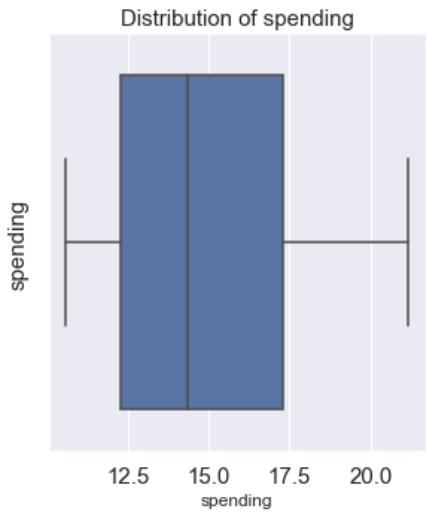
SPENDING Variable:

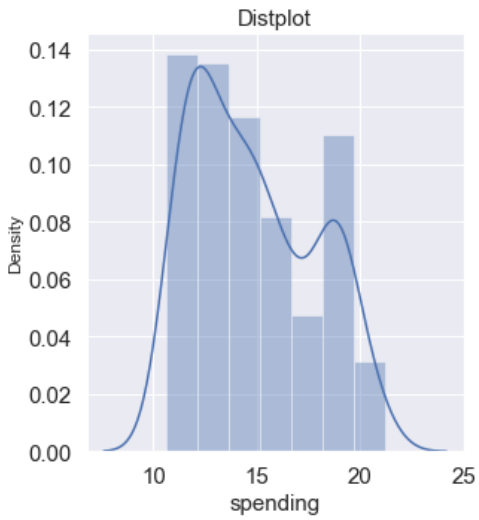
From descriptive summary:

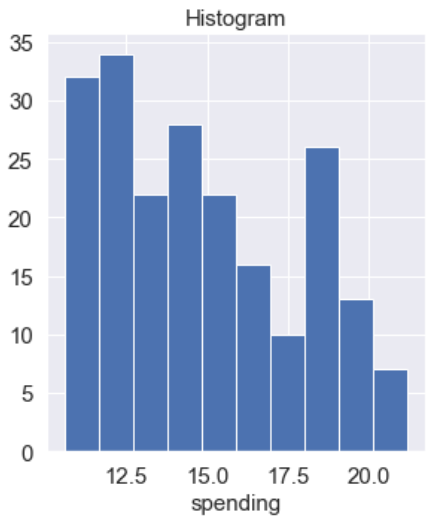
- Minimum spending: 10.59
- Maximum spending: 21.18
- Mean value: 14.847523809523818
- Median value (50%): 14.355
- Standard deviation: 2.909699430687361
- Null values: Not found

- 1st Quartile (Q1) is: 12.27
- 3rd Quartile (Q3) is: 17.305
- IQR of spending is 5.035
- Lower outliers in spending: 4.717499999999999
- Upper outliers in spending: 24.8575

Below is the Box plot, Distplot and Hostogram for the variable 'Spending'.







OBSERVATION:

- **There is no outlier present.**
- **Data is right skewed.**

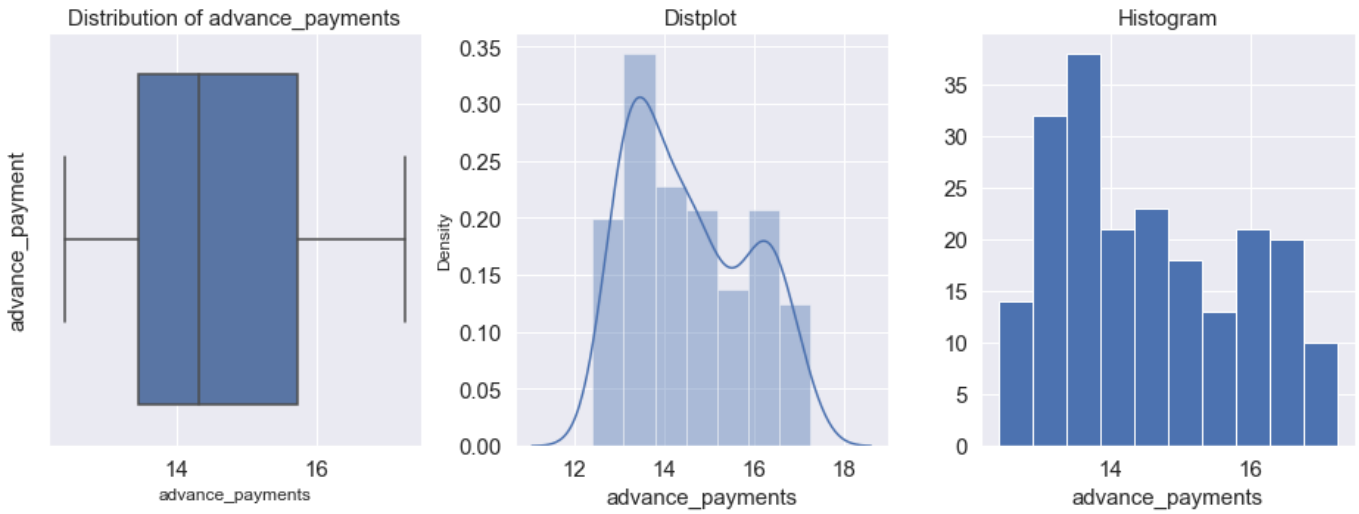
ADVANCE_PAYMENT:

From descriptive summary:

- Minimum advance_payments: 12.41
- Maximum advance_payments: 17.25
- Mean value: 14.559285714285727
- Median value (50%): 14.32
- Standard deviation: 1.305958726564022
- Null values: Not found

- 1st Quartile (Q1) is: 13.45
- 3RD Quartile (Q3) is: 15.715
- IQR of advance_payments is 2.2650000000000006
- Lower outliers in advance_payments: 10.052499999999998
- Upper outliers in advance_payments: 19.1125

Below is the Box plot, Distplot and Hostogram for the variable 'Advanced_payments'.



OBSERVATION:

- **There is no outlier present.**
- **Data is right skewed.**

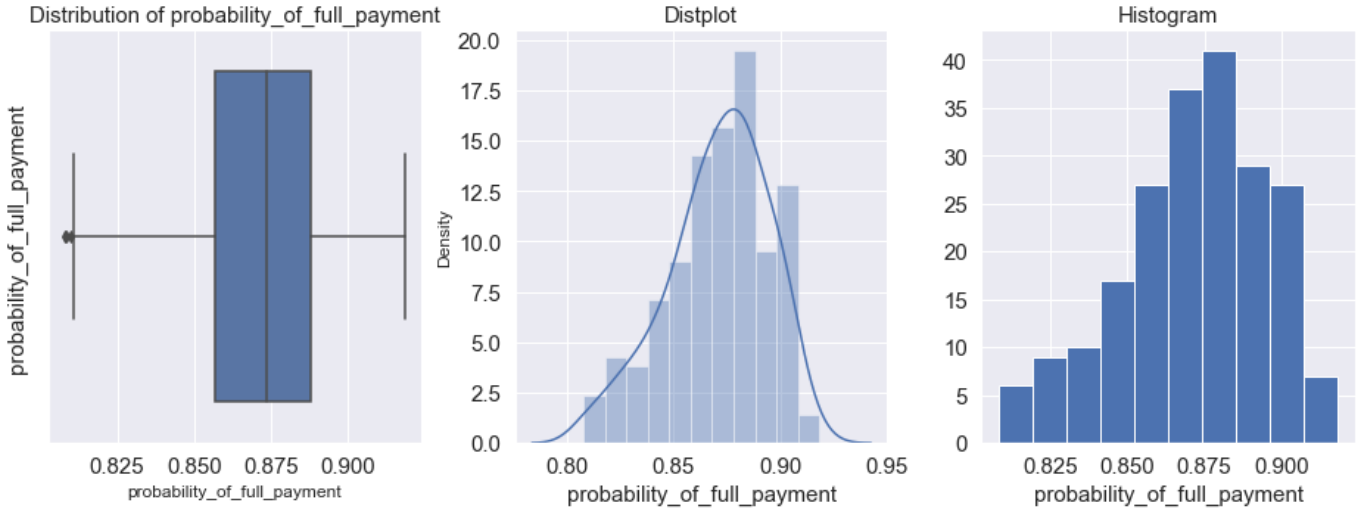
PROBABILITY_OF_FULL_PAYMENT:

From the descriptive summary:

- Minimum probability_of_full_payment: 0.8081
- Maximum probability_of_full_payment: 0.9183
- Mean value: 0.8709985714285714
- Median value (50%): 0.8734500000000001
- Standard deviation: 0.023629416583846496
- Null values: Not found

- 1st Quartile (Q1) is: 0.8569
- 3rd Quartile (Q3) is: 0.887775
- IQR of probability_of_full_payments is 0.030874999999999986
- Lower outliers in probability_of_full_payment: 0.8105875
- Upper outliers in probability_of_full_payment: 0.9340875

Below is the Box plot, Distplot and Hostogram for the variable 'Probability_of_full_payment'.



OBSERVATION:

- **There are three outliers present below3rd quartile.**
- **Data is mostly left skewed.**

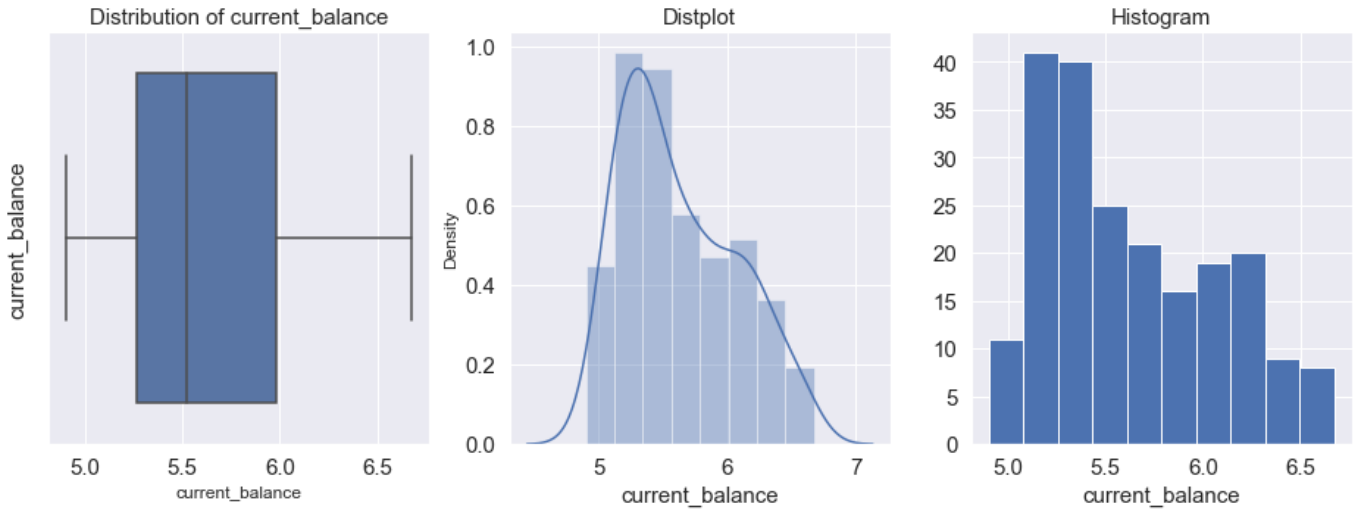
CURRENT_BALANCE:

From the descriptive summary:

- Minimum current_balance: 4.899
- Maximum current_balance: 6.675
- Mean value: 5.628533333333334
- Median value (50%): 5.5235
- Standard deviation: 0.4430634777264493
- Null values: Not found.

- 1st Quartile (Q1) is: 5.26225
- 3rd Quartile (Q3) is: 5.97975
- IQR of current_balance is 0.7175000000000002
- Lower outliers in current_balance: 4.186
- Upper outliers in current_balance: 7.056000000000001

Below is the Box plot, Distplot and Hostogram for the variable 'Current_balance'.



OBSERVATION:

- **There are no outliers present in the data.**
- **Data is mostly right skewed.**

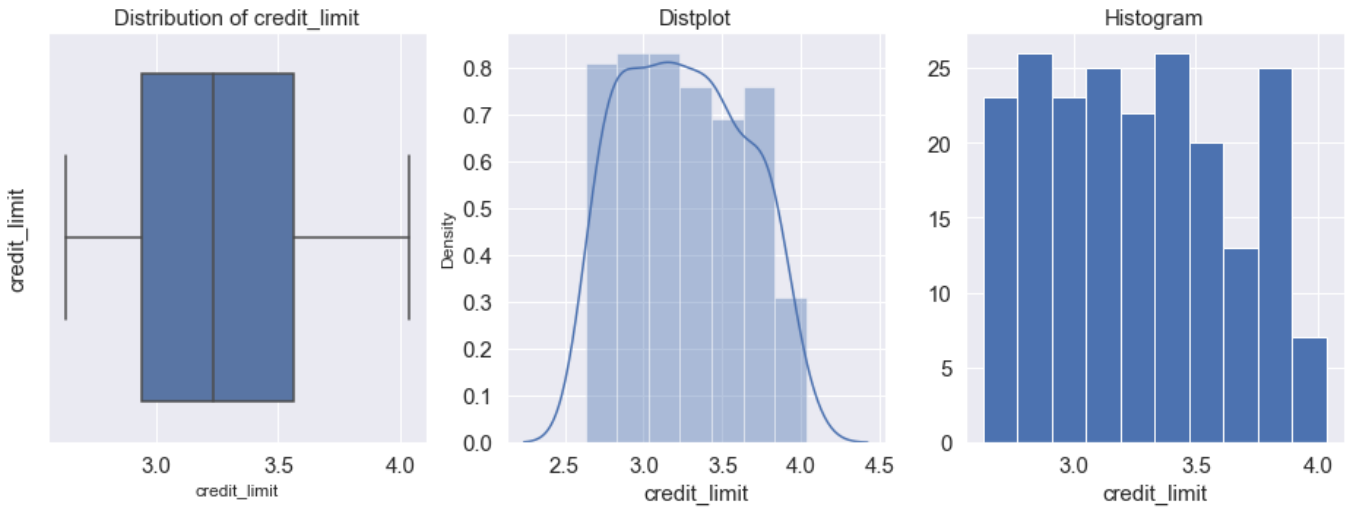
CREDIT_LIMIT:

From the descriptive summary:

- Minimum credit_limit: 2.63
- Maximum credit_limit: 4.033
- Mean value: 3.258604761904763
- Median value (50%): 3.237
- Standard deviation: 0.3777144449065874
- Null values: Not found

- 1st Quartile (Q1) is: 2.944
- 3st Quartile (Q3) is: 3.56175
- IQR of credit_limit is 0.61775
- Lower outliers in credit_limit: 2.017375
- Upper outliers in credit_limit: 4.488375

Below is the Box plot, Distplot and Hostogram for the variable 'Credit_limit'.



OBSERVATION:

- **There are no outliers present in the data.**
- **Data is equally distributed.**

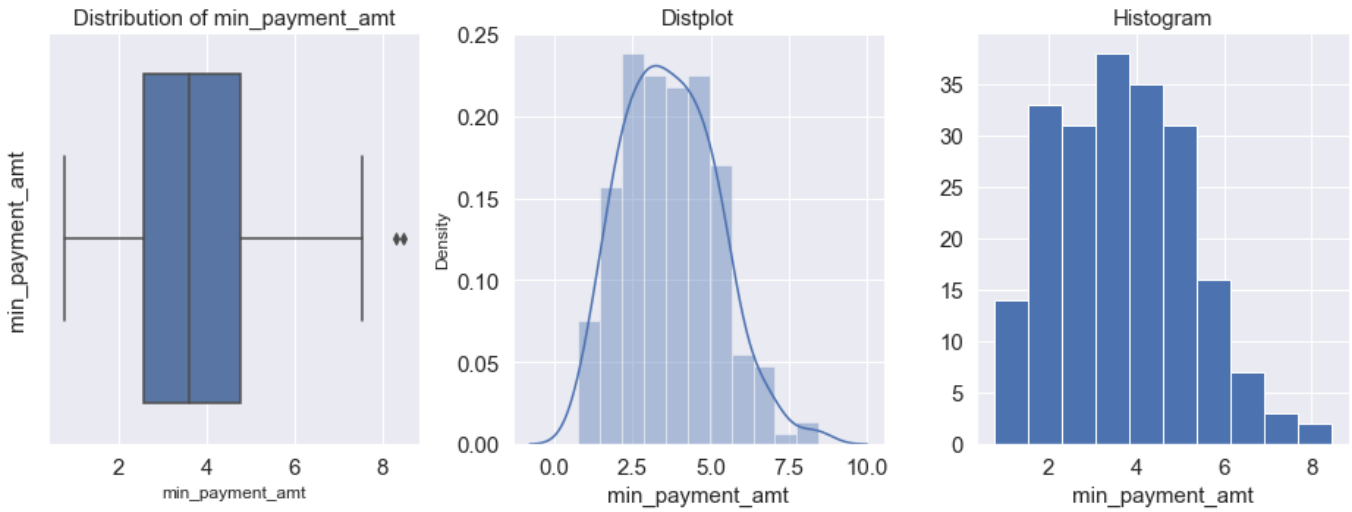
MIN_PAYMENT_AMT:

From the descriptive summary:

- Minimum min_payment_amt: 0.7651
- Maximum min_payment_amt: 8.456
- Mean value: 3.7002009523809507
- Median value (50%): 3.599
- Standard deviation: 1.5035571308217792
- Null values: Not Found

- 1st Quartile (Q1) is: 2.5615
- 3rd Quartile (Q3) is: 4.76875
- IQR of min_payment_amt is 2.2072499999999997
- Lower outliers in min_payment_amt: -0.7493749999999992
- Upper outliers in min_payment_amt: 8.079625

Below is the Box plot, Distplot and Hostogram for the variable 'min_payment_amt'.



OBSERVATION:

- **There are 2 outliers are present in the data which is in the right side of the upper limit (3rd quartile).**
- **Data is right skewed.**

MAX_SPENT_IN_SINGLE_SHOPPING:

From the descriptive summary:

- Minimum max_spent_in_single_shopping: 4.519
- Maximum max_spent_in_single_shoppings: 6.55
- Mean value: 5.408071428571429
- Median value (50%): 5.223000000000001
- Standard deviation: 0.4914804991024054
- Null values: Not found

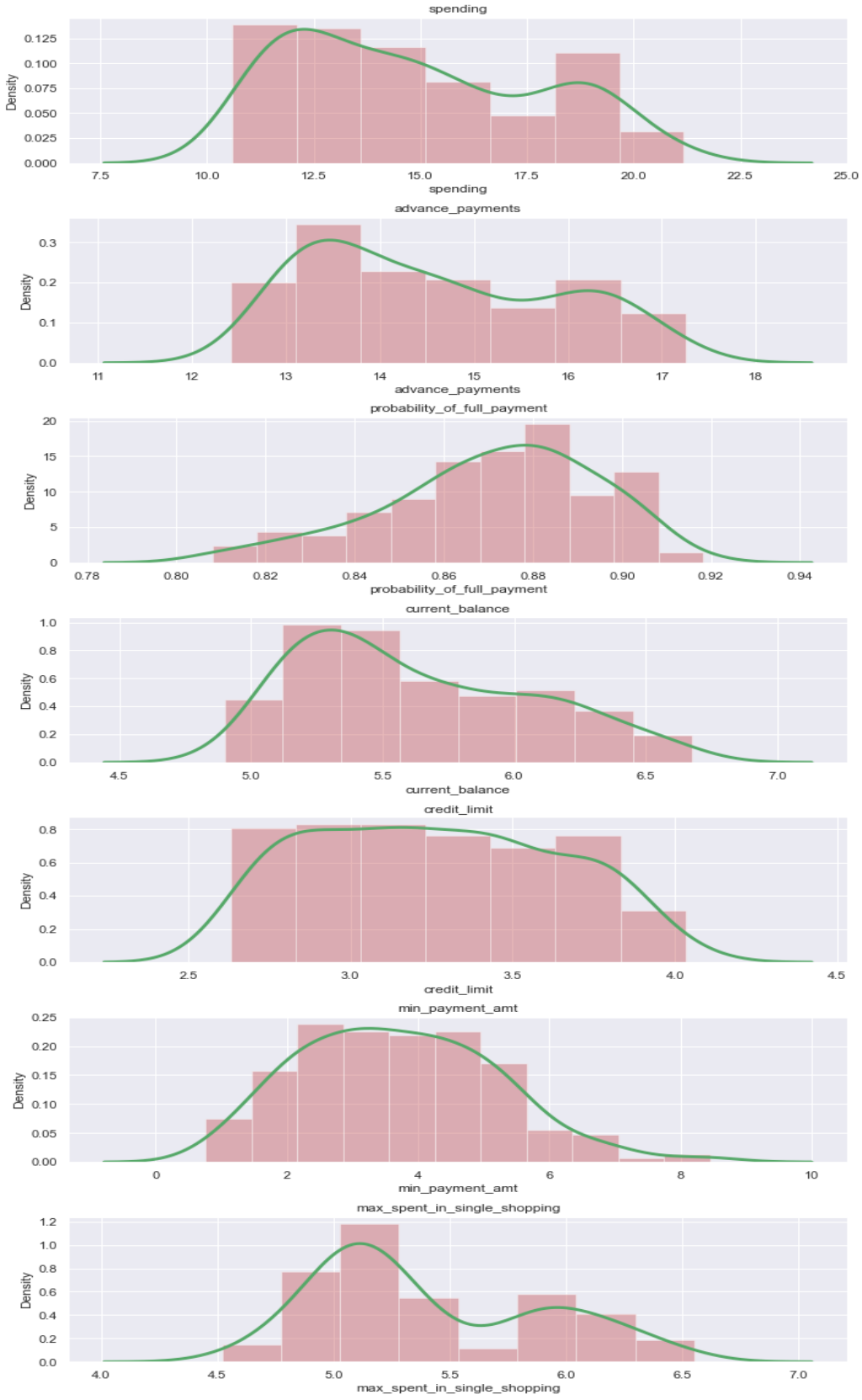
- 1st Quartile (Q1) is: 5.045
- 3rd Quartile (Q3) is: 5.877
- IQR of max_spent_in_single_shopping is 0.8319999999999999
- Lower outliers in max_spent_in_single_shopping: 3.797
- Upper outliers in max_spent_in_single_shopping: 7.125

Below is the Box plot, Distplot and Hostogram for the variable 'max_spent_in_single_shopping'.

OBSERVATION:

- **There are no outliers are present in the data.**
- **Data is right skewed.**

KDE to represent the skewness and the density of the data:



The skewness values quantitatively:

```
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
```

Observations

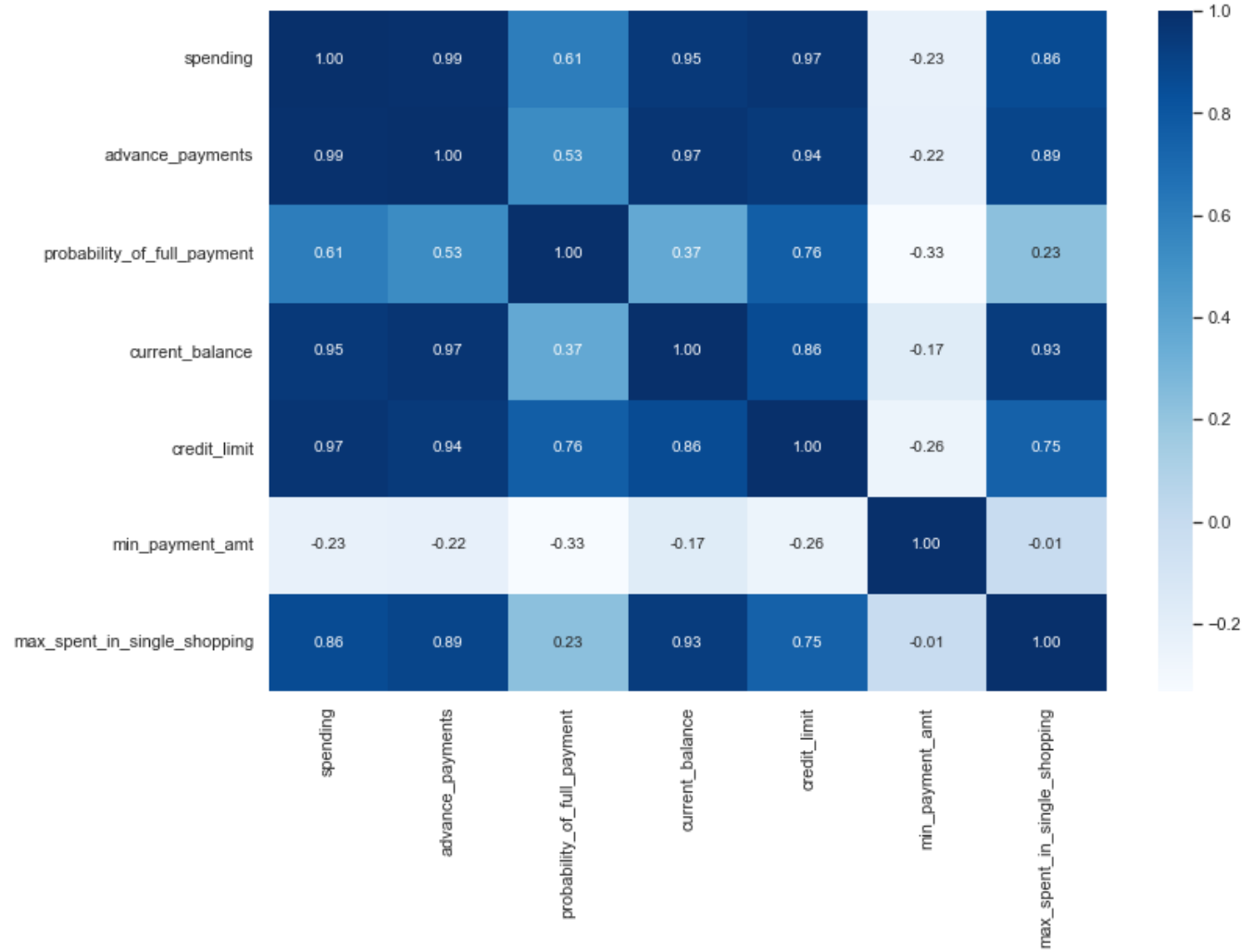
- **Credit_limit has the least skewness (nearer to zero), and it is almost normally distributed.**
- **Remaining all the variables are right skewed execpt probability_of_full_payment variable, which is left skewed.**
- **max_spent_in_single_shopping variable with highest skewness with 0.562 which is right skewed.**
- **The only left skewed variable is probability_of_full_payment with -0.538**

MULTIVARIENT ANALYSIS:

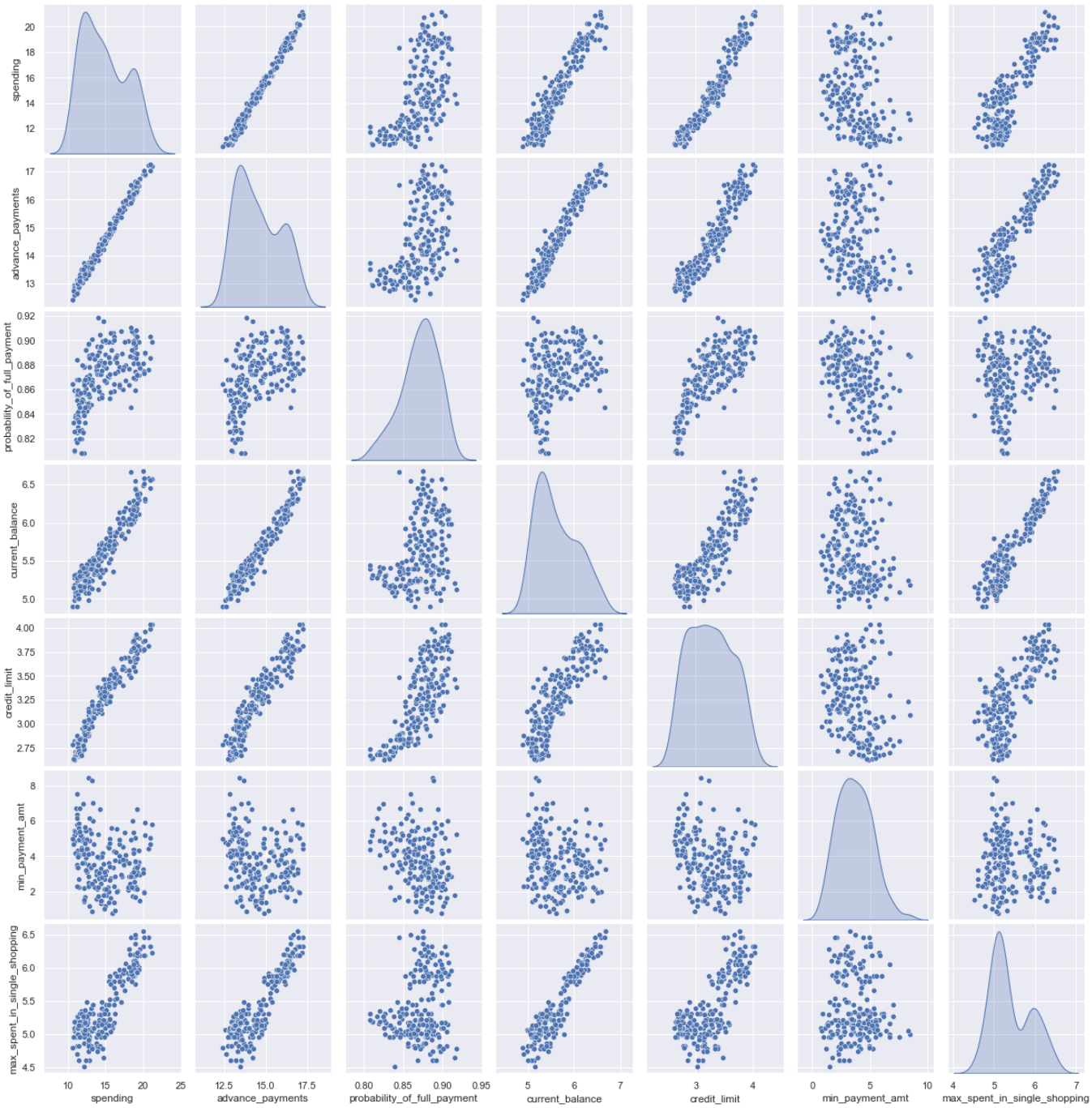
Below table denotes the multicollinearity between variables.

	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
bability_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
ɔent_in_single_shopping	0.863693	0.890784	0.226825	0.932806	0.749131	-0.011079	1.000000

Visualizing the correlation between variables.



Correlation representation using pair plot.



OBSERVATION:

- **Strong positive correlation between: -spending & advance_payments,**
  - -spending & current_balance,

- -spending & credit_limit,
- -advance_payments & current_balance,
- -credit_limit & advance_payments,
- -max_spent_in_single_shopping and current_balance.

- **The lowest negative correlation is between min_payment_amt and probability_of_full_payment**

NOTE:

We are not treating outliers as there are no extreme outliers are present in the dataset. There are totally 5 outliers are from 2 variables which are not extreme values.

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

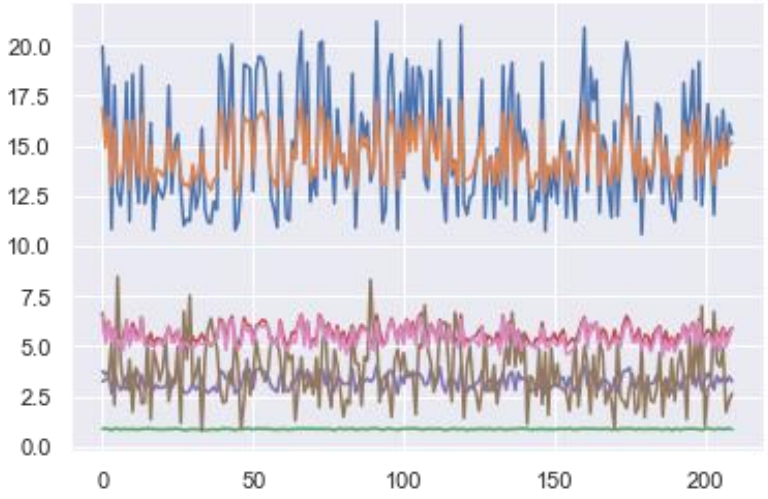
Let's investigate the below descriptive summary.

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

OBSERVATION:

- Scaling needs to be done as the values of the variables are different.
- From the above data, spending, advance_payments are slightly in different values, and this may get more weightage.
- Scaling can have all the values in the relative same range.
- Here I am using z-score to standardize the data.
- Below, I have displayed the plot of the data before and after scaling.

The plot of the data before Scaling:

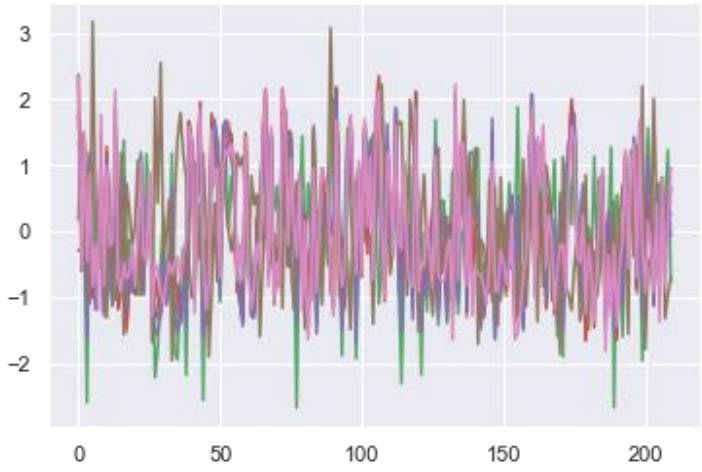


After scaling the attributes using z-score method.

	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

Table look good after scaling. I have used z-score to standardize the data to relative same scale -3 to +3.

The plot of the data after Scaling:



OBSERVATION:

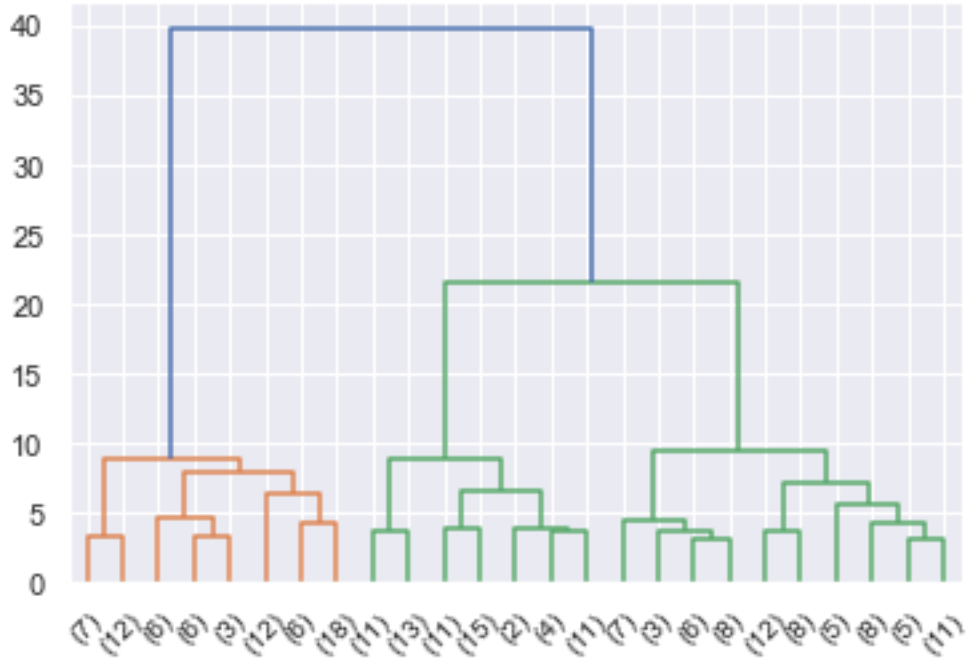
After scaling, we have all the values in the relative same range which is observed in the above plot.

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

We are creating Dendogram using two different linkage methods, ward and average.

Choosing **Ward** Linkage method:

Below is the Dendogram after Cutting with suitable clusters (Clusters P=25).



If we consider 3 clusters using the criterion as 'maxclust', below is the data in array format.

```
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)
```

We are creating a new column for the cluster selection in the original dataframe. Below is the original dataframe with the new column 'Clusters3' obtained from wards linkage method.

	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

Below is the table which represents the frequency of each cluster and the mean of each variable belong to the cluster.

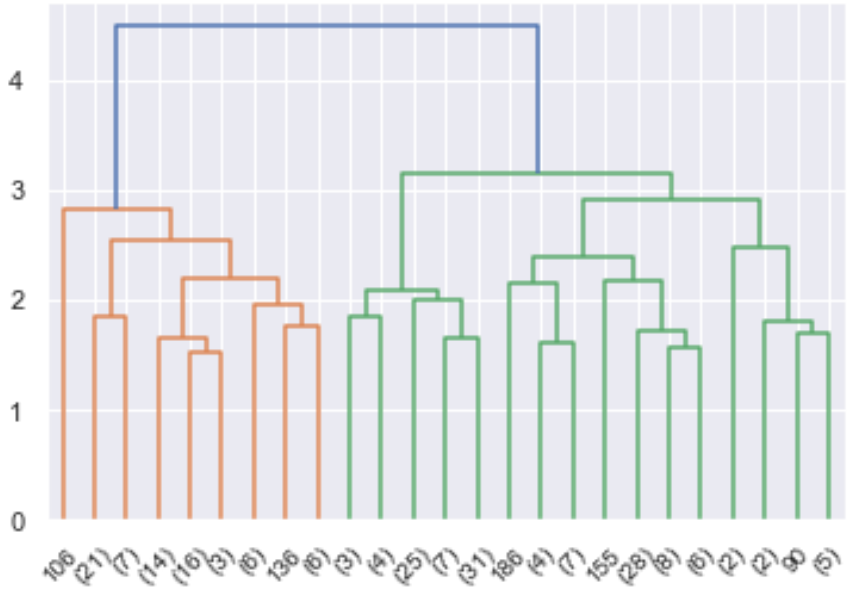
clusters-3	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371	70
2	11.872388	13.257015	0.848072	5.238940	2.848537	4.949433	5.122209	67
3	14.199041	14.233562	0.879190	5.478233	3.226452	2.612181	5.086178	73

OBSERVATION:

Out of 210 rows, 70 rows fall to the first cluster and 67 rows fall to the second cluster and 73 rows fall to the third cluster.

Choosing **Average** Linkage method:

Below is the Dendogram after Cutting with suitable clusters (Clusters P=25)



If we consider 3 clusters using the criterion as 'maxclust', below is the data in array format.

```
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)
```

We are creating a new column for the cluster selection in the original dataframe. Below is the original dataframe with the new column 'Clusters3' obtained from wards linkage method.

	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

Below is the table which represents the frequency of each cluster and the mean of each variable belong to the cluster.

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

Out of 210 rows, 75 rows fall to the first cluster and 70 rows fall to the second cluster and 65 rows fall to the third cluster.

OBSERVATION:

- **Both the ward and average are almost similar means, minor variation, which we know it occurs.**
- **For cluster grouping based on the dendrogram, 3 or 4 looks good.**
- **We 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.**
- **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).**

1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score.

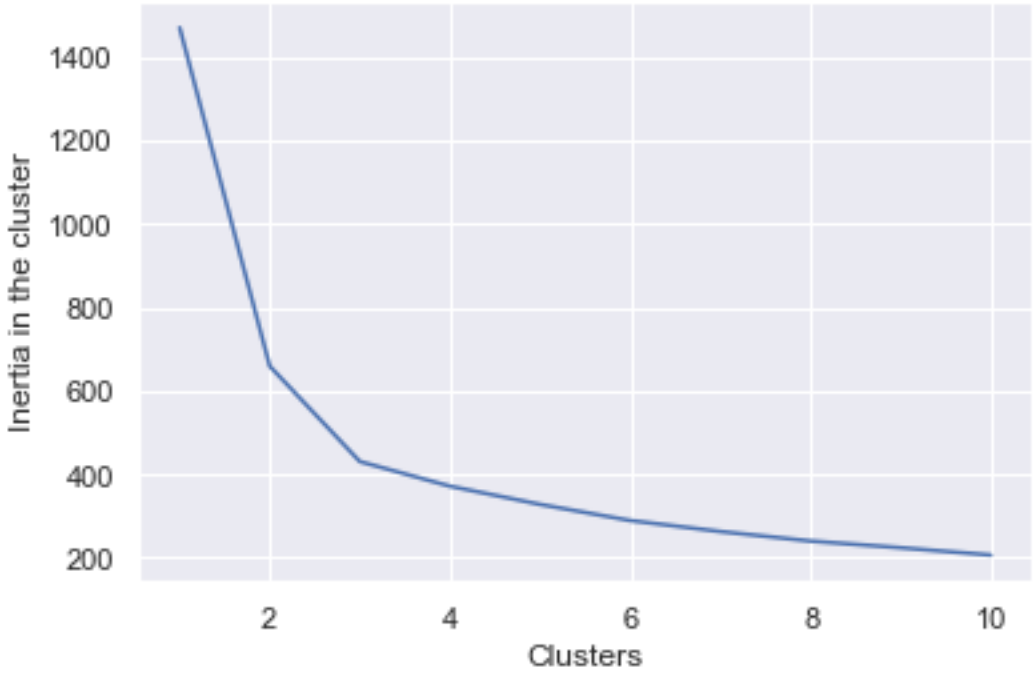
Calculation of WSS (inertia for each value of k):

Below table represents the inertia for each value of cluster k from 1 to 10.

```
[1469.9999999999995,
 659.1717544870411,
 430.65897315130064,
 371.5811909715524,
 327.9852689524273,
 290.2329426676273,
 262.0015171444639,
 246.67172110083214,
 222.0713773446485,
 206.42883602930118]
```

We can observe that there is a huge difference between the values till the k value become 3 and 4. There is not much difference after k=4. Therefore, we can suspect k=3 or k=4 are the optimal values.

Let us visualize the same with the WSS(within sum of square) plot.



Elbow Method is applied and now visualized with different values of K. Finalizing k=3 as the optimal value as there is no drastical reduction in inertia after k=3.

Using the K-means clustering method on scaled data, we assigned each record into the one of the three optimal clusters according to their distance from each cluster. We have appended this data into a new column Clus_kmeans in the original dataframe.

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

Now we are validating if the observations from the previous step are correctly mapped to correct clusters using Silhouette score.

NOTE: We found silhouette_score as 0.4007 which is not a negative or close to zero value. As it is positive and closer to 1, we can consider this is a well distinguished cluster.

Now we are validating Silhoutte Score for different cluster k values ranges from 2 to 10 and plotting the sc scores.

```
[0.46577247686580914,
 0.40072705527512986,
 0.3347542296283262,
 0.28621461554288646,
 0.285726896652541,
 0.2756098749293962,
 0.24943558736282168,
 0.2666366921192433,
 0.2731288488219916]
```

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

We considered k=3 is the optimal cluster and calculating the Sil-Width for all records.

spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Clus_kmeans	sil_width
19.94	16.92	0.8752	6.675	3.763	3.252	6.550	1	0.573699
15.99	14.89	0.9064	5.363	3.582	3.336	5.144	2	0.366386
18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1	0.637784
10.83	12.96	0.8099	5.278	2.641	5.182	5.185	0	0.512458
17.99	15.86	0.8992	5.890	3.694	2.068	5.837	1	0.362276

To check the proper mapping, we should fetch the minimum sil-width value from the final list.

We found 0.00271 is the minimum sil-width which is not a negative value. So, we can conclude that, mapping is good.

Below are the k-mean labels for entire records (cluster k=3) in array format.

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

Proportion or Frequency of labels classified:

```
2    72
0    71
1    67
dtype: int64
```

72 records belong to 1st cluster, 67 records belong to 2nd cluster and 72 records belong to the 3rd cluster.

K-Means Clustering & Cluster Information:

Below is the mean value of all variables belonging to each cluster.

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

Percentage of records in each cluster.

	Cluster_Size	Cluster_Percentage
1	71	33.81
2	72	34.29
3	67	31.90

OBSERVATION:

From the above analysis,

- **Based on current dataset given, 3 cluster solution makes sense based on the spending pattern i.e., High, Medium, Low**
- **We can perform the analysis on k=4 and k=5, but k=3 holding the better spending pattern, I am finalizing the k=3.**

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

After performing the Mean of clusters for K=3 using K-means clustering method.:

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

After performing the Mean of clusters for K=3 using hierarchical clustering method.:

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

From the above data, we can group the 3 clusters into 3 profiles.

- **Group 1: High Spending**
- **Group 2: Low Spending**
- **Group 3: Medium Spending**

Promotional strategies for each cluster:

Group 1: High Spending Group

- We could see, the maximum max_spent_in_single_shopping is high for this group, so can be offered discount/offer on next transactions upon full payment.
- Increase spending habits
- Giving any reward points might increase their purchases.
- Increase the credit limit to encourage the spending limit.
- as they are customers with good repayment record. can be offered with loan against the credit card.
- Can be tie up with new popular and unique luxury brands, which might drive more one_time_maximun spending.

Group 2: Low Spending Group

- Offers can be provided on early payments to improve their payment rate.
- As the full payment history is good, can be increase the credit limit, also lower down interest rate based on their purchases.
- Increase their spending habits by tying up with convenient brands, providing the reward points for each purchase etc.

Group 3: Medium Spending Group

- They are potential target customers who are paying bills and doing purchases and maintaining comparatively good credit score. So, we can increase credit limit or can lower down interest rate.
- Promote premium cards/loyalty rewards to increase transactions.
- Increase spending habits by trying with premium ecommerce sites, travel portal, travel airlines/hotel, as this will encourage them to spend more

Executive Summary (PROBLEM-2)

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 for Market Segmentation:

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

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

Sample of 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

Observation:

- From the above data, the variable 'Claimed' can be considered as dependent variable.
- Remaining variables, we can consider as independent variables.

Exploratory Data Analysis:

Let us check the types of variables and Missing Values in the data frame.

```
<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
```

Observation:

- There 10 variables
- numeric variable: Age, Commision, Duration and Sales.
- rest are categorical variables
- Found 3000 records
- No missing records.
- 9 independent variable and one target/Dependent variable 'Claimed'.

We are checking for null values in each column. Below is the result.

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

Observation:

- We found no missing values in the dataframe.

Descriptive Statistics Summary:

	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

Observation:

- Found min value of 'duration' is -1 which is negative. As it is not possible, this is a wrong entry.
- Found total 5 unique insurance product name for 3 unique destinations.

Identifying the unique values in each categorical variable.

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


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


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


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


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


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

Note: We noticed that there are 139 duplicate records found in the original dataframe.

Below are the sample duplicate records.

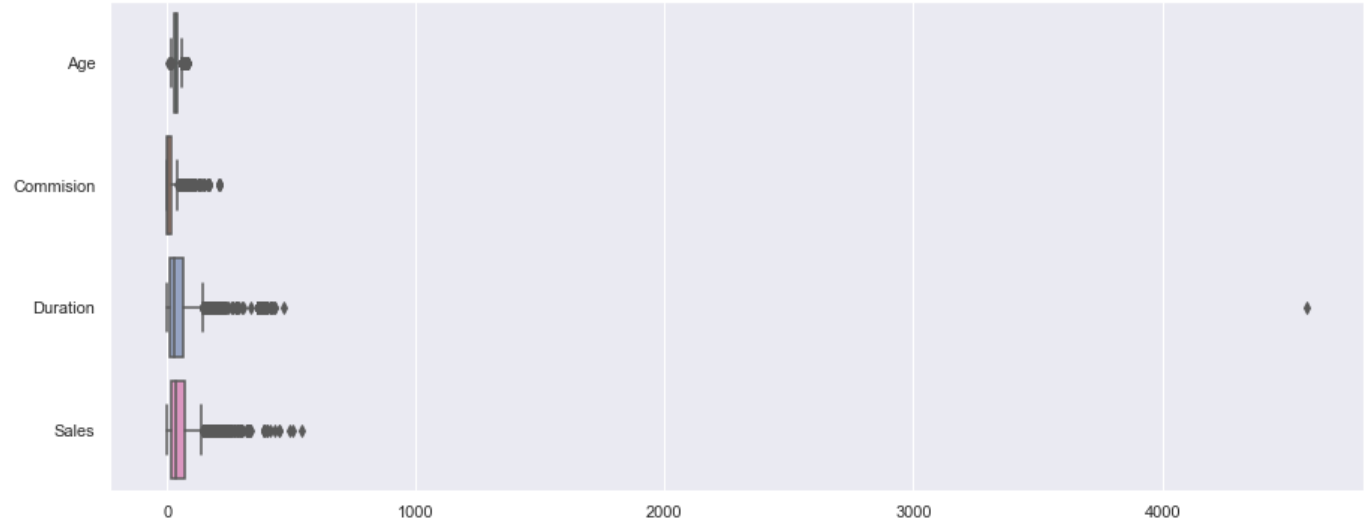
	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
63	30	C2B	Airlines	Yes	15.0	Online	27	60.0	Bronze Plan	ASIA
329	36	EPX	Travel Agency	No	0.0	Online	5	20.0	Customised Plan	ASIA
407	36	EPX	Travel Agency	No	0.0	Online	11	19.0	Cancellation Plan	ASIA
411	35	EPX	Travel 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

Note:

We can notice that the unique customer details like customer ID, email ID or phone number are not provided. So, these duplicate data can be of different customers. So, I am not going to removing the duplicate entries in dataframe.

UNIVARIATE ANALYSIS

Let's investigate the boxplot to identify the outliers of continuous variables.



Observation:

- We could see that all the numeric variables are having outliers in the data.
- Duration variable is having the highest value of outlier.

AGE Variable:

From Descriptive statistical summary:

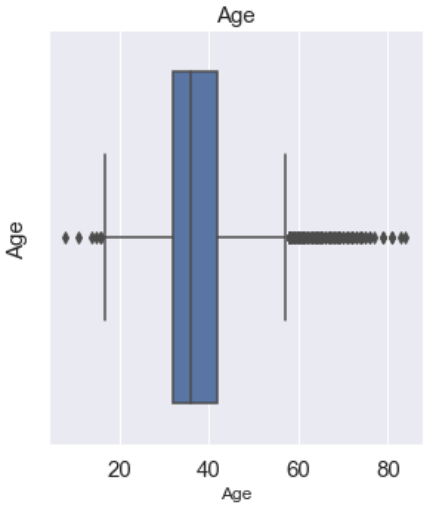
- Minimum Age: 8
- Maximum Age: 84
- Mean value: 38.091
- Median value (50%): 36.0
- Standard deviation: 10.463518245377944
- Null values: Not found

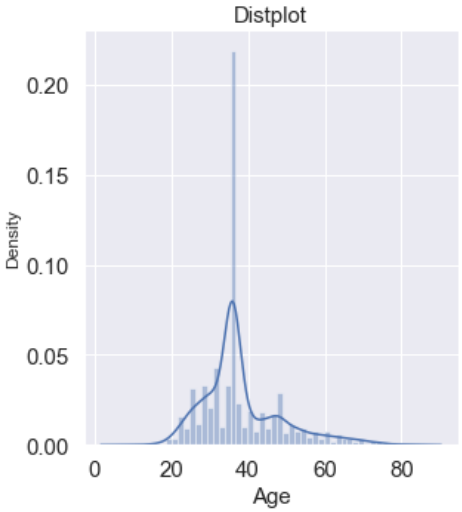
- spending - 1st Quartile (Q1) is: 32.0
- spending – 3rd Quartile (Q3) is: 42.0
- Interquartile range (IQR) of Age is 10.0
- Lower outliers in Age: 17.0
- Upper outliers in Age: 57.0

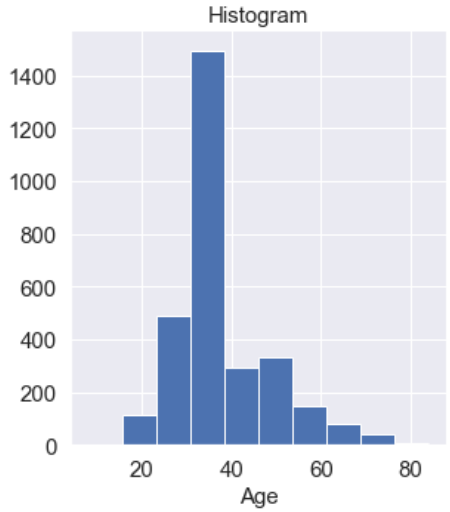
Outlier Details:

- Number of outliers in Age upper: 198
- Number of outliers in Age lower: 6
- percentage of Outlier in Age upper: 6.6 %
- percentage of Outlier in Age lower: 0.2 %

Below is the Box plot, Distplot and Hostogram for the variable 'Age'.







COMMISIOM Variable:

From Descriptive statistical summary:

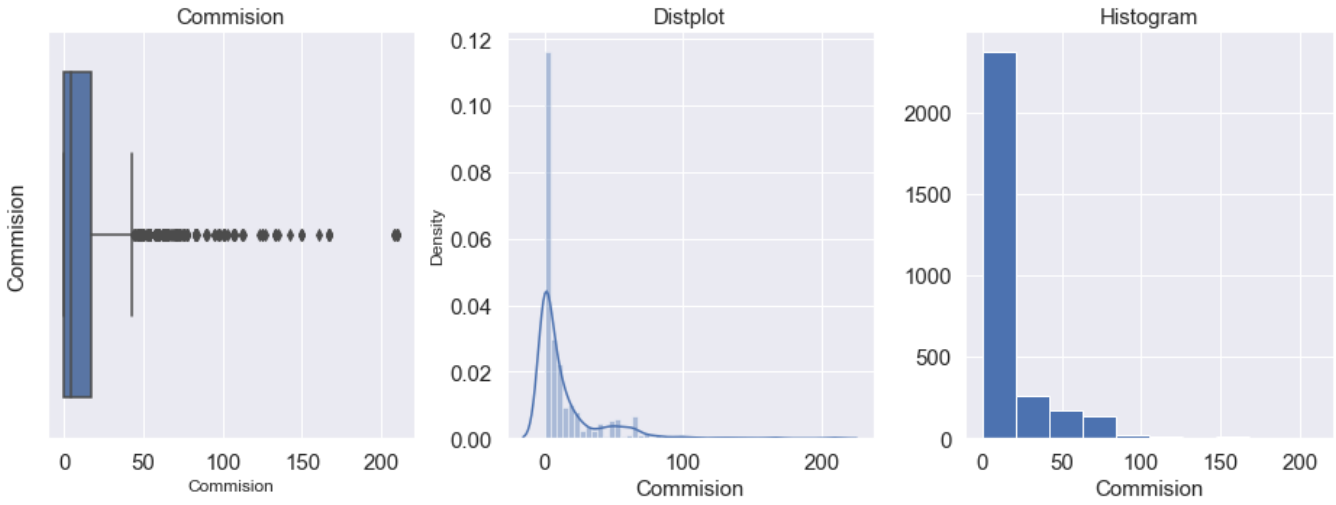
- Minimum Commision: 0.0
- Maximum Commision: 210.21
- Mean value: 14.529203333333266
- Median value: 4.63
- Standard deviation: 25.48145450662553
- Null values: Not found

- Commision - 1st Quartile (Q1) is: 0.0
- Commision – 3RD Quartile (Q3) is: 17.235
- Interquartile range (IQR) of Commision is 17.235
- Lower outliers in Commision: -25.8525
- Upper outliers in Commision: 43.0875

Outlier Details:

- Number of outliers in Commision upper: 362
- Number of outliers in Commision lower: 0
- Percentage of Outlier in Commision upper: 12.066666666666666 %
- Percentage of Outlier in Commision lower: 0.0 %

Below is the Box plot, Distplot and Hostogram for the variable 'Commision'.



DURATION Variable:

From Descriptive statistical summary:

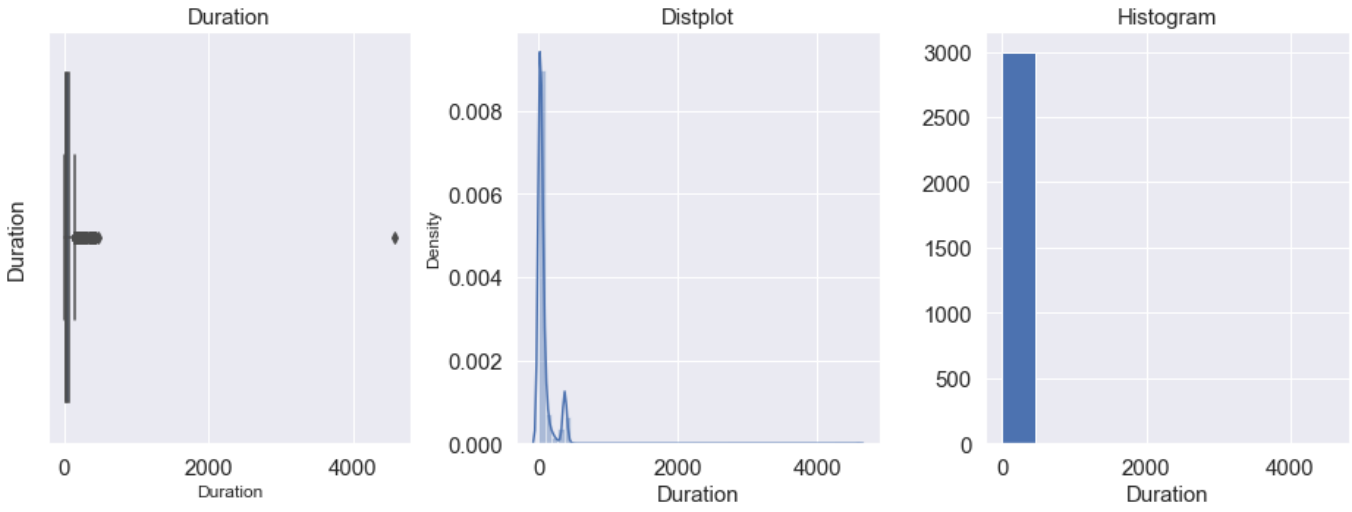
- Minimum Duration: -1
- Maximum Duration: 4580
- Mean value: 70.00133333333333
- Median value: 26.5
- Standard deviation: 134.05331313253495
- Null values: Not found

- Duration - 1st Quartile (Q1) is: 11.0
- Duration – 3rd Quartile (Q3) is: 63.0
- Interquartile range (IQR) of Duration is 52.0
- Lower outliers in Duration: -67.0
- Upper outliers in Duration: 141.0

Outlier Details:

- Number of outliers in Duration upper: 382
- Number of outliers in Duration lower: 0
- percentage of Outlier in Duration upper: 12.733333333333333 %
- percentage of Outlier in Duration lower: 0.0 %

Below is the Box plot, Distplot and Hostogram for the variable ‘Duration’.



SALES Variable:

From Descriptive statistical summary:

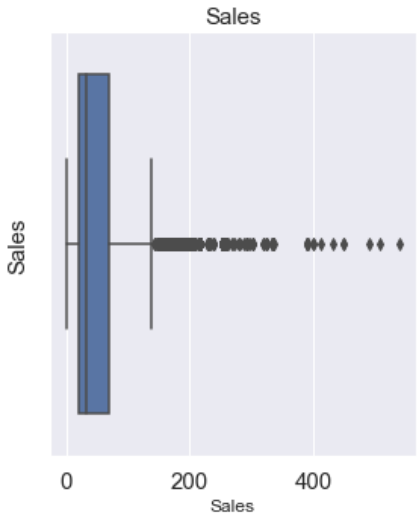
- Minimum Sales: 0.0
- Maximum Sales: 539.0
- Mean value: 60.24991333333344
- Median value: 33.0
- Standard deviation: 70.73395353143047
- Null values: Not found

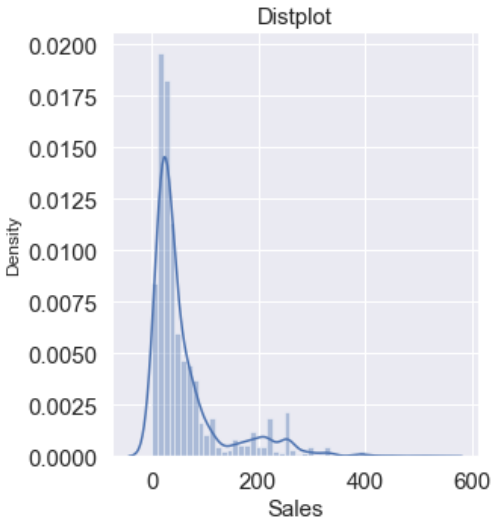
- Sales - 1st Quartile (Q1) is: 20.0
- Sales – 3rd Quartile (Q3) is: 69.0
- Interquartile range (IQR) of Sales is 49.0
- Lower outliers in Sales: -53.5
- Upper outliers in Sales: 142.5

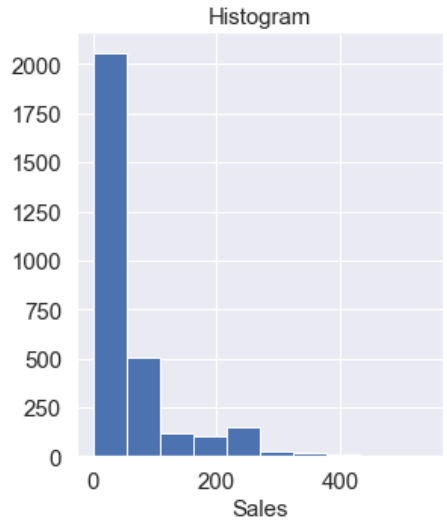
Outlier Details:

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

Below is the Box plot, Distplot and Hostogram for the variable 'Sales'.







Skewness observation through above details.

```
Duration     13.784681
Commision     3.148858
Sales         2.381148
Age           1.149713
dtype: float64
```

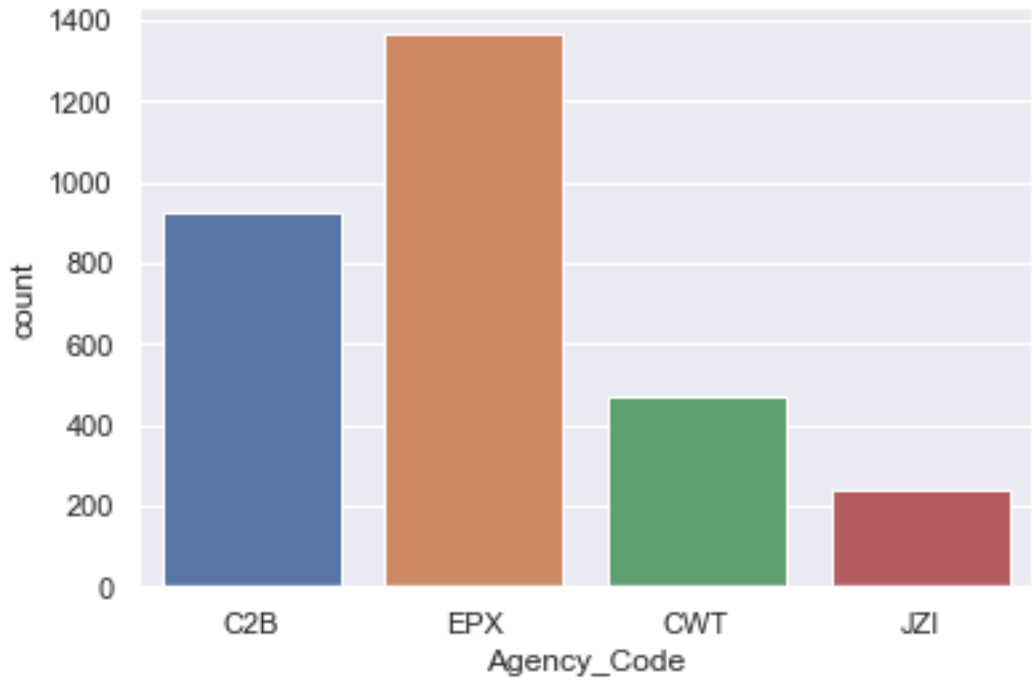
Observation:

- There are outliers in all the variables, but the sales and commision can be a genuine business value.
- Random Forest and CART can handle the outliers. Hence, Outliers are not treated, and we will keep the data as it is.
- We are treating the outliers for the ANN model to compare the same after the all the steps just for comparison.
- All the data are right skewed.

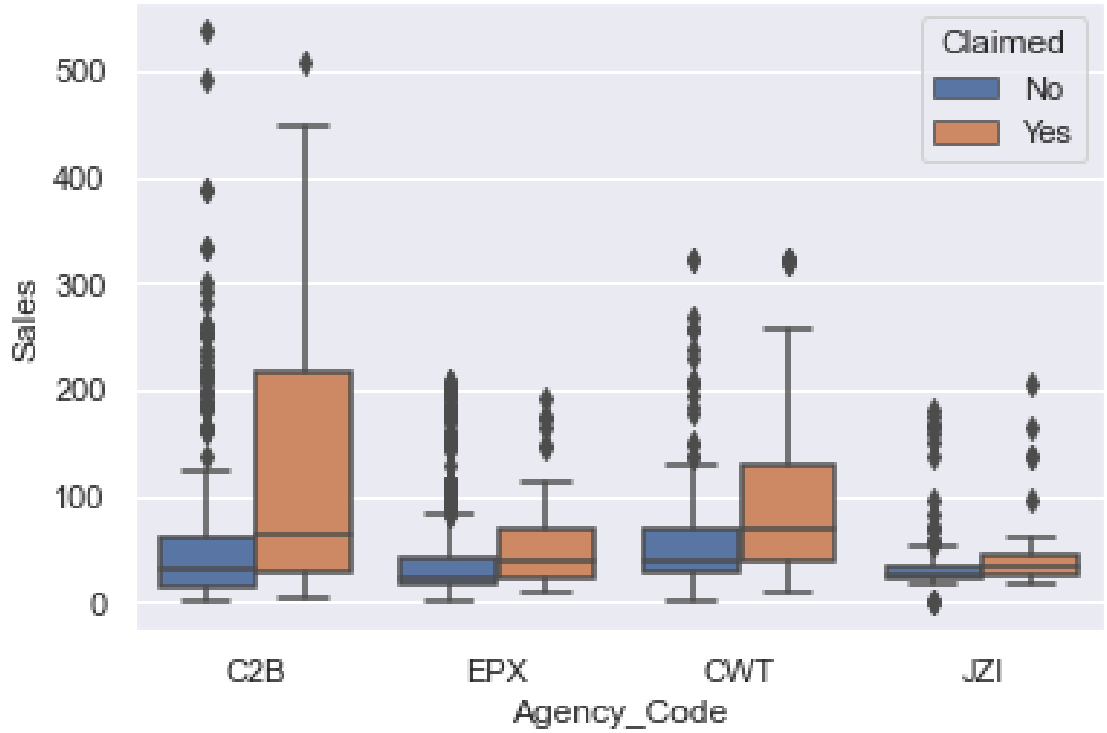
Categorical Variables

AGENT_CODE Variable:

Count plot:



Boxplot:

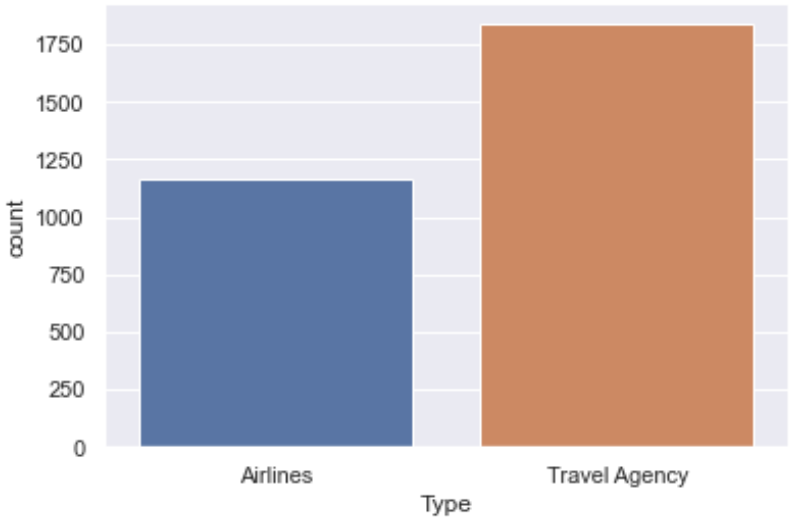


OBSERVATION:

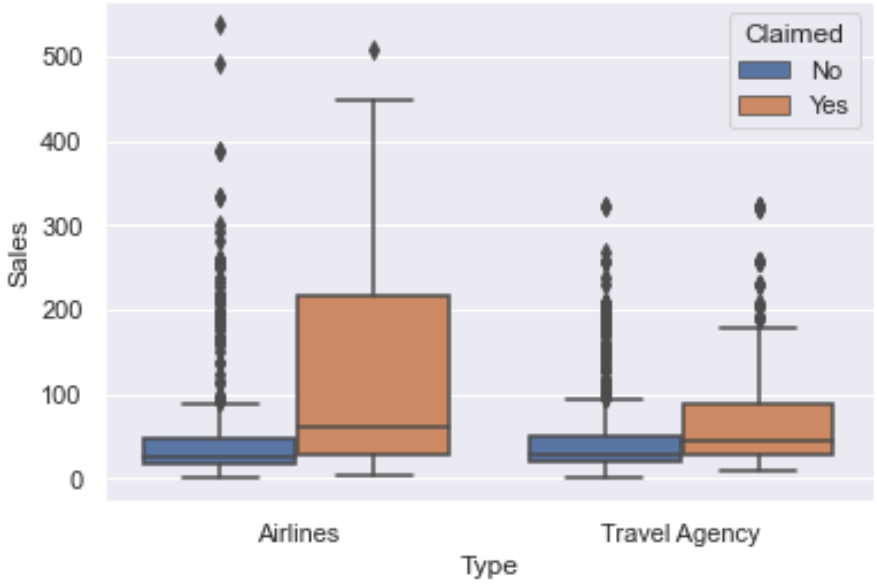
- There are highest number of records with agent_code EPX with 1365 records.
- All the Agent codes has outliers against Sales.

TYPE Variable:

Count plot:



Boxplot:

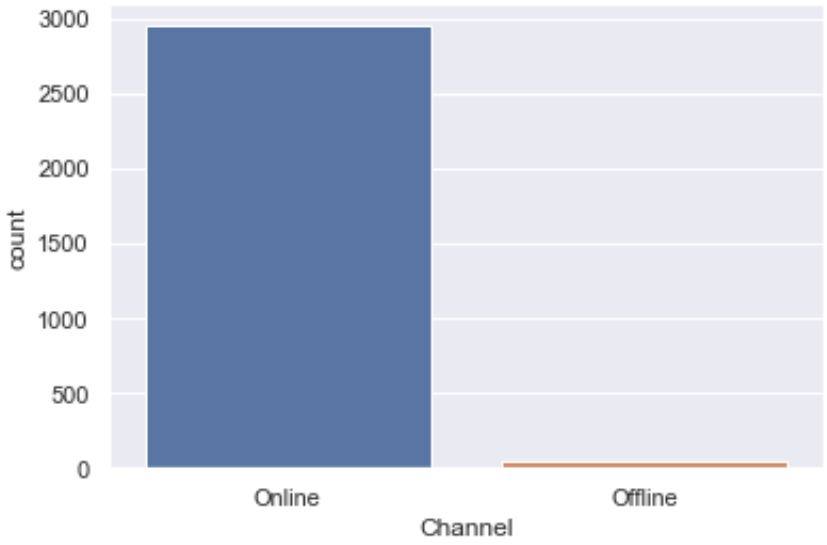


OBSERVATION:

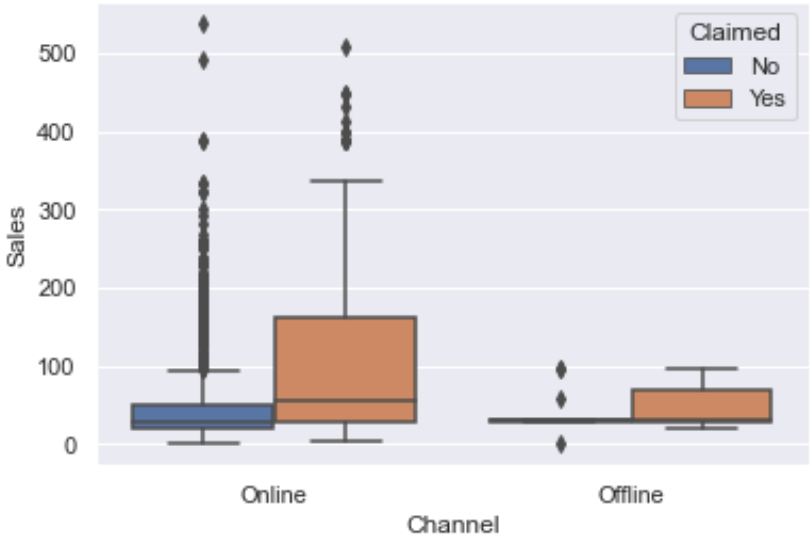
- There are highest number of records with Type 'Travel Agency' with 1837 records.
- Both the types have outliers against Sales.
- Number of successful claims are more in both the types.

CHANNEL Variable:

Count plot:



Boxplot:

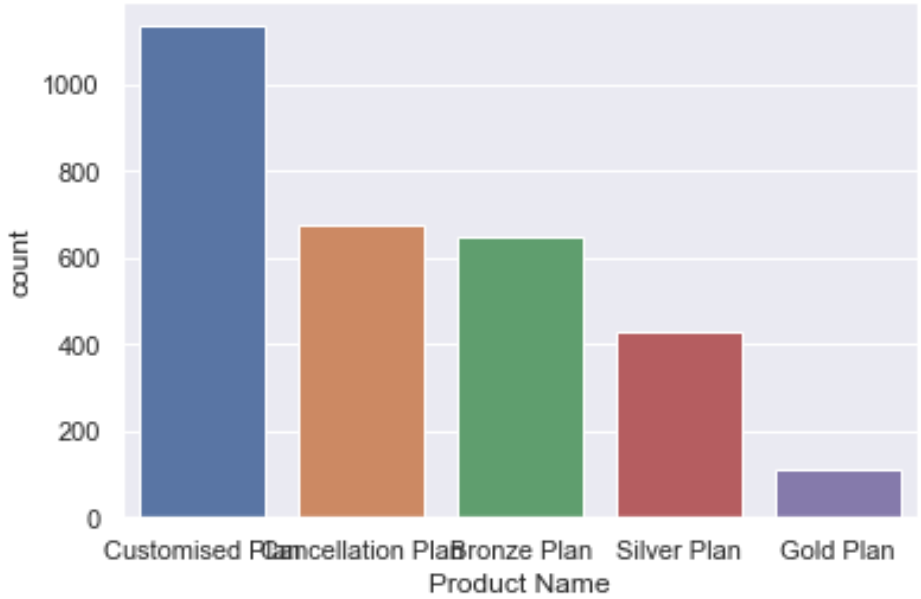


OBSERVATION:

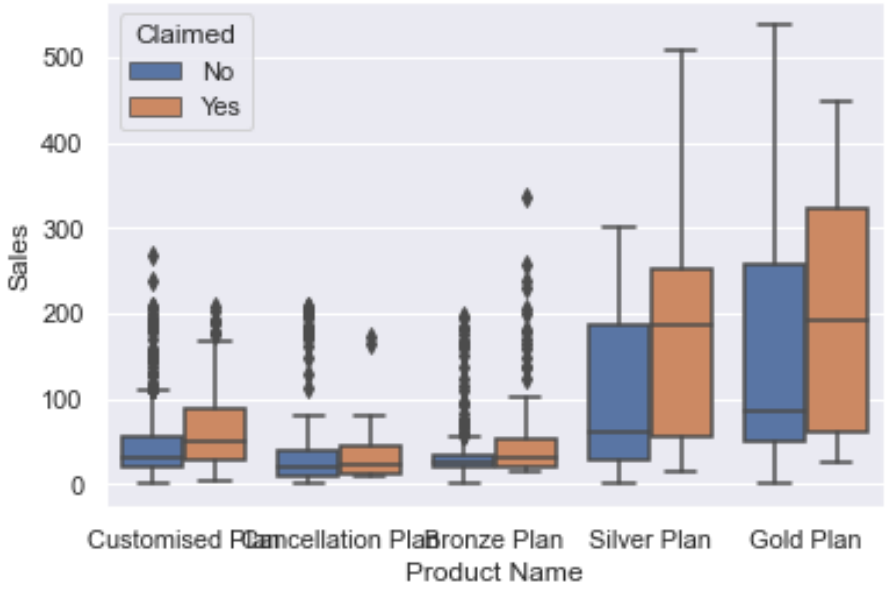
- There are highest number of records with channel ‘Online’ with 2954 records.
- Both the channels have outliers against Sales.

PRODUCT_NAME Variable:

Count plot:



Boxplot:

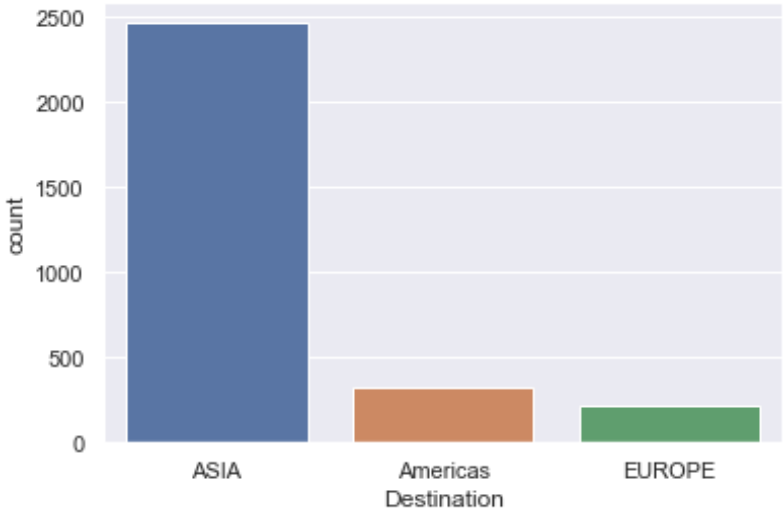


OBSERVATION:

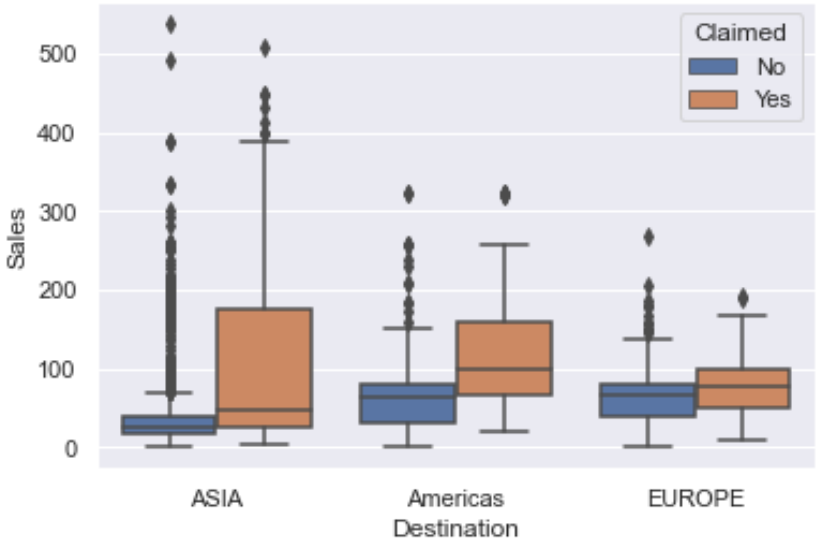
- There are highest number of records with Product name ‘Customised’ with 1136 records.
- All the product names have outliers other than ‘Silver Plan’ and ‘Gold Plan’ against the Sales record.

DESTINATION Variable:

Count plot:



Boxplot:



OBSERVATION:

- There are highest number of records with Destination name 'Asia' with 2456 records.
- All the Destinations have outliers.

Overall Observation:

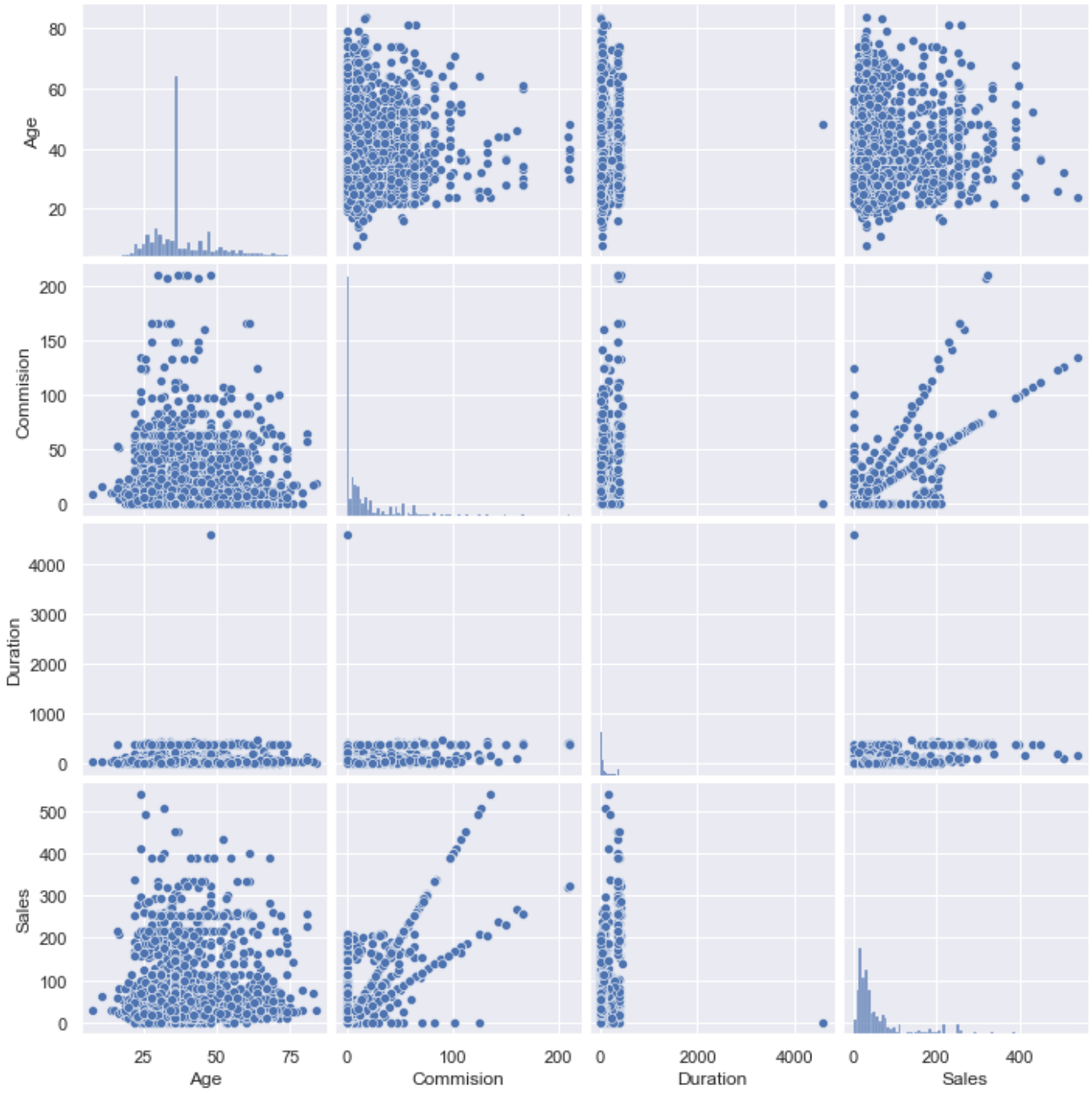
- **We plotted all the categorical variables against Sales and Claimed. We found that all the categorical variables having outliers mostly on upper level.**
- **We analyzed the countplot of all the categorical variables.**

Checking pairwise distribution of the continuous variables.

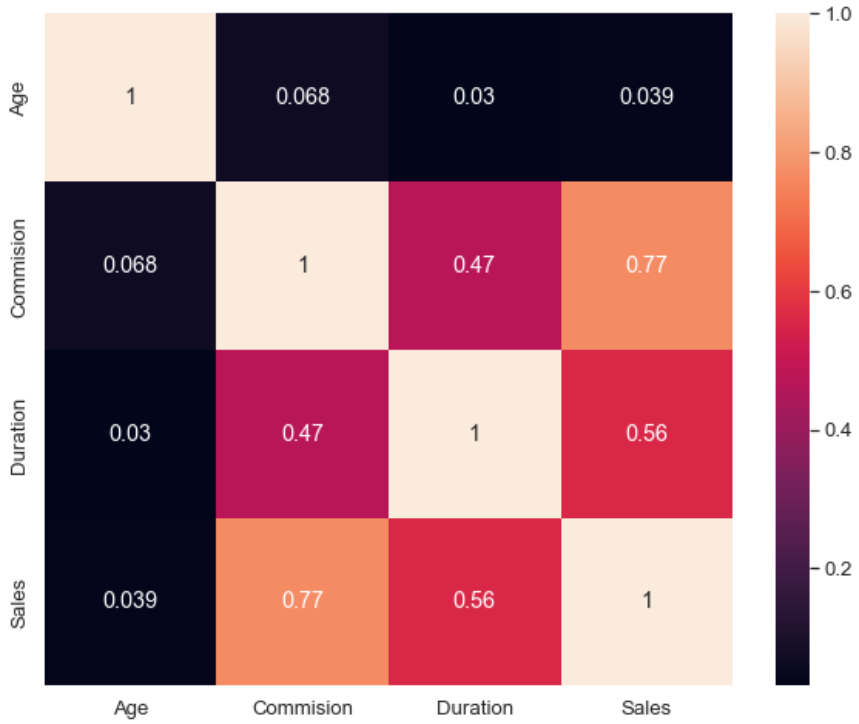
Correlation matrix:

	Age	Commision	Duration	Sales
Age	1.000000	0.067717	0.030425	0.039455
Commision	0.067717	1.000000	0.471389	0.766505
Duration	0.030425	0.471389	1.000000	0.558930
Sales	0.039455	0.766505	0.558930	1.000000

Below is the pairplot to represent the correlation between variables.



Visual representation of correlation between variables using the heat map.



OBSERVATION:

- Found comparatively good positive correlation between Commission and Sales.
- There is no negative correlation between any variables.

Converting all objects to categorical codes.

We are converting all the object type variables to Categorical types of variables.

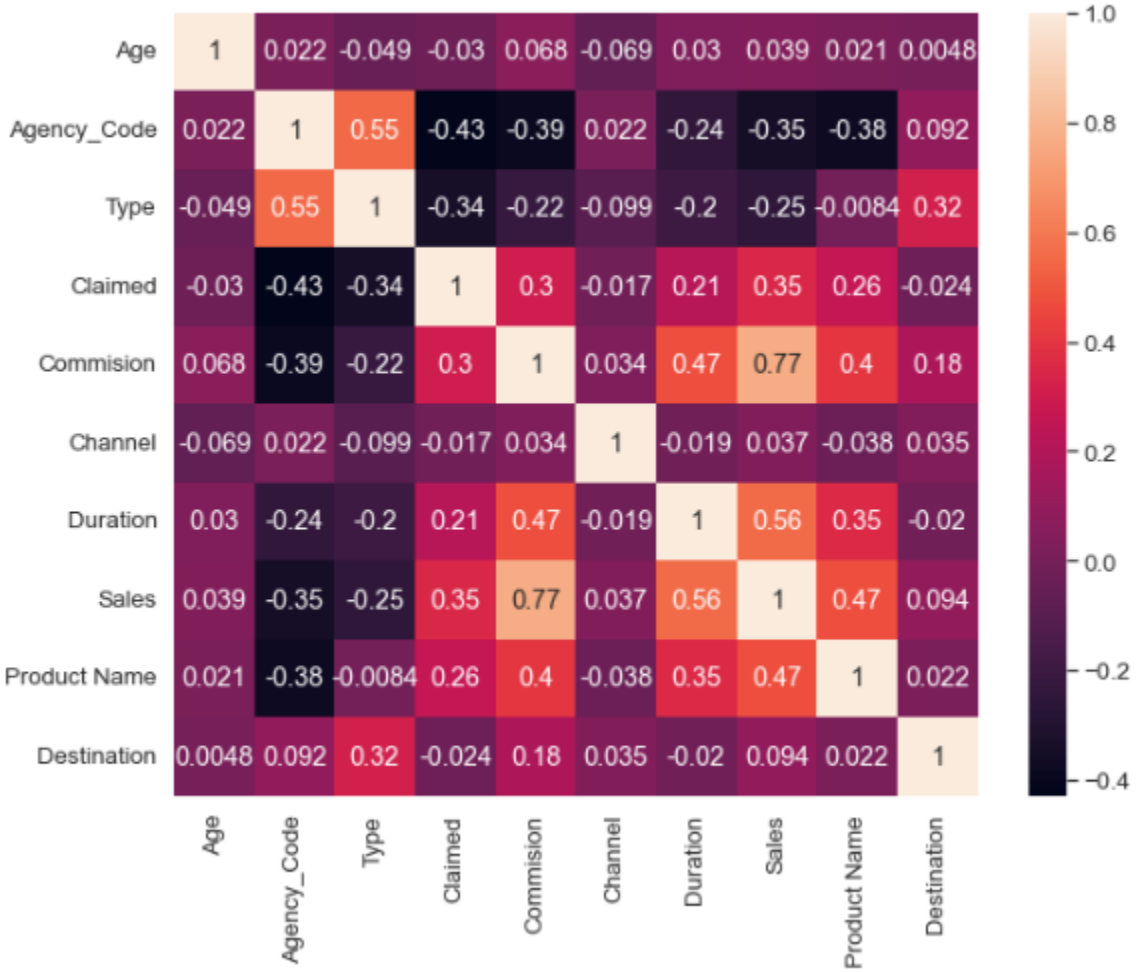
Below table represents the data types of all the variables after converting to categorical types.

```
<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
```

The sample data after data type conversion.

	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

Correlation between variables after conversion.



OBSERVATION:

- There are no new significant correlation variables pairs.

Let us investigate the percentage of claimed records in our data.

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

OBSERVATION:

- **69.2 % of the insurance are not claimed.**
- **Only 30.8% of insurance are claimed.**

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

We are separating independent variables and dependent variables and storing in separate dataframes.

Below is the new data frame sample without dependent variable 'Claimed'.

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

We are scaling the independent variable before performing the analysis. I am using zscore scaling method here. Below is the sample data after scaling.

	Age	Agency_Code	Type	Commision	Channel	Duration	Sales	Product Name	Destination
0	0.947162	-1.314358	-1.256796	-0.542807	0.124788	-0.470051	-0.816433	0.268835	-0.434646
1	-0.199870	0.697928	0.795674	-0.570282	0.124788	-0.268605	-0.569127	0.268835	-0.434646
2	0.086888	-0.308215	0.795674	-0.337133	0.124788	-0.499894	-0.711940	0.268835	1.303937
3	-0.199870	0.697928	0.795674	-0.570282	0.124788	-0.492433	-0.484288	-0.525751	-0.434646
4	-0.486629	1.704071	-1.256796	-0.323003	0.124788	-0.126846	-0.597407	-1.320338	-0.434646

- We are splitting the data into Training and Test data set. The 30% of the original data set data is split into Test data and remaining 70% data is train data.
- We are using parameter grid to obtain the finest values for parameters max_depth, min_sample_leaf and min_sample_split. In the 1st attempt we obtained with below best grid values.

```
{'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 50, 'min_samples_split': 450}

DecisionTreeClassifier(max_depth=10, min_samples_leaf=50, min_samples_split=450,
                       random_state=1)
```

- Further we are checking for finest values. Below are the results of each iteration until we find the finest parameter grid values.

2nd Iteration:

```
{'criterion': 'gini', 'max_depth': 5, 'min_samples_leaf': 20, 'min_samples_split': 150}

DecisionTreeClassifier(max_depth=5, min_samples_leaf=20, min_samples_split=150,
                       random_state=1)
```

3rd Iteration:

```
{'criterion': 'gini', 'max_depth': 4.85, 'min_samples_leaf': 44, 'min_samples_split': 250}

DecisionTreeClassifier(max_depth=4.85, min_samples_leaf=44,
                       min_samples_split=250, random_state=1)
```

4th Iteration:

```
{'criterion': 'gini', 'max_depth': 4.85, 'min_samples_leaf': 44, 'min_samples_split': 260}

DecisionTreeClassifier(max_depth=4.85, min_samples_leaf=44,
                       min_samples_split=260, random_state=1)
```

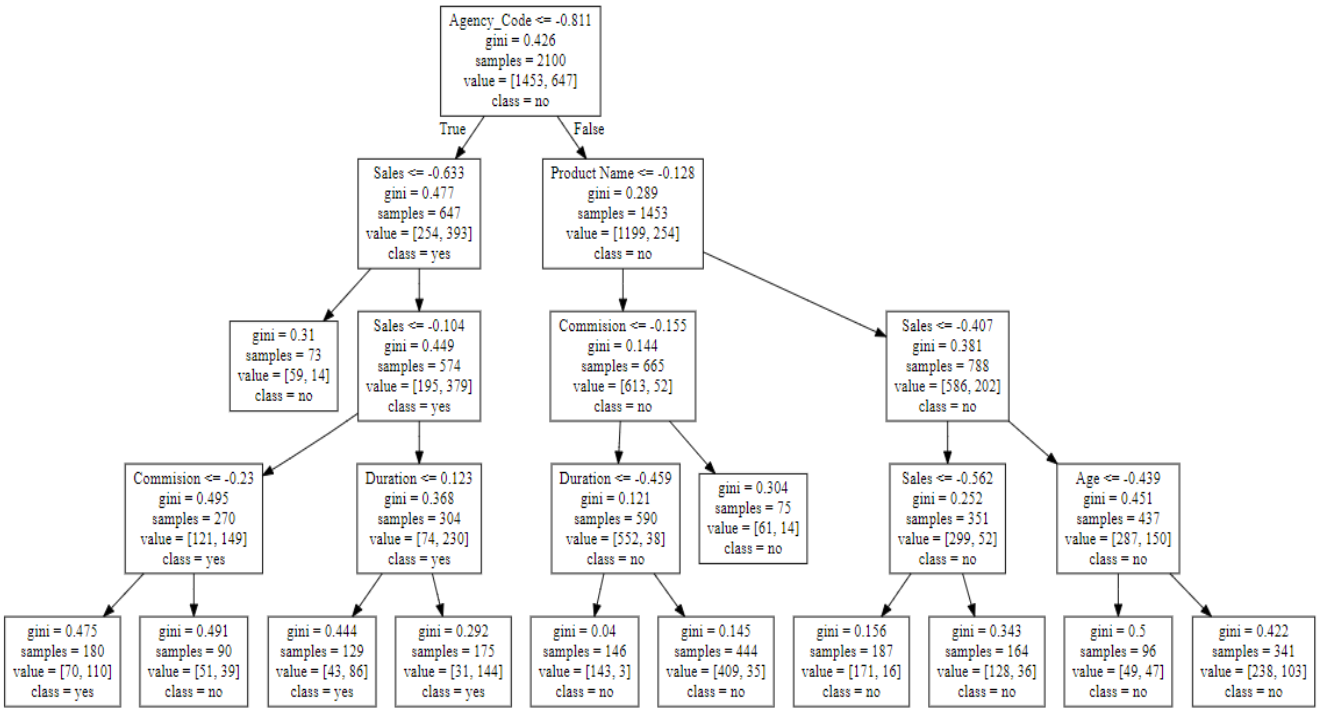
NOTE:

After multiple pruning process, we are finalizing max_depth= 4.85, min_samples_leaf= 44, min_samples_split= 260 to generate our decision tree.

Generating the Tree:

- We are using the best grid parameter from the above method and creating a dot file to generate the tree model.

- Below is the Tree generated from dot file (using Webgraphviz).



Importance of features in the tree building (The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance).

	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

Observation:

From the above result, Agent_code has the highest importance among other variables in building the tree model prediction.

- We have predicted the Training and Test dataset
- We also got the predicted classes and probes.

Below is the sample of predicted prob of test data.

	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

Building the Random Forest Classifier.

We are using parameter grid to obtain the finest values for parameters max_depth, max_features, min_sample_leaf, min_sample_split and n_eliminator. After multiple trial and error, we are finalizing the below grid parameters to build random forest model.

```
{'max_depth': 6, 'max_features': 3, 'min_samples_leaf': 8, 'min_samples_split': 46, 'n_estimators': 350}

RandomForestClassifier(max_depth=6, max_features=3, min_samples_leaf=8,
                       min_samples_split=46, n_estimators=350, random_state=1)
```

We are finalising the best grip random forest classifier

- We have predicted the Training and Test dataset
- We also got the predicted classes and probes.

Below is the sample of predicted prob of test data.

	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

Variable Importance via Random Forest:

	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

Observation:

Agency_code, Product_name, Sales and Commision plays the important rule in predicting the dependent variable.

Building a Neural Network Classifier.

We are using parameter grid to obtain the finest values for parameters hidden_layer_sizes, max_iter, solver and tol. After multiple trial and error, we are finalizing the below grid parameters to build neural network classifier.

```
MLPClassifier(hidden_layer_sizes=200, max_iter=2000, random_state=1, tol=0.01)
```

- We have predicted the Training and Test dataset
- We also got the predicted classes and probes.

Below is the sample of predicted prob of test data.

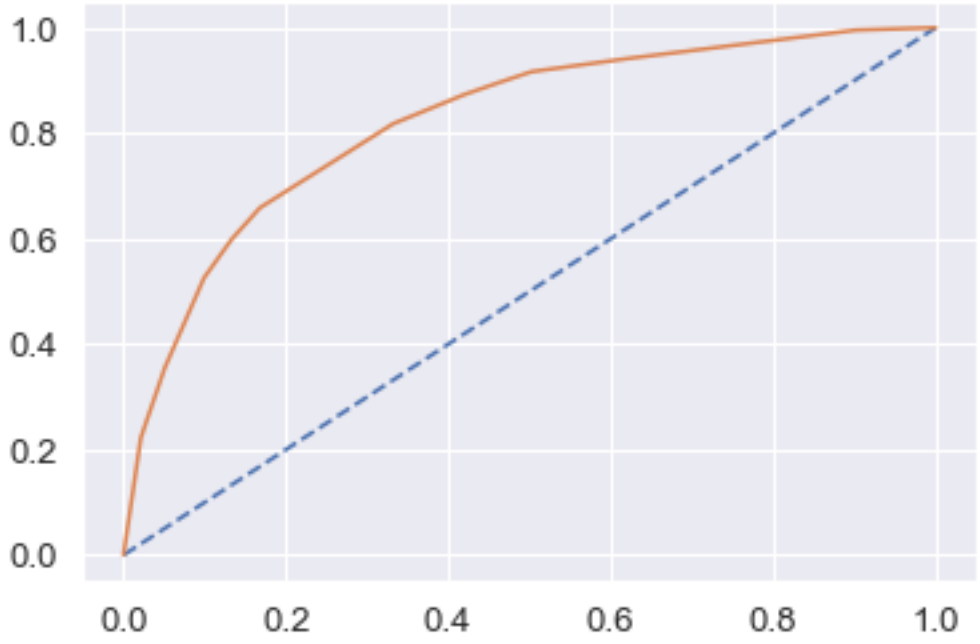
	0	1
0	0.822676	0.177324
1	0.933407	0.066593
2	0.918772	0.081228
3	0.688933	0.311067
4	0.913425	0.086575

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

CART - AUC and ROC for the training data:

Steps followed:

- Predicted the probabilities for train data using the best grid parameters obtained in CART method.
- Kept only the probabilities for the positive outcome and calculated the AUC
- Calculated the ROC curve and plot the ROC curve for the CART model.
- AUC for Train data is **0.823**
- Below is the ROC curve for CART model.

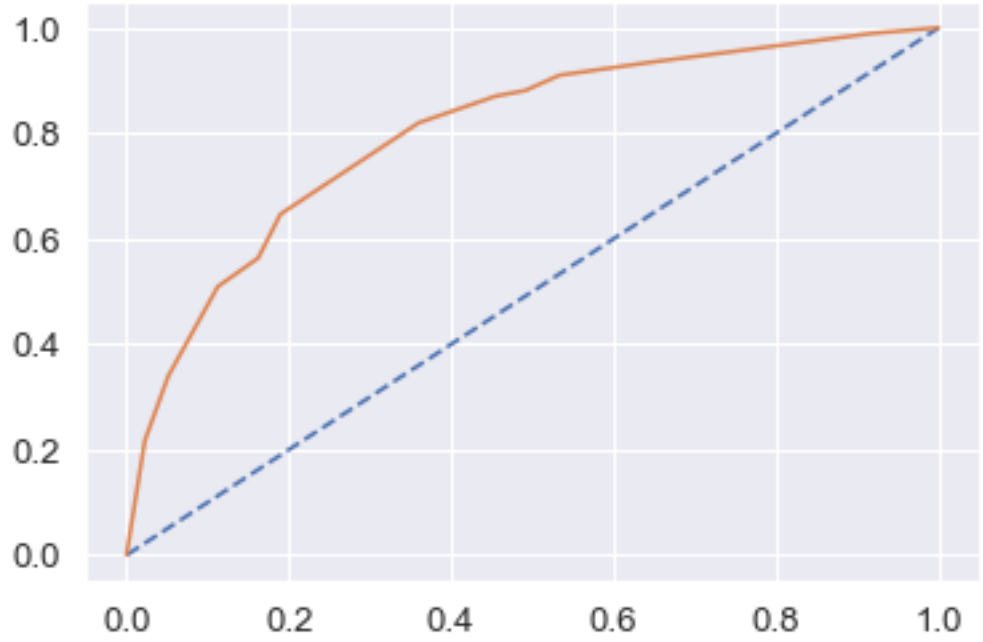


CART - AUC and ROC for the test data:

Steps followed:

- Predicted the probabilities for test data using the best grid parameters obtained in CART method.
- Kept only the probabilities for the positive outcome and calculated the AUC
- Calculated the ROC curve and plot the ROC curve for the CART model.
- AUC for Test data is **0.801**

- Below is the ROC curve for CART model.



CART Confusion Matrix and Classification Report for the training data:

Below confusion matrix for training data.

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

- True Positive: 1309
- True Negative: 340
- False Positive: 307
- False Negative: 144
- Train Data Accuracy: 0.78523

Below is the classification report for train data:

	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

Observation:

- cart train precision: 0.7
- cart train recall: 0.53
- cart train f1: 0.6
- cart train Accuracy ~79%

CART Confusion Matrix and Classification Report for the testing data:

Below confusion matrix for testing data.

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

- True Positive: 553
- True Negative: 141
- False Positive: 136
- False Negative: 70
- Train Data Accuracy: 0.7711

Below is the classification report for test data:

	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

Observation:

- cart train precision: 0.67
- cart train recall: 0.51
- cart train f1: 0.58
- cart train Accuracy ~77%

CART Conclusion

Train Data:

- cart train Accuracy ~79%
- cart train precision 0.7
- cart train recall 0.53
- cart train f1 0.6

Test Data:

- cart train Accuracy: ~77%
- cart train precision: 0.67
- cart train recall: 0.51
- cart train f1: 0.58

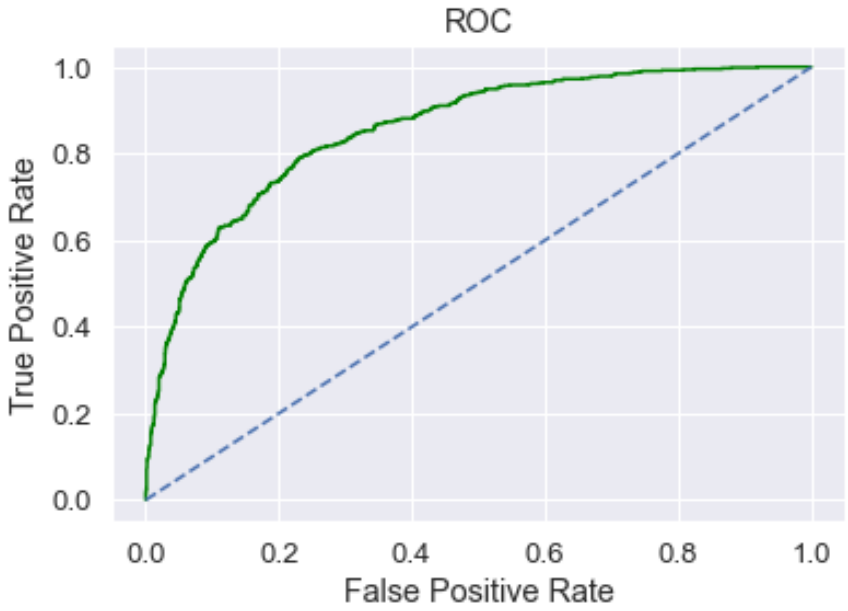
Training and Test set results are almost similar, and with the overall measures high, the model is a good model.

Agency_Code, Sales, Product Name and Commision are the most important variables for predicting the dependable variable 'Claimed'.

RandomForest - AUC and ROC for the training data:

Steps followed:

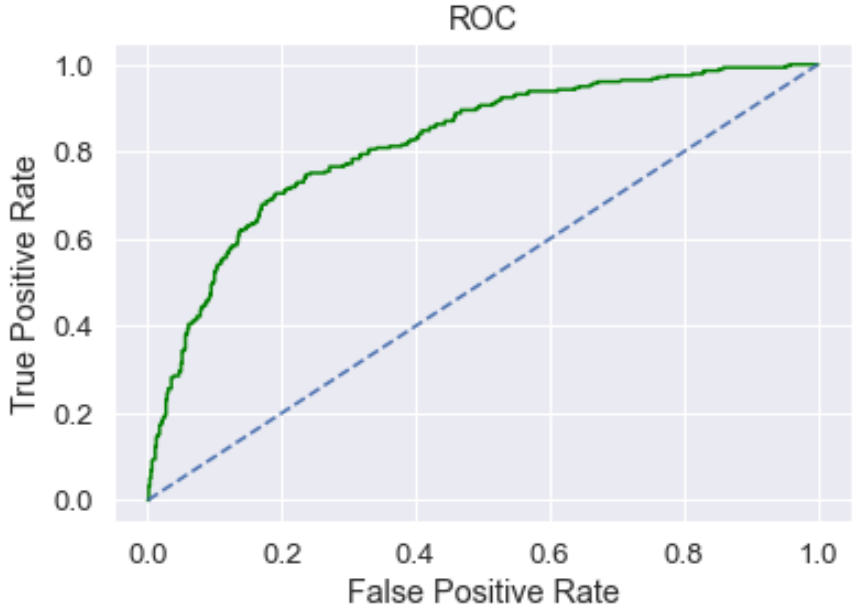
- Predicted the probabilities for train data using the best grid parameters obtained in RF method.
- Kept only the probabilities for the positive outcome and calculated the AUC
- Calculated the ROC curve and plot the ROC curve for the RF model.
- AUC for Train data is **0.8563**
- Below is the ROC curve for RF model.



Random Forest - AUC and ROC for the test data:

Steps followed:

- Predicted the probabilities for test data using the best grid parameters obtained in RF method.
- Kept only the probabilities for the positive outcome and calculated the AUC
- Calculated the ROC curve and plot the ROC curve for the RF model.
- AUC for Test data is **0.8181**
- Below is the ROC curve for RF model.



RF Confusion Matrix and Classification Report for the training data:

Below confusion matrix for training data.

```
array([[1297,  156],
       [ 255,  392]], dtype=int64)
```

- True Positive: 1297
- True Negative: 392
- False Positive: 255
- False Negative: 156
- Train Data Accuracy: 0.8042

Below is the classification report for train data:

	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

Observation:

- cart train precision: 0.72
- cart train recall: 0.61
- cart train f1: 0.66
- cart train Accuracy ~80%

RF Confusion Matrix and Classification Report for the testing data:

Below confusion matrix for testing data.

```
array([[550,  73],
       [121, 156]], dtype=int64)
```

- True Positive: 550
- True Negative: 156
- False Positive: 121
- False Negative: 73
- Train Data Accuracy: 0.7844

Below is the classification report for test data:

	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

Observation:

- cart train precision: 0.68
- cart train recall: 0.56
- cart train f1: 0.62
- cart train Accuracy ~78%

Random Forest Conclusion

Train Data:

- RF train Accuracy ~80%
- RF train precision 0.72
- RF train recall 0.61
- RF train f1 0.66

Test Data:

- RF train Accuracy: ~78%
- RF train precision: 0.68
- RF train recall: 0.56
- RF train f1: 0.62

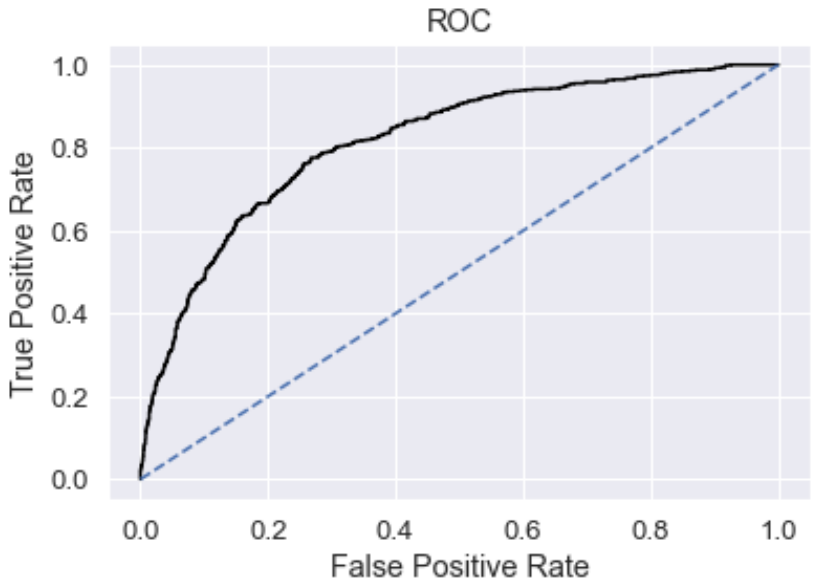
Training and Test set results are almost similar and with the overall measures high, the model is a good model.

Agency_Code, Product Name, Sales and Commision are the most important variables for predicting the dependable variable 'Claimed'.

Neural Network Model Performance Evaluation on Training data.

Steps followed:

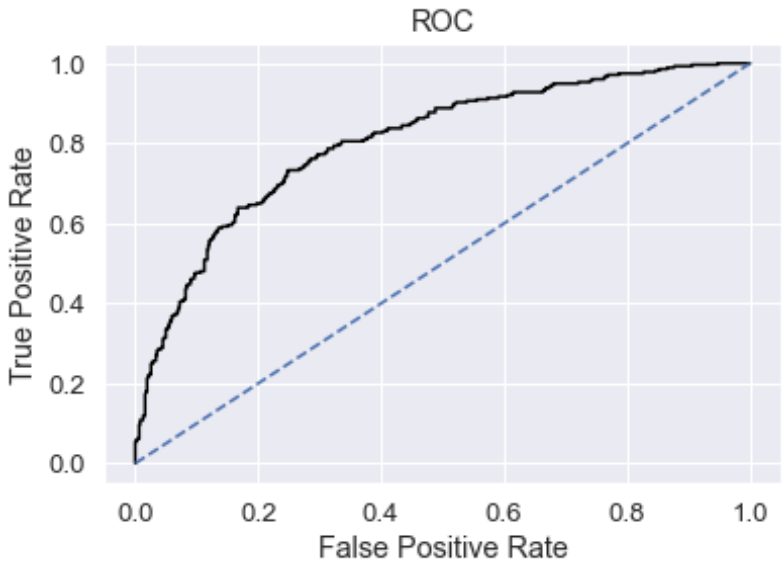
- Predicted the probabilities for train data using the best grid parameters obtained in NN method.
- Kept only the probabilities for the positive outcome and calculated the AUC
- Calculated the ROC curve and plot the ROC curve for the NN model.
- AUC for Train data is **0.8166**
- Below is the ROC curve for NN model.



Neural Network - AUC and ROC for the test data:

Steps followed:

- Predicted the probabilities for test data using the best grid parameters obtained in NN method.
- Kept only the probabilities for the positive outcome and calculated the AUC
- Calculated the ROC curve and plot the ROC curve for the NN model.
- AUC for Test data is **0.8044**
- Below is the ROC curve for NN model.



NN Confusion Matrix and Classification Report for the training data:

Below confusion matrix for training data.

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

- True Positive: 1298
- True Negative: 332
- False Positive: 315
- False Negative: 155
- Train Data Accuracy: 0.7761

Below is the classification report for train data:

	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

Observation:

- cart train precision: 0.68
- cart train recall: 0.51
- cart train f1: 0.59
- cart train Accuracy ~79%

NN Confusion Matrix and Classification Report for the testing data:

Below confusion matrix for testing data.

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

- True Positive: 553
- True Negative: 139
- False Positive: 138
- False Negative: 70
- Train Data Accuracy: 0.7688

Below is the classification report for test data:

	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

Observation:

- cart train precision: 0.67
- cart train recall: 0.50
- cart train f1: 0.57
- cart train Accuracy ~77%

Artificial Neural Network Conclusion

Train Data:

- NN train Accuracy ~78%
- NN train precision 0.68
- NN train recall 0.51
- NN train f1 0.59

Test Data:

- NN train Accuracy: ~77%

- NN train precision: 0.67
- NN train recall: 0.50
- NN train f1: 0.57

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

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

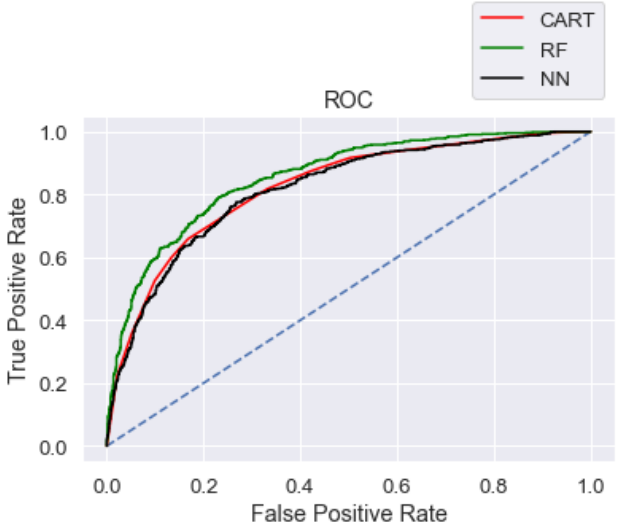
We are comparing Accuracy, AUC, Recall, Precision and F1 score of Test and Train data of CART, Random Forest and Artificial Neural Network models, below is the result.

	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.785	0.771	0.804	0.784	0.776	0.769
AUC	0.823	0.801	0.856	0.818	0.817	0.804
Recall	0.530	0.510	0.610	0.560	0.510	0.500
Precision	0.700	0.670	0.720	0.680	0.680	0.670
F1 Score	0.600	0.580	0.660	0.620	0.590	0.570

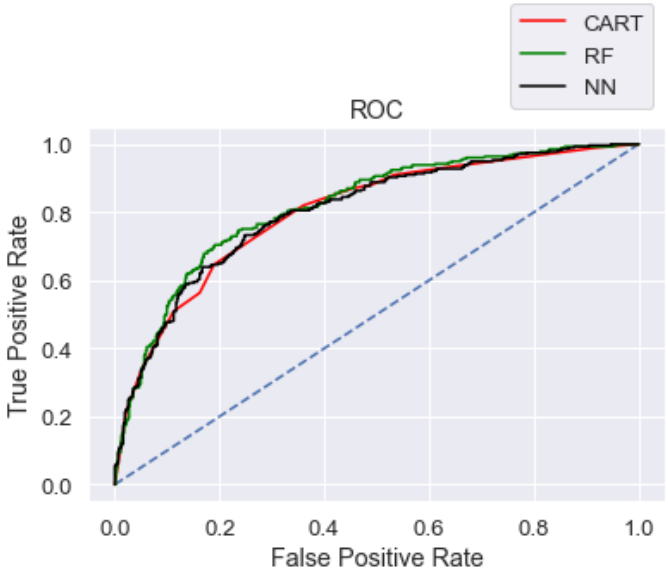
Observation:

Random Forest has better accuracy, precision, recall and f1 score better than other two CART & ANN.

We are plotting the ROC curve for the training data of all three models.



We are also plotting the ROC curve for the test data of all three models.



OBSERVATION:

- The steeper the ROC curve, stronger the model. If we compare the ROC curves of all three models, Random Forest model ROC curve steeper than other two models.
- Also, it has better accuracy, precision, recall, f1 score better than other two CART & ANN.
- Therefore, I am selecting RF model over CART and ANN.

2.5 Inference: Basis on these predictions, what are the business insights and recommendations

Basis on the analysis of Insuarance dataset, predictions by different models, below are the insights.

- I strongly recommended we collect more real time unstructured data and past data if possible.
- This is understood by looking at the insurance data by drawing relations between different variables such as day of the incident, time, age group, and associating it with other external information such as location, behavior patterns, weather information, airline/vehicle types, etc.

- Streamlining online experiences benefitted customers, leading to an increase in conversions, which subsequently raised profits.
- As per the data 90% of insurance is done by online channel.
- Other interesting fact, is almost all the offline business has a claimed associated, need to find why?
- Need to train the JZI agency resources to pick up sales as they are in bottom, need to run promotional marketing campaign or evaluate if we need to tie up with alternate agency
- Also based on the model we are getting 80%accuracy, so we need customer books airline tickets or plans, cross sell the insurance based on the claim data pattern.
- Other interesting fact is more sales happen via Agency than Airlines and the trend shows the claim are processed more at Airline. So, we may need to deep dive into the process to understand the workflow and why.

Key performance indicators (KPI)

The KPI's of insurance claims are:

- Reduce claims cycle time
- Increase customer satisfaction
- Combat fraud
- Optimize claims recovery
- Reduce claim handling costs

- Insights gained from data and AI-powered analytics could expand the boundaries of insurability, extend existing products, and give rise to new risk transfer solutions in areas like a non-damage business interruption and reputational damage.

----------THANK YOU----------