BUSINESS REPORT

ON

DATA MINING

By Kshitij Nishant

Problem 1: Clustering

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

1.1 Read the data and do exploratory data analysis (3 pts). Describe the data briefly. Interpret the inferences for each (3 pts). Initial steps like head() .info(), Data Types, etc . Null value check. Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Summary stats, Skewness, Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.

Answer:

After reading the data,

	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

- Data seems to be perfect
- No missing or null values in the dataset
- The info provided tells us that all values are float values

Description of data

	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

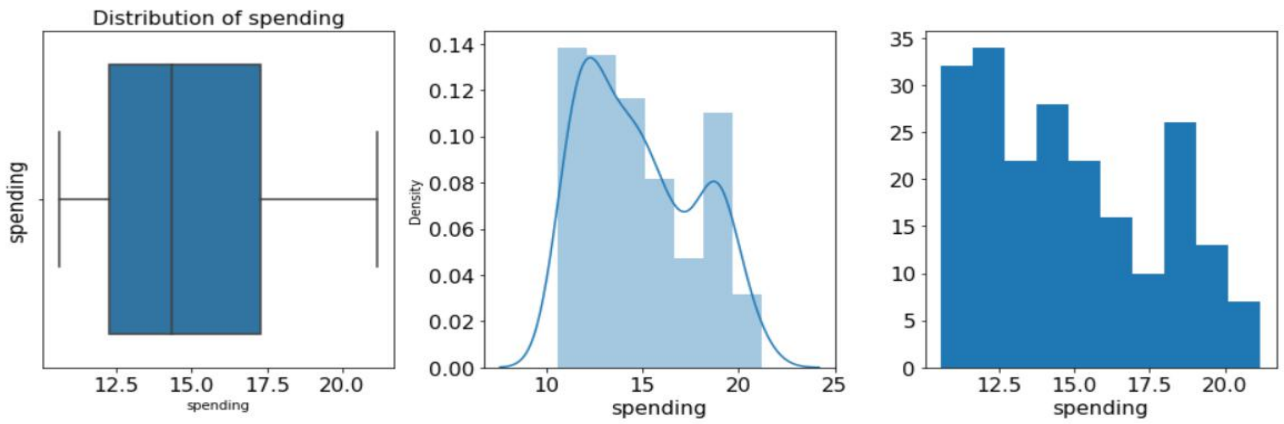
- Mean and Median seem to be almost equal.
- No duplicates in the data-set.
- The standard deviation of spending is higher than other variables.

Exploratory Data Analysis

A. Univariate analysis

1) Spending

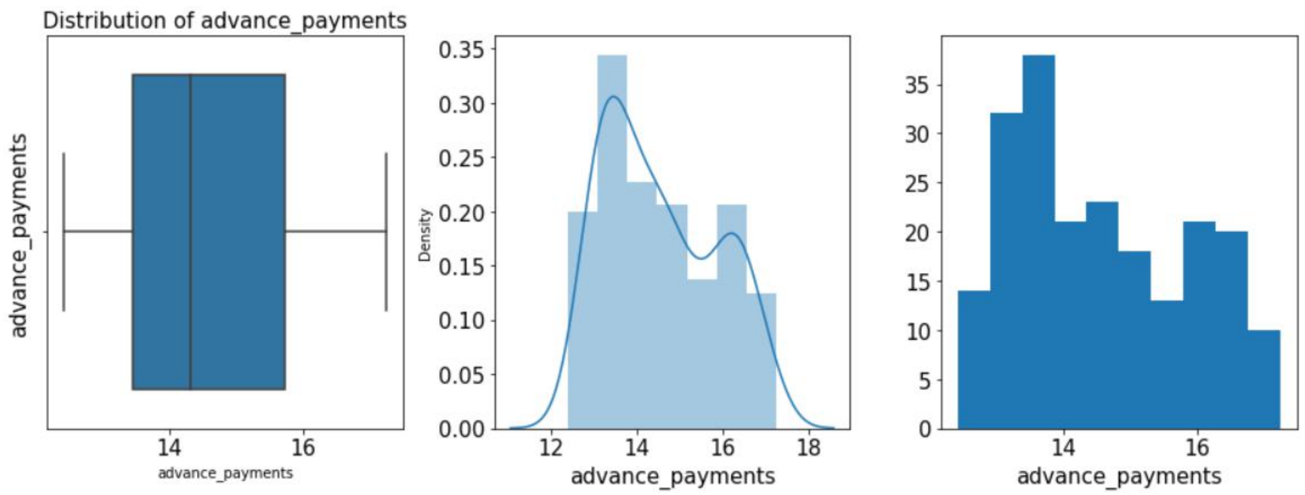
Range of values: 10.59
Minimum spending: 10.59
Maximum spending: 21.18
Mean value: 14.847523809523818
Median value: 14.355
Standard deviation: 2.909699430687361
Null values: False
spending - 1st Quartile (Q1) is: 12.27
spending - 3st Quartile (Q3) is: 17.305
Interquartile range (IQR) of spending is 5.035



Box Plot shows no outliers.
It is positively skewed: 0.399889

2) <u>Advance Payments</u>

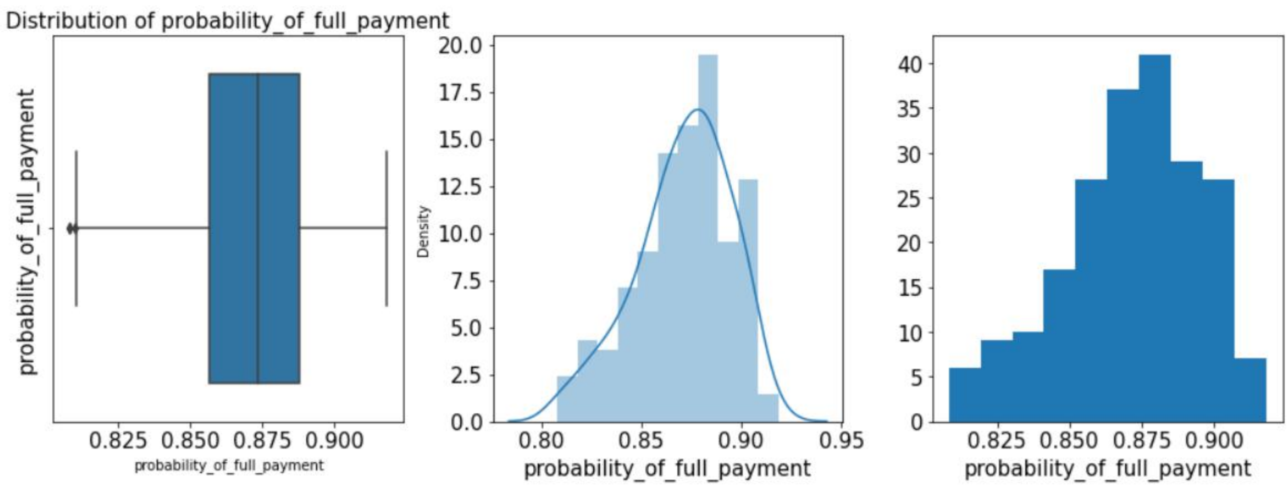
Range of values: 4.84
Minimum advance_payments: 12.41
Maximum advance_payments: 17.25
Mean value: 14.559285714285727
Median value: 14.32
Standard deviation: 1.305958726564022
Null values: False
advance_payments - 1st Quartile (Q1) is: 13.45
advance_payments - 3st Quartile (Q3) is: 15.715
Interquartile range (IQR) of advance_payments is 2.2650000000000006



Box Plot shows no outliers.
It is positively skewed: 0.386573

3) <u>Probability of Full Payment</u>

Range of values: 0.11019999999999996
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
probability_of_full_payment - 1st Quartile (Q1) is: 0.8569
probability_of_full_payment - 3st Quartile (Q3) is: 0.887775
Interquartile range (IQR) of probability_of_full_payment is 0.030874999999999986

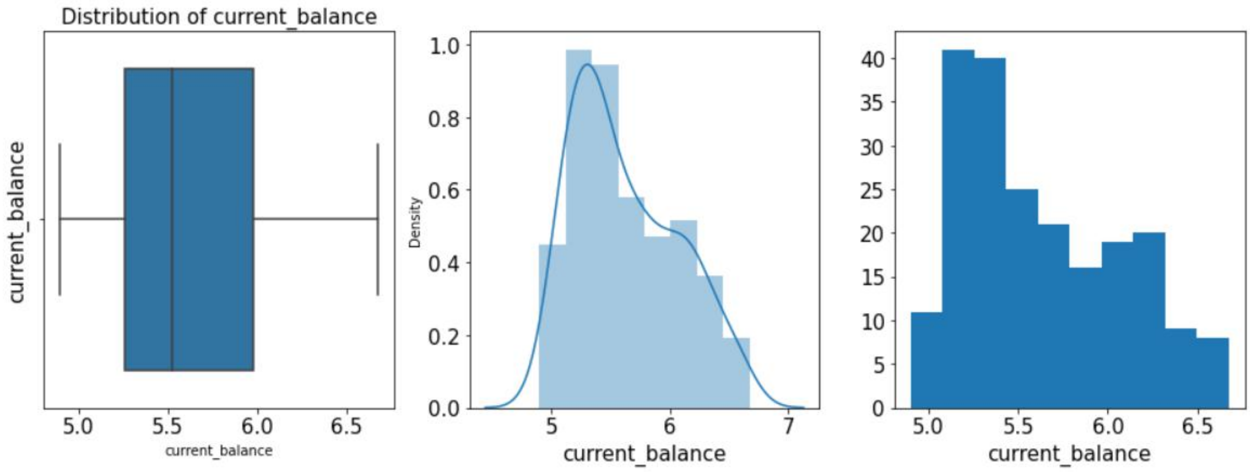


Box Plot shows few outliers.
It is negatively skewed: -0.537954

4) <u>Current Balance</u>

Range of values: 1.7759999999999998
Minimum current_balance: 4.899
Maximum current_balance: 6.675
Mean value: 5.628533333333335

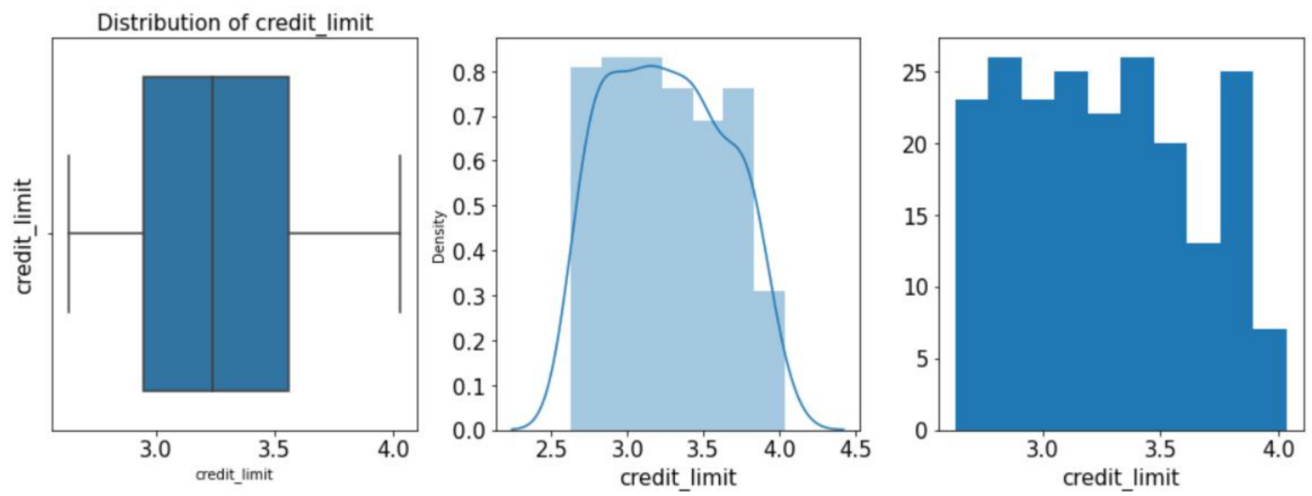
Median value: 5.5235
Standard deviation: 0.44306347772644944
Null values: False
current_balance - 1st Quartile (Q1) is: 5.26225
current_balance - 3st Quartile (Q3) is: 5.97975
Interquartile range (IQR) of current_balance is 0.7175000000000002



The box plot shows no outliers.
It is positively skewed: 0.525482

5) <u>Credit Limit</u>

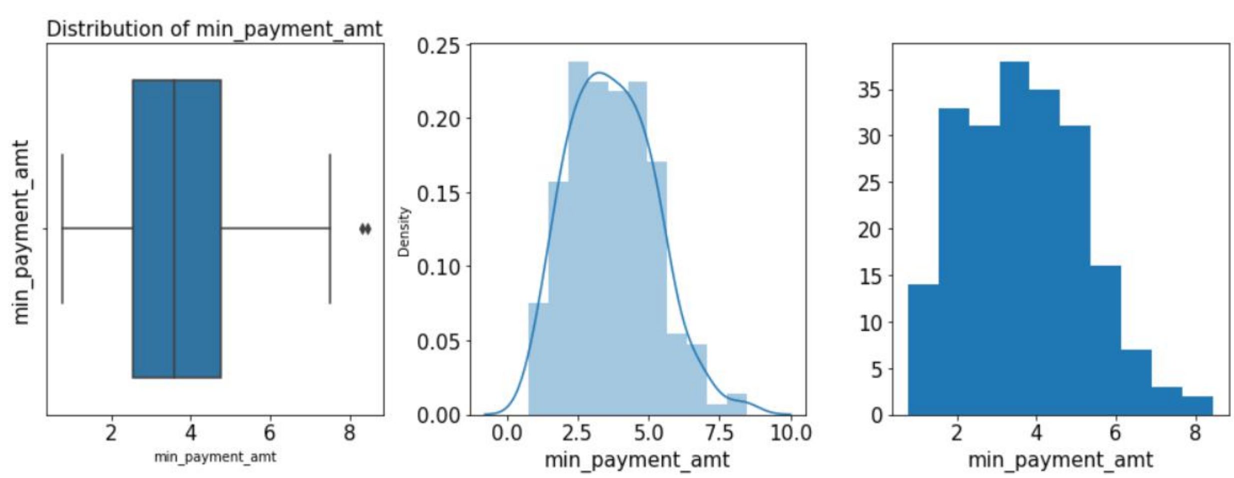
Range of values: 1.4030000000000005
Minimum credit_limit: 2.63
Maximum credit_limit: 4.033
Mean value: 3.258604761904763
Median value: 3.237
Standard deviation: 0.37771444490658734
Null values: False
credit_limit - 1st Quartile (Q1) is: 2.944
credit_limit - 3st Quartile (Q3) is: 3.56175
Interquartile range (IQR) of credit_limit is   0.61775



The box plot shows no outliers.
It is positively skewed: 0.134378

6) <u>Minimum Payment Amount</u>

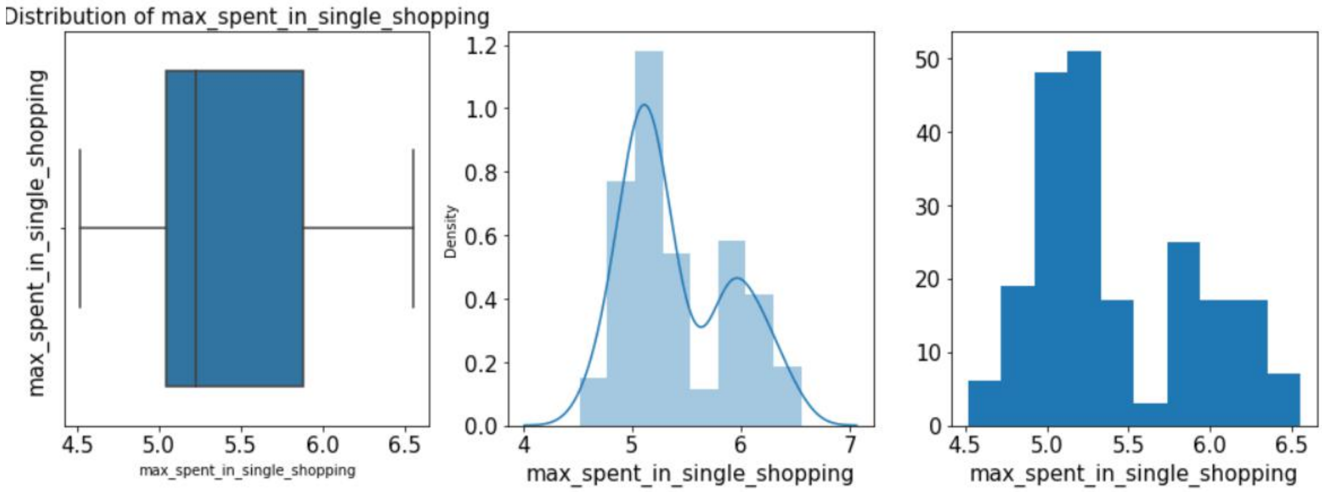
Range of values: 7.690899999999999
Minimum min_payment_amt: 0.7651
Maximum min_payment_amt: 8.456
Mean value: 3.7002009523809503
Median value: 3.599
Standard deviation: 1.5035571308217792
Null values: False
min_payment_amt - 1st Quartile (Q1) is: 2.5615
min_payment_amt - 3st Quartile (Q3) is: 4.76875
Interquartile range (IQR) of min_payment_amt is 2.2072499999999997



Box Plot shows few outliers.
It is positively skewed: 0.401667

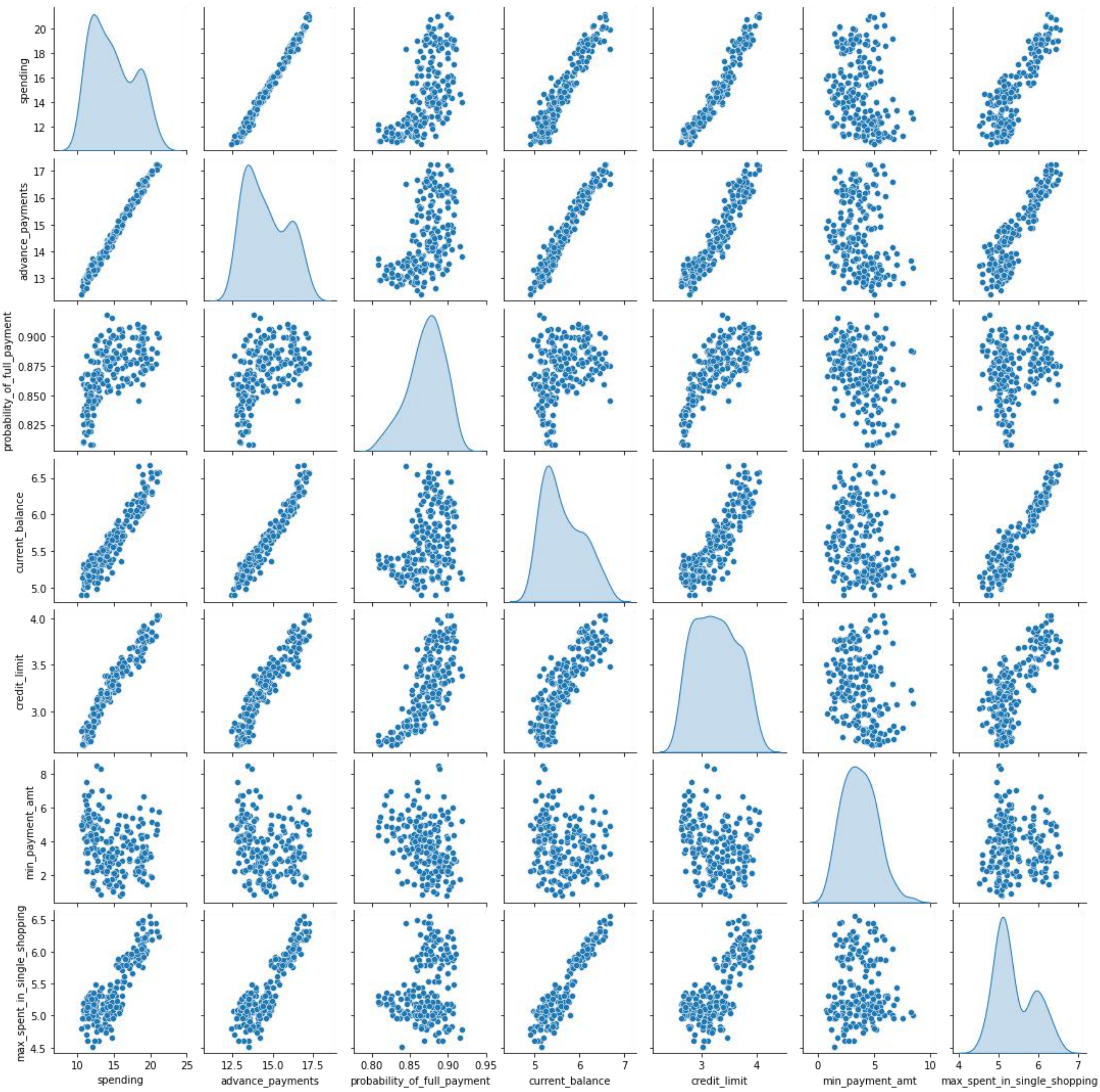
7) <u>Maximum Spent in a Single Shopping</u>

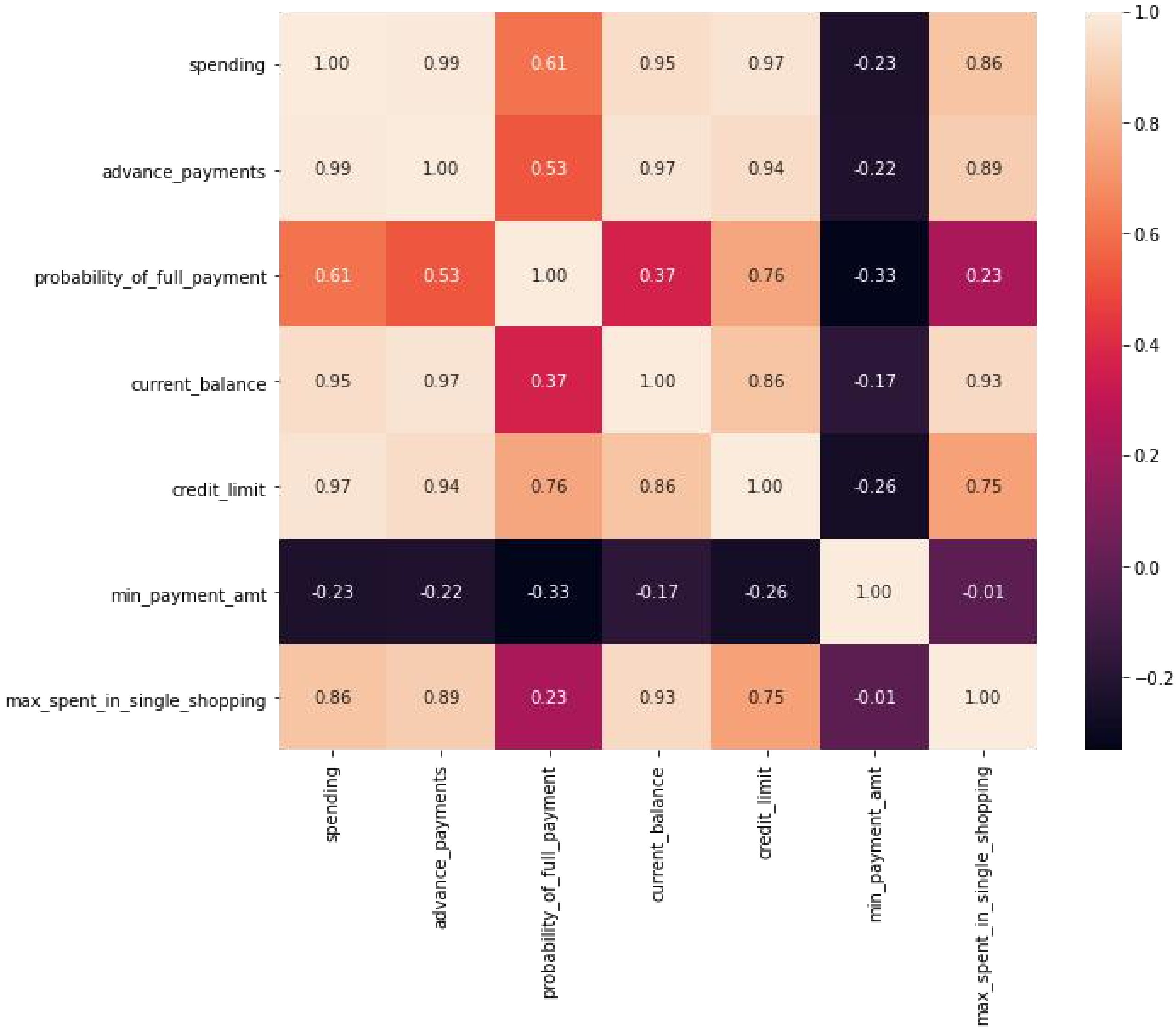
Range of values: 2.0309999999999997
Minimum max_spent_in_single_shopping: 4.519
Maximum max_spent_in_single_shoppings: 6.55
Mean value: 5.408071428571429
Median value: 5.223000000000001
Standard deviation: 0.49148049910240543
Null values: False
max_spent_in_single_shopping - 1st Quartile (Q1) is: 5.045
max_spent_in_single_shopping - 3st Quartile (Q3) is: 5.877
Interquartile range (IQR) of max_spent_in_single_shopping is 0.8319999999999999



Box Plot shows no outliers.
It is positively skewed: 0.561897

B. Multivariate Analysis





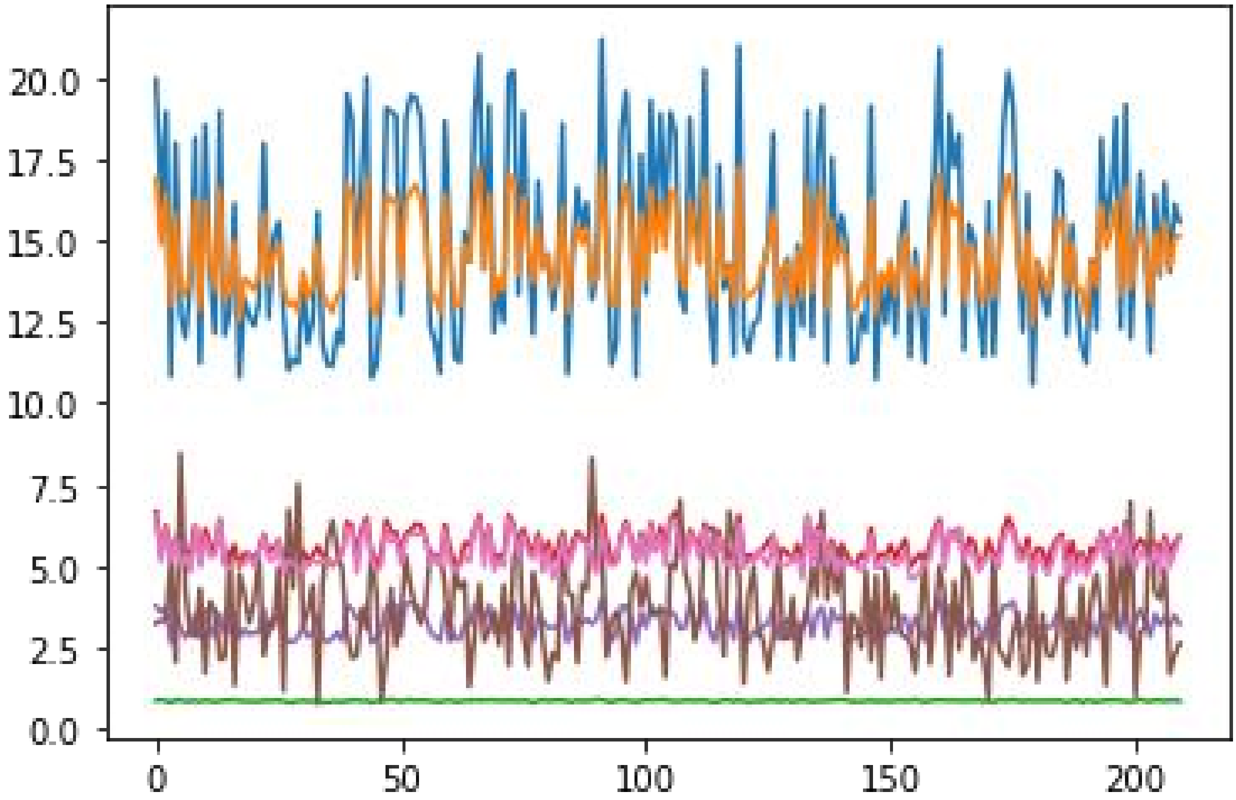
Strong positive correlation between

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

1.2 Do you think scaling is necessary for clustering in this case? Justify The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for example std dev, variance, etc. Should justify whether there is a necessity for scaling and which method is he/she using to do the scaling. Can also comment on how that method works.

<u>Answer</u>:

Scaling needs to be done as the values of the variables are different. spending, advance_payments are in different values and this may get more weight-age.
Scaling will have all the values in the relative same range.
I have used z-score to standardize the data to relative same scale -3 to +3.

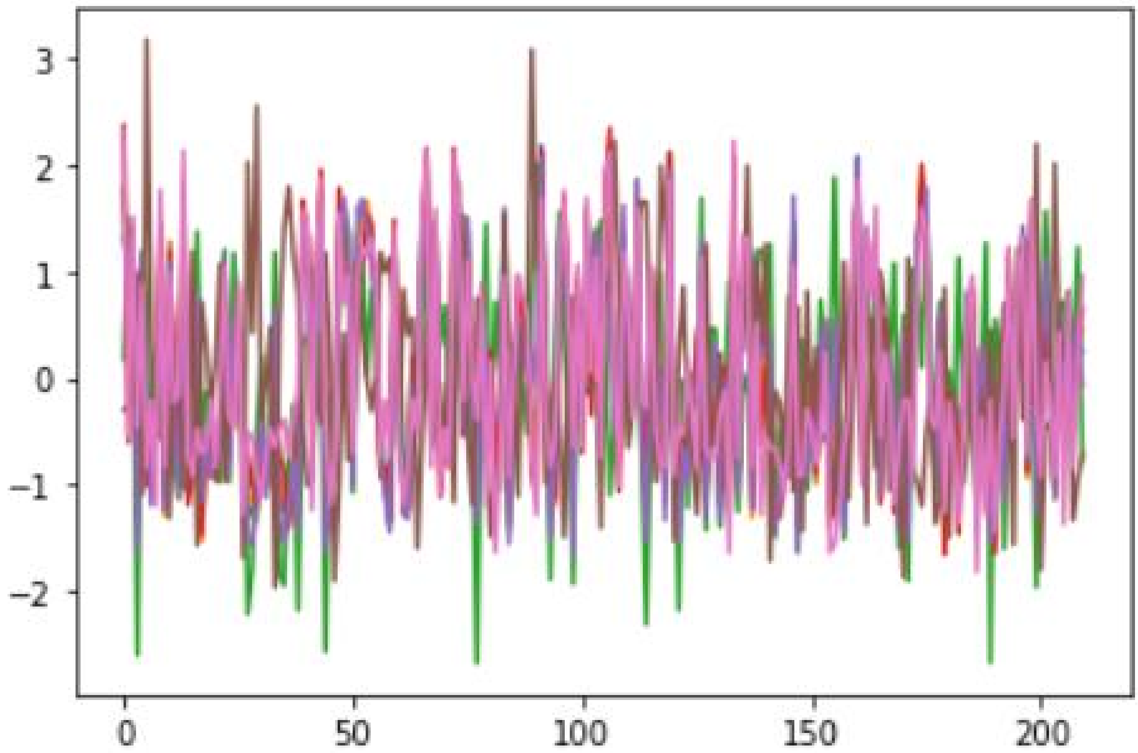


<u>Data prior to scaling</u>

As model works on distance based computations scaling is necessary for unscaled data.

	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

Scaled Data

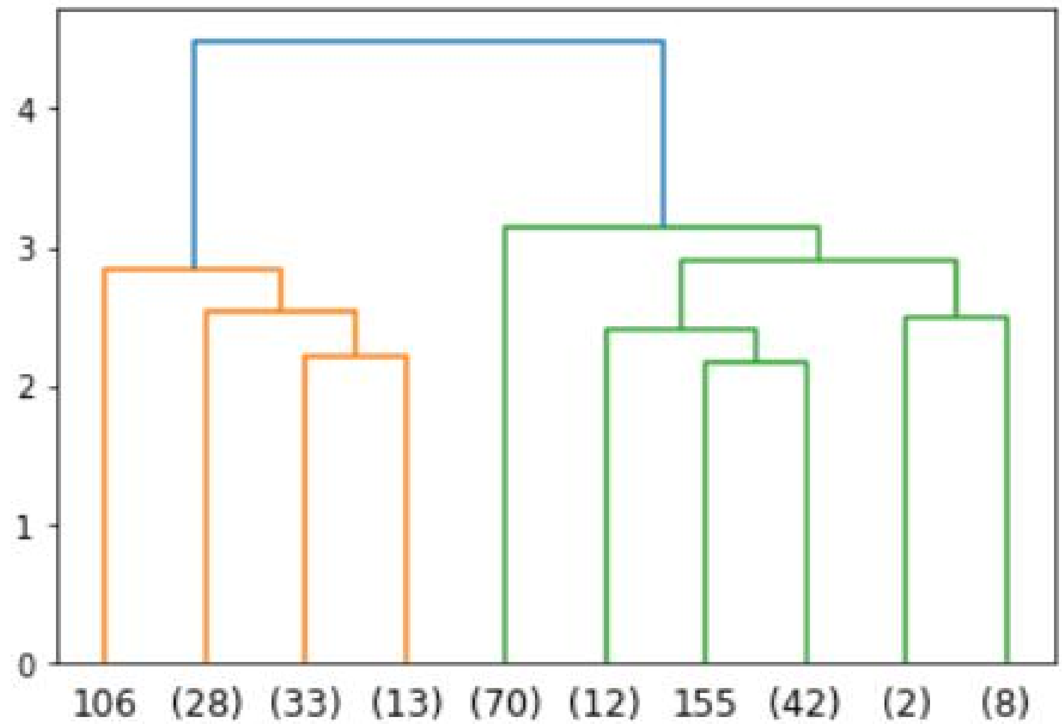


After Scaling

1.3 Apply hierarchical clustering to scaled data (3 pts). Identify the number of optimum clusters using Dendrogram and briefly describe them (4). Students are expected to apply hierarchical clustering. It can be obtained via Fclusters or Agglomerative Clustering. Report should talk about the used criterion, affinity and linkage. Report must contain a Dendrogram and a logical reason behind choosing the optimum number of clusters and Inferences on the dendrogram. Customer segmentation can be visualized using limited features or whole data but it should be clear, correct and logical. Use appropriate plots to visualize the clusters.

Answer:

We use dendrogram for visualisation,



We can now understand that all data clustered into 3 clusters.
Next, we map these cluster into the dataset,

```
clusters_3 = fcluster(link_method, 3, criterion='maxclust')
clusters_3
```

```
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 use the “maxclust” criterion for it.

	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

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

We can also see the cluster frequency in our dataset.

Then we do cluster profiling to understand the business problem,

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

cluster grouping based on the dendrogram, 3 or 4 looks good and based on the dataset had gone for 3 group cluster solution based on the hierarchical clustering.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 (2 pts). Apply elbow curve and silhouette score (3 pts). Interpret the inferences from the model (2.5 pts). K-means clustering code application with different number of clusters. Calculation of WSS(inertia for each value of k) Elbow Method must be applied and visualized with different values of K. Reasoning behind the selection of the optimal value of K must be explained properly. Silhouette Score must be calculated for the same values of K taken above and commented on. Report must contain logical and correct explanations for choosing the optimum clusters using both elbow method and silhouette scores. Append cluster labels obtained from K-means clustering into the original data frame. Customer Segmentation can be visualized using appropriate graphs.

<u>Answer:</u>

I decided to go with 3 clusters at first according to the dendrogram, then applied K-means technique to scaled data:

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

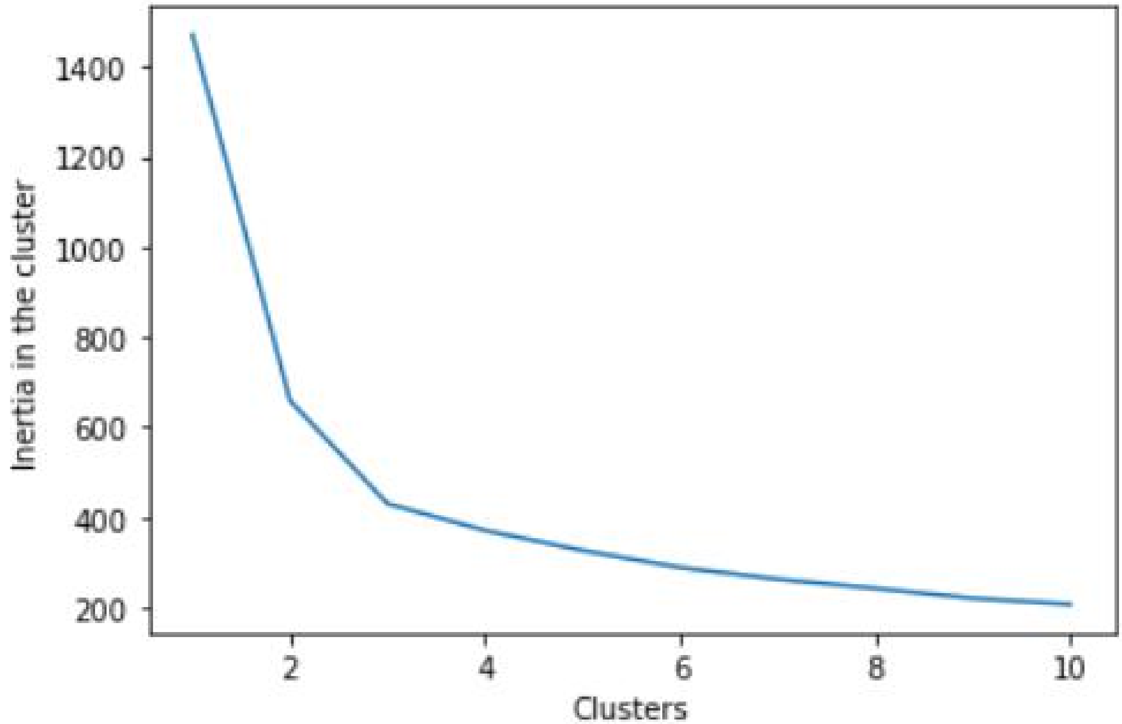
Here, we have 3 clusters: 0,1,2

To find the optimal number of clusters, we preform K-elbow method.
To find optimal number of clusters, I ran a loop to check for inertia value of clusters 1 to 11:

```
wss
```

```
[1469.9999999999995,
 659.1717544870411,
 430.6589731513064,
 371.30172127754 2,
 326.93482873449955,
 289.3608470485395,
 263.2599534317033,
 242.9918367686723 6,
 220.9097676522645 8,
 207.7985163988968 8]
```

The elbow curve also shown that there is no significant value drop after 3 clusters:



I saw that inertia value for 3 clusters also looks good enough.
So I checked for it's Silhouette score as well:

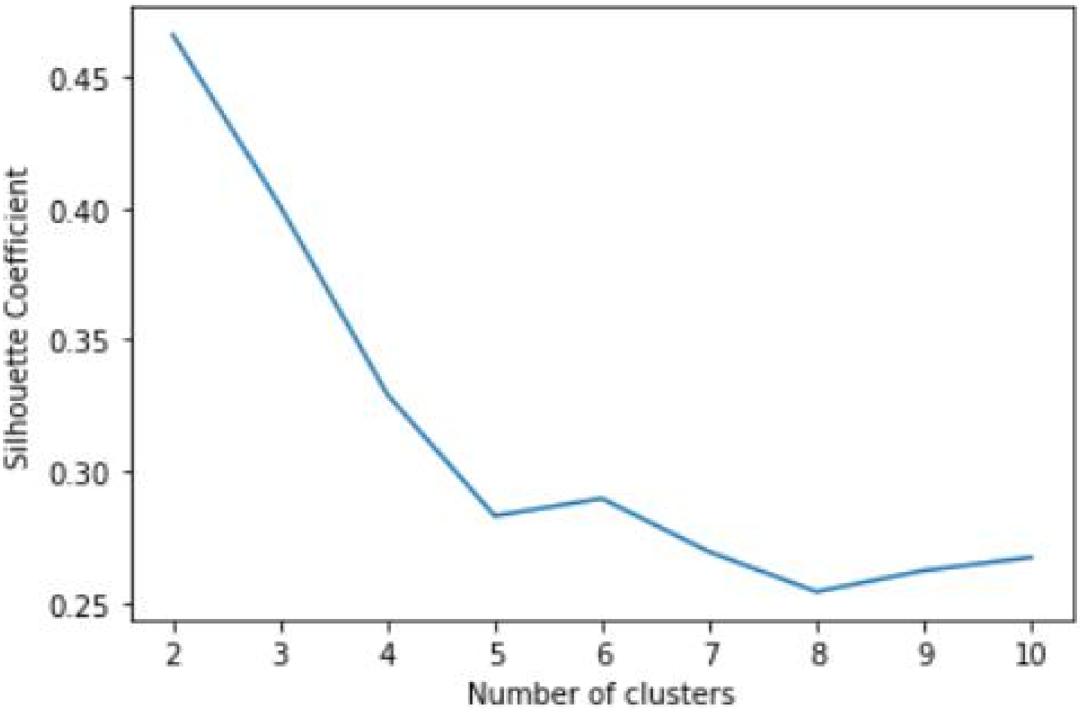
```
silhouette_score(scaled_df1,labels_3)
```

```
0.4007270552751298 6
```

I also compared it to other silhouette scores from 2 to 11 clusters:

```
scores
```

```
[0.46577247686580914,
 0.40072705527512986,
 0.3291966792017613,
 0.28316654897654814,
 0.2897583830272519,
 0.26948443551168536,
 0.2543731602750563,
 0.2623959398663564,
 0.2673980772529918]
```



So, I decided to go with 3 clusters.

This is the dataset with individual silhouette scores:

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	0	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	0	0.637784
10.83	12.96	0.8099	5.278	2.641	5.182	5.185	1	0.512458
17.99	15.86	0.8992	5.890	3.694	2.068	5.837	0	0.362276

Distribution of dataset according to 3 clusters:

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

Cluster frequency of dataset.

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

3 group cluster via Kmeans.
(I changed the numbering from 0,1,2 to 1,2,3 for clearer understanding)

1.5 Describe cluster profiles for the clusters defined (2.5 pts). Recommend different promotional strategies for different clusters in context to the business problem in-hand (2.5 pts). After adding the final clusters to the original dataframe, do the cluster profiling. Divide the data in the finalyzed groups and check their means. Explain each of the group briefly. There should be at least 3-4 Recommendations. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks will only be allotted if the recommendations are correct and business specific. variable means. Students to explain the profiles and suggest a mechanism to approach each cluster. Any logical explanation is acceptable.

<u>Answer:</u>

Cluster Profiling:

```
#transposing the cluster via Kmeans
cluster_3_T = kmeans_mean_cluster.T
cluster_3_T
```

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

```
##transposing the cluster via Kmeans
aggdata.T
```

clusters-3	1	2	3
spending	18.129200	11.916857	14.217077
advance_payments	16.058000	13.291000	14.195846
probability_of_full_payment	0.881595	0.846766	0.884869
current_balance	6.135747	5.258300	5.442000
credit_limit	3.648120	2.846000	3.253508
min_payment_amt	3.650200	4.619000	2.768418
max_spent_in_single_shopping	5.987040	5.115071	5.055569
Freq	75.000000	70.000000	65.000000

1st dataset is based on Kmeans clustering and 2nd dataset is based on hierarchical clustering with which we infer:

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

Promotional strategies for each cluster:

Group 1 : High Spending Group

1) Giving any reward points might increase their purchases.

2) maximum max_spent_in_single_shopping is high for this group, so can be offered discount/offer on next transactions upon full payment.

3) Increase there credit limit.

4) Increase spending habits.

5) Give loan against the credit card, as they are customers with good repayment record.

6) Tie up with luxury brands, which will drive more one_time_maximum spending.

Group 2 : Low Spending Group

1) customers should be given remainders for payments. Offers can be provided on early payments to improve their payment rate.

2) Increase there spending habits by tying up with grocery stores, utilities (electricity, phone, gas, others).

Group 3 : Medium Spending Group

1) *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.*

2) *Promote premium cards/loyalty cars to increase transactions.*

3) *Increase spending habits by trying with premium e-commerce sites, travel portal, travel airlines/hotel, as this will encourage them to spend more.*

End of 1st Business Report, 2nd Business Report starts from next page.

Problem 2: CART-RF-ANN

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

2.1 Read the data and do exploratory data analysis (4 pts). Describe the data briefly. Interpret the inferences for each (2 pts). Initial steps like head() .info(), Data Types, etc . Null value check. Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Summary stats, Skewness, Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.

Answer:

After reading the data,

	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

```
df2.tail()
```

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
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

- 10 variables
- Age, Commission, Duration, Sales are numeric variable.
- rest are categorical variables.
- 3000 records, no missing records.
- 9 independent variable and one target variable.

Description of Data

	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

We have 4 numeric and 6 categorical values. The most preferred type seems to be travel agency and top channel is online. Customized plan is most used by customers. The most sought destination is ASIA it seems. Duration has negative value, it is not possible so there’s a wrong entry.
Commission & Sales- mean and median varies significantly.
We check for duplicates and found that there are 139 duplicate rows. As there are no unique identifiers I won’t drop the duplicates, as it may be different customer’s data.

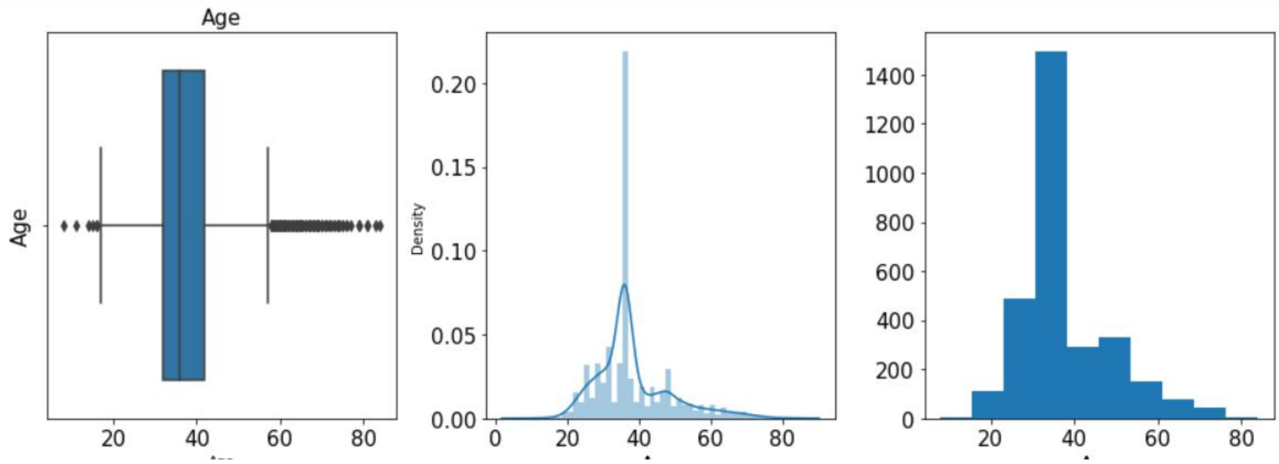
Exploratory Data Analysis

A. Univariate Analysis

NUMERIC VALUES

1) <u>Age Variable</u>

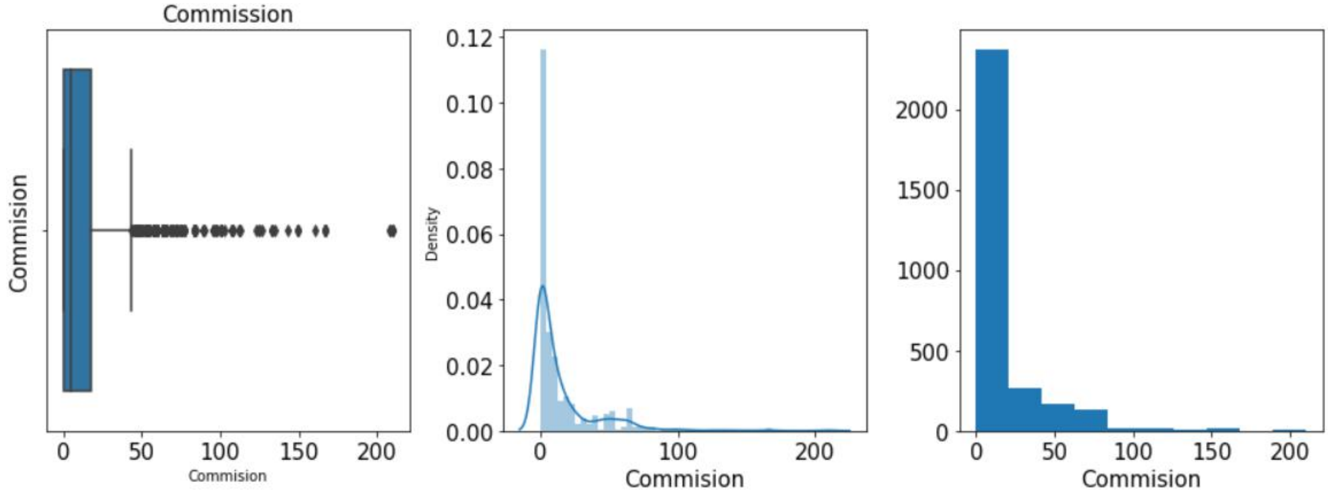
Range of values: 76
Minimum Age: 8
Maximum Age: 84
Mean value: 38.091
Median value: 36.0
Standard deviation: 10.463518245377944
Null values: False
spending - 1st Quartile (Q1) is: 32.0
spending - 3st Quartile (Q3) is: 42.0
Inter-quartile range (IQR) of Age is 10.0
Lower outliers in Age: 17.0
Upper outliers in Age: 57.0
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 %



2) <u>Commission Variable</u>

Range of values: 210.21
Minimum Commission: 0.0
Maximum Commission: 210.21
Mean value: 14.529203333333266
Median value: 4.63
Standard deviation: 25.48145450662553

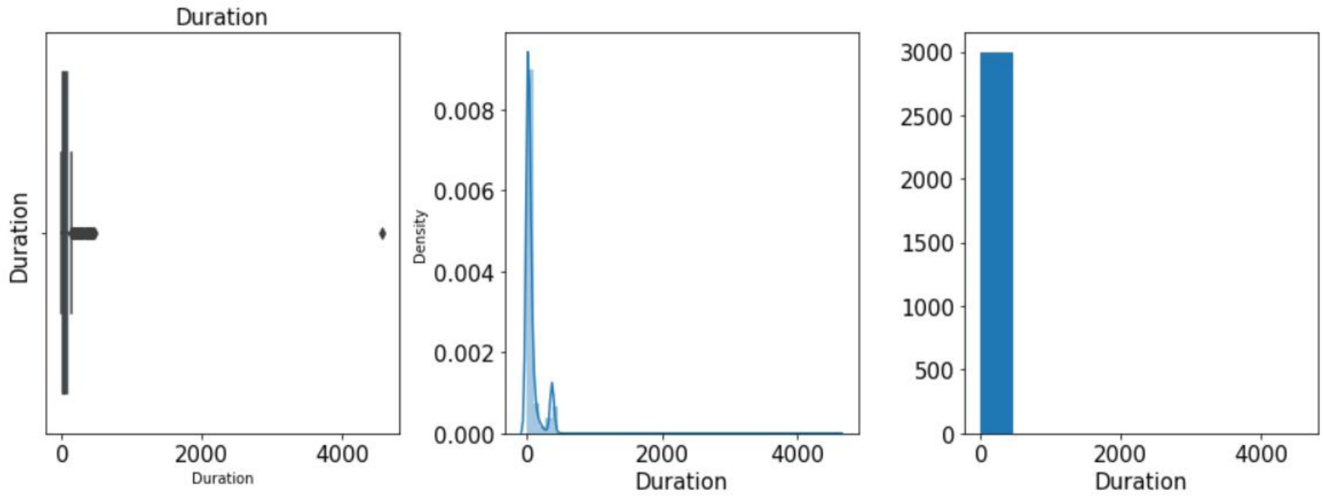
Null values: False
Commission - 1st Quartile (Q1) is: 0.0
Commission - 3st Quartile (Q3) is: 17.235
Inter-quartile range (IQR) of Commission is   17.235
Lower outliers in Commission: -25.8525
Upper outliers in Commission: 43.0875
Number of outliers in Commission upper :   362
Number of outliers in Commission lower :   0
% of Outlier in Commission upper: 12 %
% of Outlier in Commission lower: 0 %



3) Duration Variable

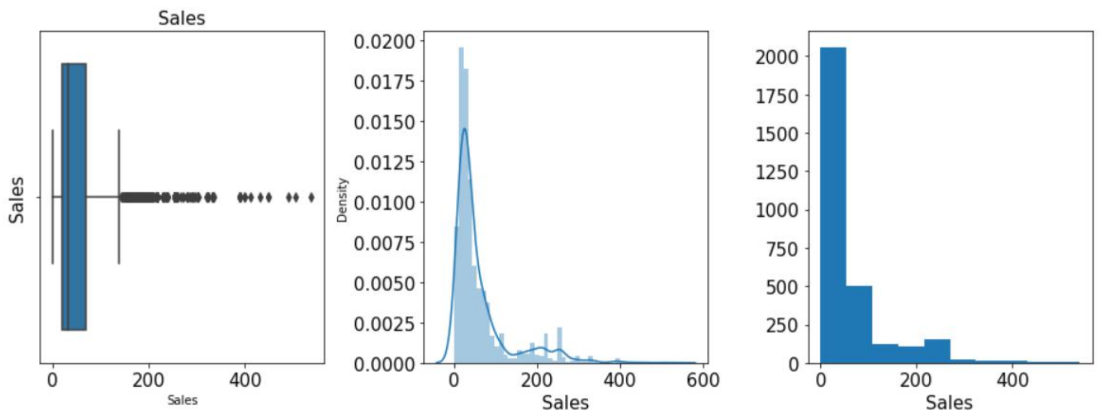
Range of values: 4581
Minimum Duration: -1
Maximum Duration: 4580
Mean value: 70.00133333333333
Median value: 26.5
Standard deviation: 134.05331313253495
Null values: False
Duration - 1st Quartile (Q1) is: 11.0
Duration - 3st Quartile (Q3) is: 63.0
Interquartile range (IQR) of Duration is   52.0
Lower outliers in Duration: -67.0
Upper outliers in Duration: 141.0
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 %



4) Sales Variable

Range of values: 539.0
Minimum Sales: 0.0
Maximum Sales: 539.0
Mean value: 60.24991333333344
Median value: 33.0
Standard deviation: 70.73395353143047
Null values: False
Sales - 1st Quartile (Q1) is: 20.0
Sales - 3st 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
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 %

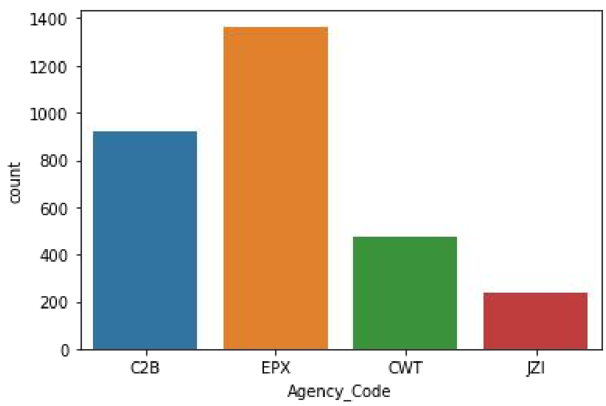


There are outliers in all the variables, but the sales and commision can be a genuine business value. Random Forest and CART can handle the outliers. Hence, Outliers are not treated for now, we will keep the data as it is.

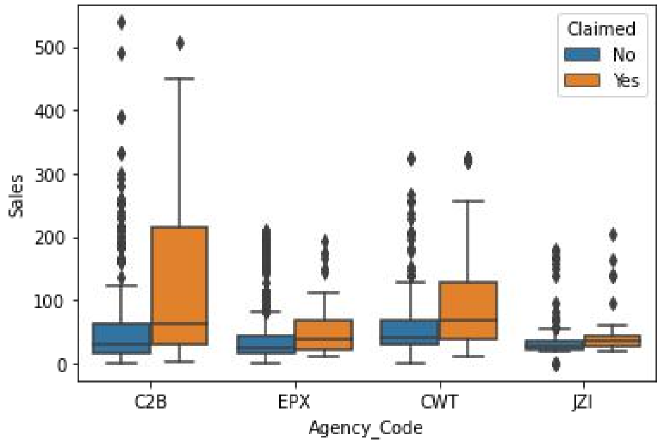
I will treat the outliers for the ANN model to compare the same after the all the steps just for comparison.

CATEGORICAL VARIABES

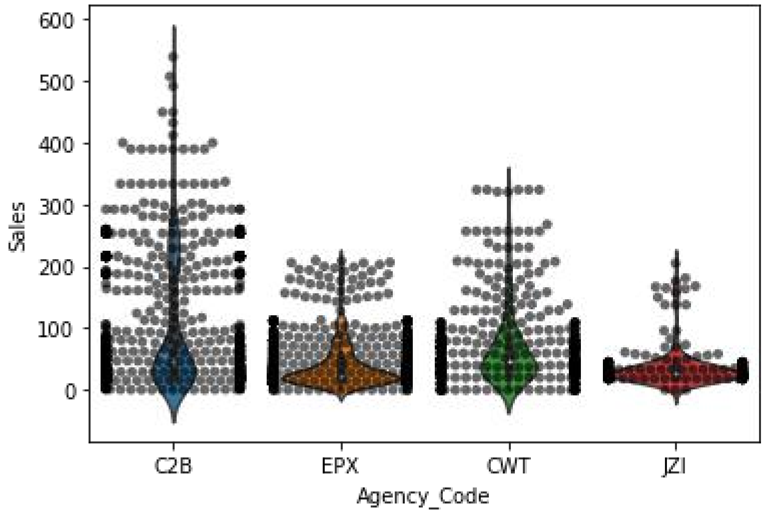
1) Agency_Code



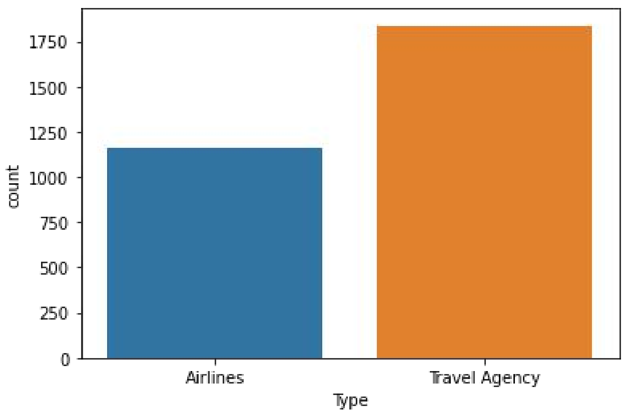
EPX has the highest frequency.



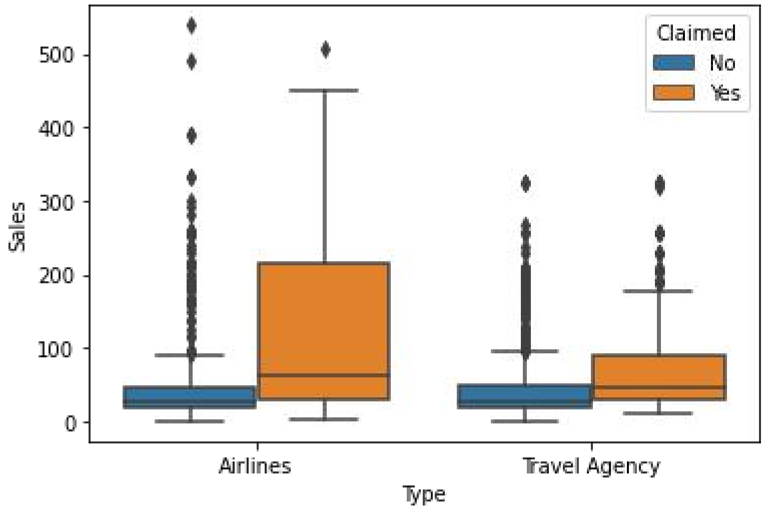
C2B has more claims than any other agency.



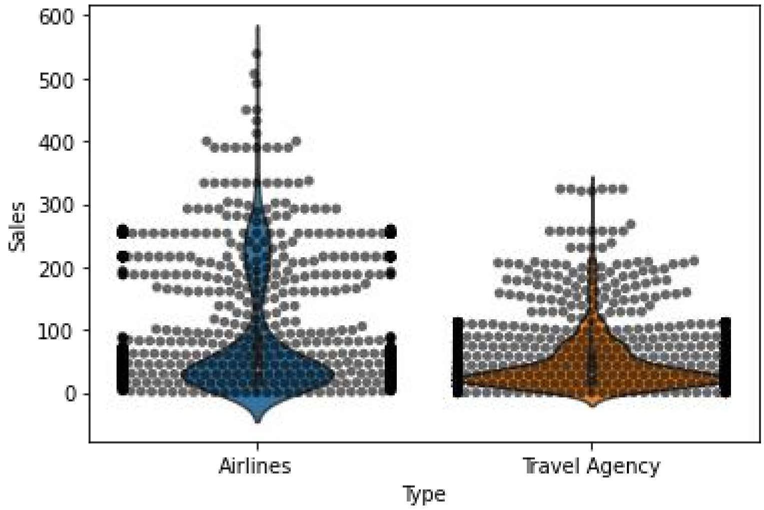
2) Type



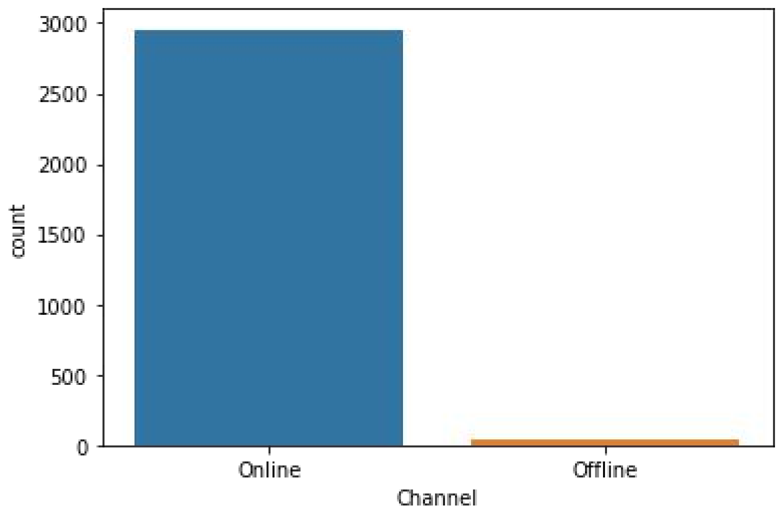
It's clearly visible that Travel Agency's frequency is higher.



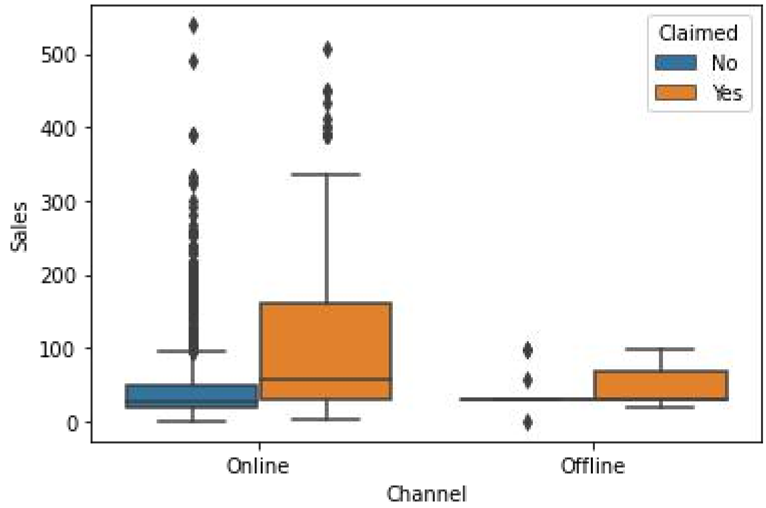
The boxplot shows Airlines have higher claim than Travel Agency.



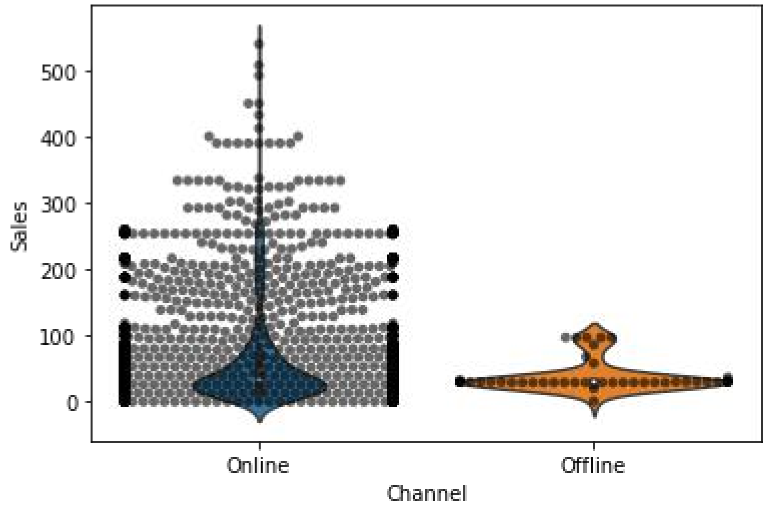
3) Channel



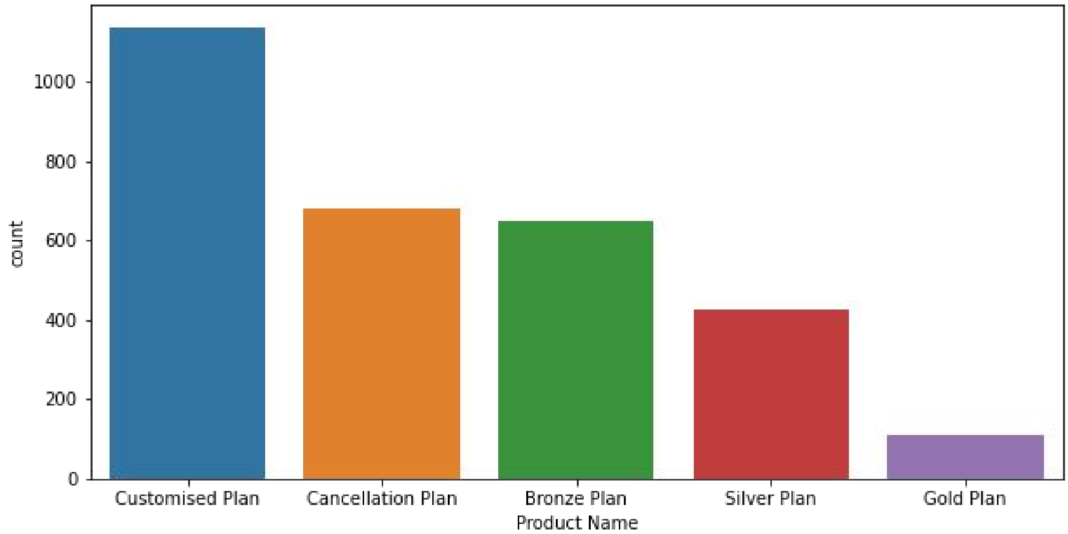
Online channels have higher user frequency.



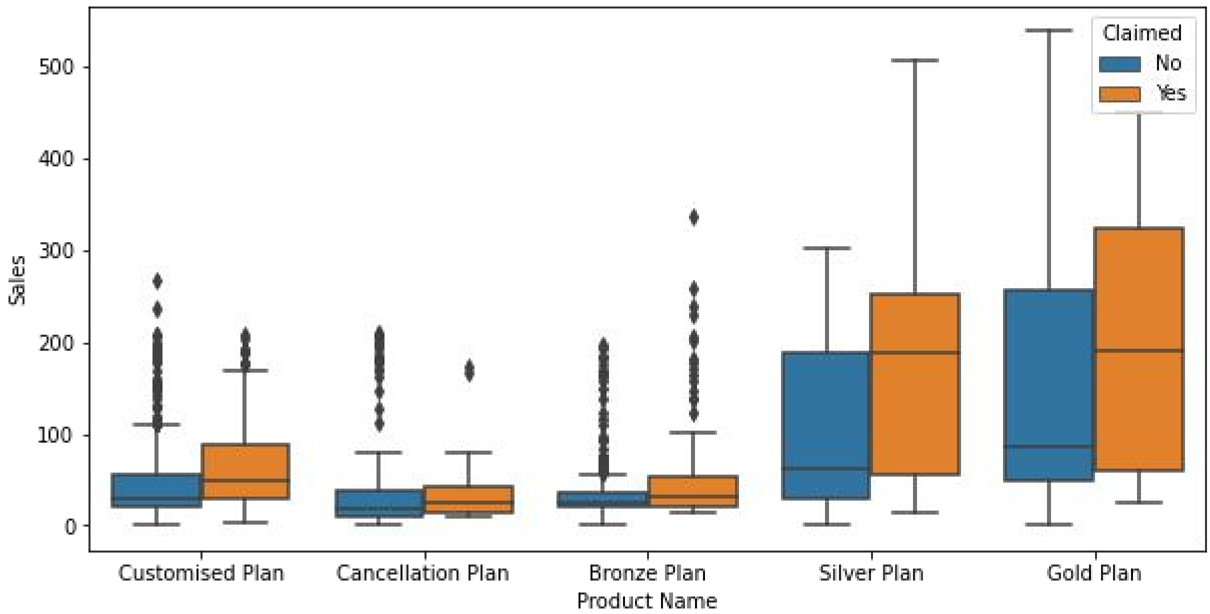
Online users have higher claims as well.



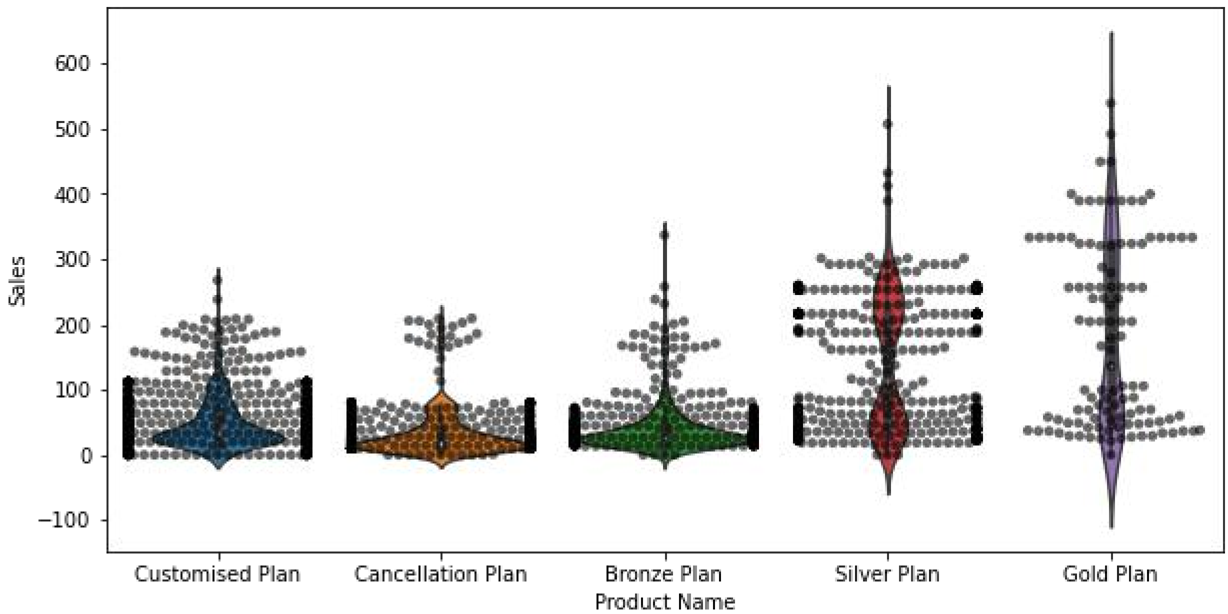
4) <u>Product Name</u>



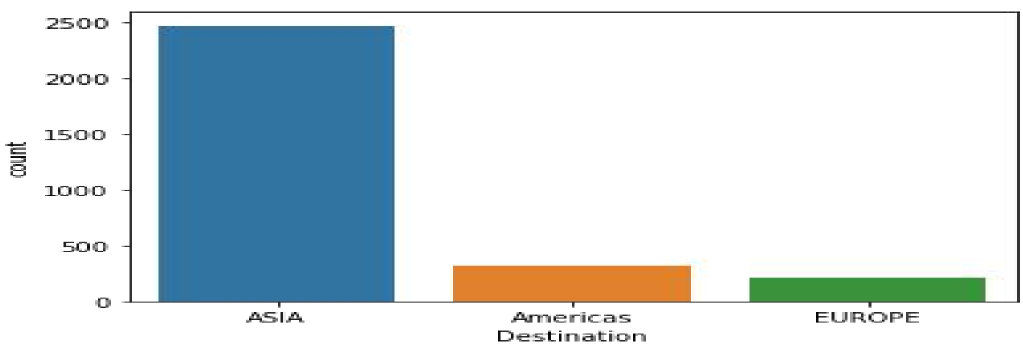
Customized Plans have higher frequency.



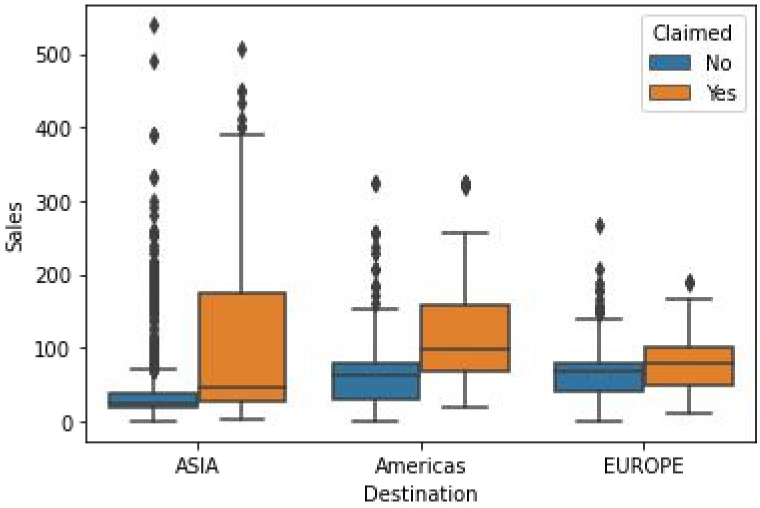
Users who bought Gold Plan have higher claim frequency.



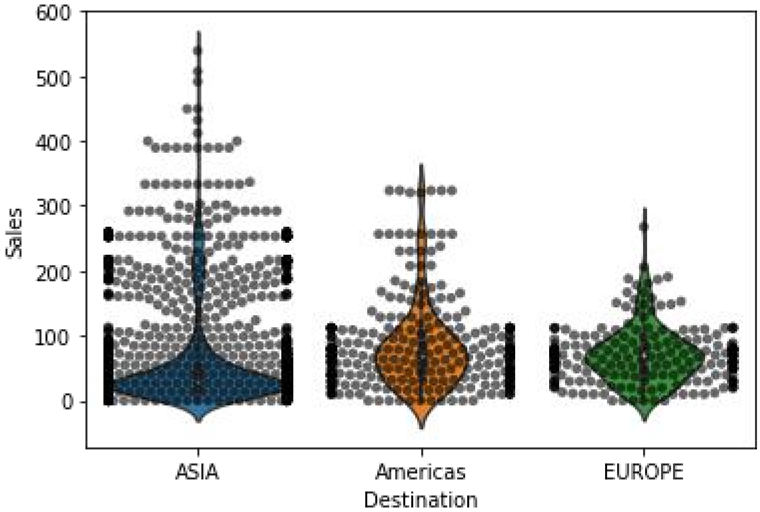
5) <u>Destination</u>



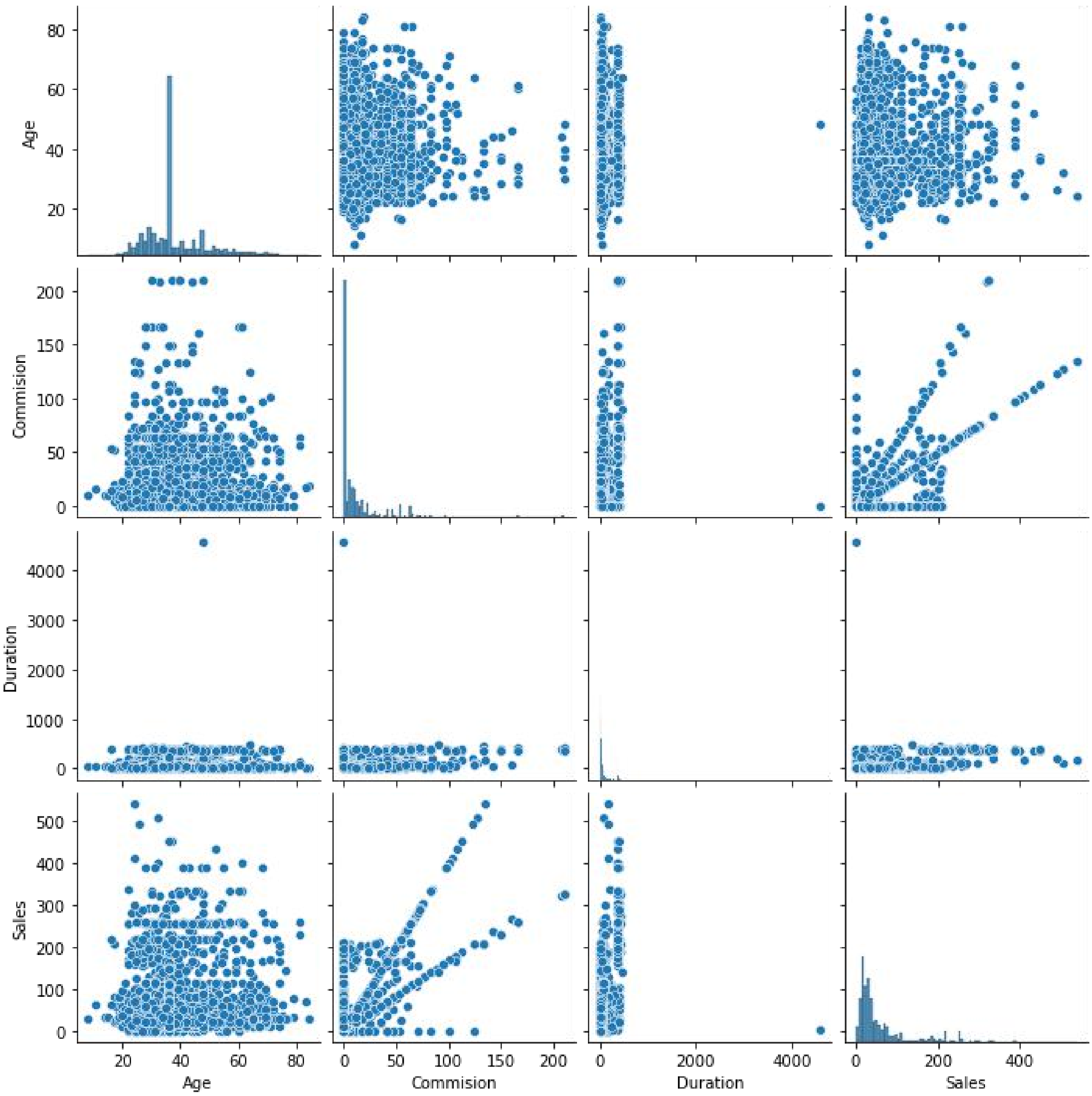
ASIA is the most preferred destination by most people.

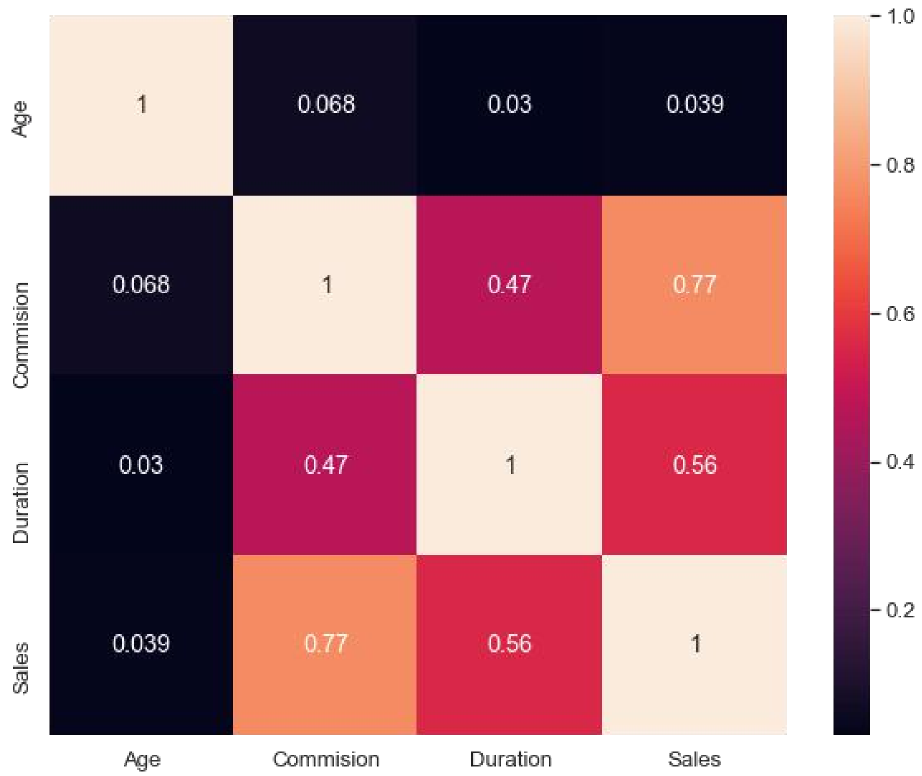


People visiting ASIA has higher claim.



B.MULTIVARIATE ANALYSIS





Not much multicollinearity is observed.
Only positive correlation.

2.2 Data Split: Split the data into test and train(1 pts), build classification model CART (1.5 pts), Random Forest (1.5 pts), Artificial Neural Network(1.5 pts). Object data should be converted into categorical/numerical data to fit in the models. (pd.categorical().codes(), pd.get_dummies(drop_first=True)) Data split, ratio defined for the split, train-test split should be discussed. Any reasonable split is acceptable. Use of random state is mandatory. Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Apply grid search for each model and make models on best_params_. Feature importance for each model.

<u>Answer:</u>

Converting all categorical data into numeric values:

```
<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
```

	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

Then we check the proportion of 1's and 0's of our target column:

```
#Frequency of 1's and 0's
df2.Claimed.value_counts(normalize=True)
```

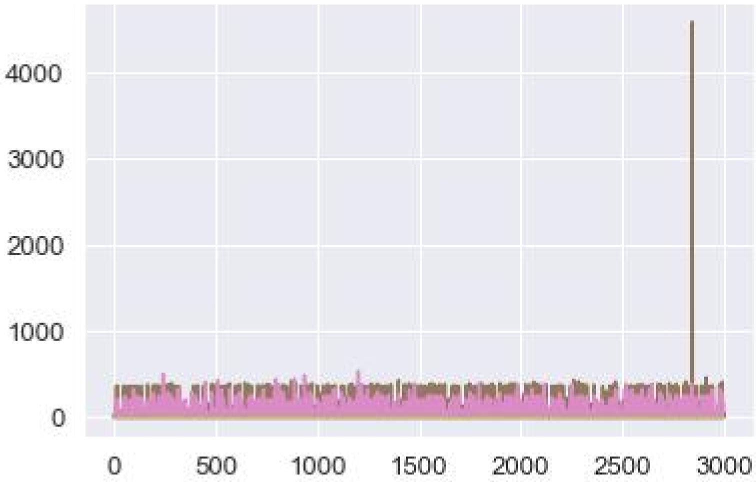
```
0    0.692
1    0.308
Name: Claimed, dtype: float64
```

For training and testing purpose we extract the target column:

```
x = df2.drop("Claimed", axis=1)

y = df2.pop("Claimed")

x.head()
```

	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

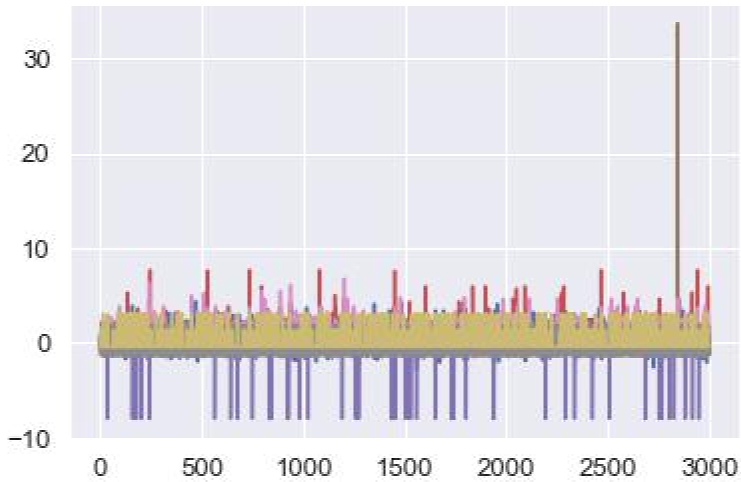
We the check data prior and post- scaling:



Prior to scaling the attributes

After scaling the data:

	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



Scaled Data

Now spliting dataset in train data and test data in 70:30 ratio:

```
x_train, x_test, train_labels, test_labels = train_test_split(x_scaled, y, test_size=.30, random_state=5)
```

Then checking the dimensions of the train and test data:

```
x_train (2100, 9)
x_test (900, 9)
train_labels (2100,)
test_labels (900,)
```

We have split the dataset into train and test data and have taken out the target column from the train and test data into separate variable for evaluation.

Model 1: CART

I created a CART model and used the Grid Search (best_params_and best_estimator_) method to find the optimal values for the parameters for the model.
It helped me in generating a regularized tree with tuned parameters.

```
param_grid_dtcl = {
    'criterion': ['gini'],
    'max_depth': [3, 5, 7, 10,12],
    'min_samples_leaf': [20,30,40,50,60],
    'min_samples_split': [150,300,450],
}

dtcl = DecisionTreeClassifier(random_state=1)

grid_search_dtcl = GridSearchCV(estimator = dtcl, param_grid = param_grid_dtcl, cv = 10)
```

```
grid_search_dtcl.fit(x_train, train_labels)
print(grid_search_dtcl.best_params_)
best_grid_dtcl = grid_search_dtcl.best_estimator_
best_grid_dtcl
```

```
{'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)
```

Regularized the tree using best_estimator_

```
# Generating Tree

train_char_label = ['no', 'yes']
tree_regularized = open('tree_regularized.dot','w')
dot_data = tree.export_graphviz(best_grid_dtcl, out_file= tree_regularized ,
                                feature_names = list(x_train),
                                class_names = list(train_char_label))

tree_regularized.close()
dot_data
```

- Variable Importance

	Imp
Agency_Code	0.596448
Sales	0.207778
Product Name	0.108327
Duration	0.034766
Commision	0.033559
Age	0.019123
Type	0.000000
Channel	0.000000
Destination	0.000000

-Predicting on train and test data

```
ytrain_predict_dtcl = best_grid_dtcl.predict(x_train)
ytest_predict_dtcl = best_grid_dtcl.predict(x_test)
```

```
ytest_predict_dtcl
ytest_predict_prob_dtcl=best_grid_dtcl.predict_proba(x_test)
ytest_predict_prob_dtcl
pd.DataFrame(ytest_predict_prob_dtcl).head()
```

	0	1
0	0.720635	0.279365
1	0.979452	0.020548
2	0.914842	0.085158
3	0.552941	0.447059
4	0.914842	0.085158

Model 2: Random Forest

Used the Grid Search(best_params_and best_estimator_) method to find the optimal values for the parameters for the model.

```
param_grid_rfcl = {
    'max_depth': [4,5,6],#20,30,40
    'max_features': [2,3,4,5],## 7,8,9
    'min_samples_leaf': [8,9,11,15],## 50,100
    'min_samples_split': [46,50,55], ## 60,70
    'n_estimators': [290,350,400] ## 100,200
}

rfcl = RandomForestClassifier(random_state=1)

grid_search_rfcl = GridSearchCV(estimator = rfcl, param_grid = param_grid_rfcl, cv = 5)
```

```
grid_search_rfcl.fit(x_train, train_labels)
print(grid_search_rfcl.best_params_)
best_grid_rfcl = grid_search_rfcl.best_estimator_
best_grid_rfcl
```

```
{'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)
```

- Predicting on train and test data

```
ytrain_predict_rfcl = best_grid_rfcl.predict(x_train)
ytest_predict_rfcl = best_grid_rfcl.predict(x_test)
```

```
ytest_predict_rfcl
ytest_predict_prob_rfcl=best_grid_rfcl.predict_proba(x_test)
ytest_predict_prob_rfcl
pd.DataFrame(ytest_predict_prob_rfcl).head()
```

	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

	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

Model 3: ANN

Used Grid search(best_params_and best_estimator_) method to find the optimal values for the parameters for the model.

```
param_grid_nncl = {
    'hidden_layer_sizes': [50,100,200], # 50, 200
    'max_iter': [2500,3000,4000], #5000,2500
    'solver': ['adam'], #sgd
    'tol': [0.01],
}

nncl = MLPClassifier(random_state=1)

grid_search_nncl = GridSearchCV(estimator = nncl, param_grid = param_grid_nncl, cv = 10)
```

```
grid_search_nncl.fit(x_train, train_labels)
grid_search_nncl.best_params_
best_grid_nncl = grid_search_nncl.best_estimator_
best_grid_nncl
```

```
MLPClassifier(hidden_layer_sizes=200, max_iter=2500, random_state=1, tol=0.01)
```

-Predicting train and test data

```
ytrain_predict_nncl = best_grid_nncl.predict(x_train)
ytest_predict_nncl = best_grid_nncl.predict(x_test)
```

```
ytest_predict_nncl
ytest_predict_prob_nncl=best_grid_nncl.predict_proba(x_test)
ytest_predict_prob_nncl
pd.DataFrame(ytest_predict_prob_nncl).head()
```

	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: Check the performance of Predictions on Train and Test sets using Accuracy (1 pts), Confusion Matrix (2 pts), Plot ROC curve and get ROC_AUC score for each model (2 pts), Make classification reports for each model. Write inferences on each model (2 pts). Calculate Train and Test Accuracies for each model. Comment on the validness of models (overfitting or underfitting) Build confusion matrix for each model. Comment on the positive class in hand. Must clearly show obs/pred in row/col Plot roc_curve for each model. Calculate roc_auc_score for each model. Comment on the above calculated scores and plots. Build classification reports for each model. Comment on f1 score, precision and recall, which one is important here.

<u>Answer:</u>

In following confusion matrices,
(0,0) - True Negative
(0,1) - False Positive
(1,0) - False Negative
(1,1) - True positive

F1-Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

Recall is the ratio of correctly predicted positive observations to the all observations.

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.High precision relates to the low false positive rate.

So, we are getting high claim frequency according to the Insurance Firm.
So, I would take Recall as important metric to consider first. Then I would go with Precision and finally check F1-score.

TRAINING DATA

- Confusion matrix and Classification Report

```
confusion_matrix(train_labels, ytrain_predict_dtcl)
```

```
array([[1281,  172],
       [ 257,  390]], dtype=int64)
```

```
print(classification_report(train_labels, ytrain_predict_dtcl))
```

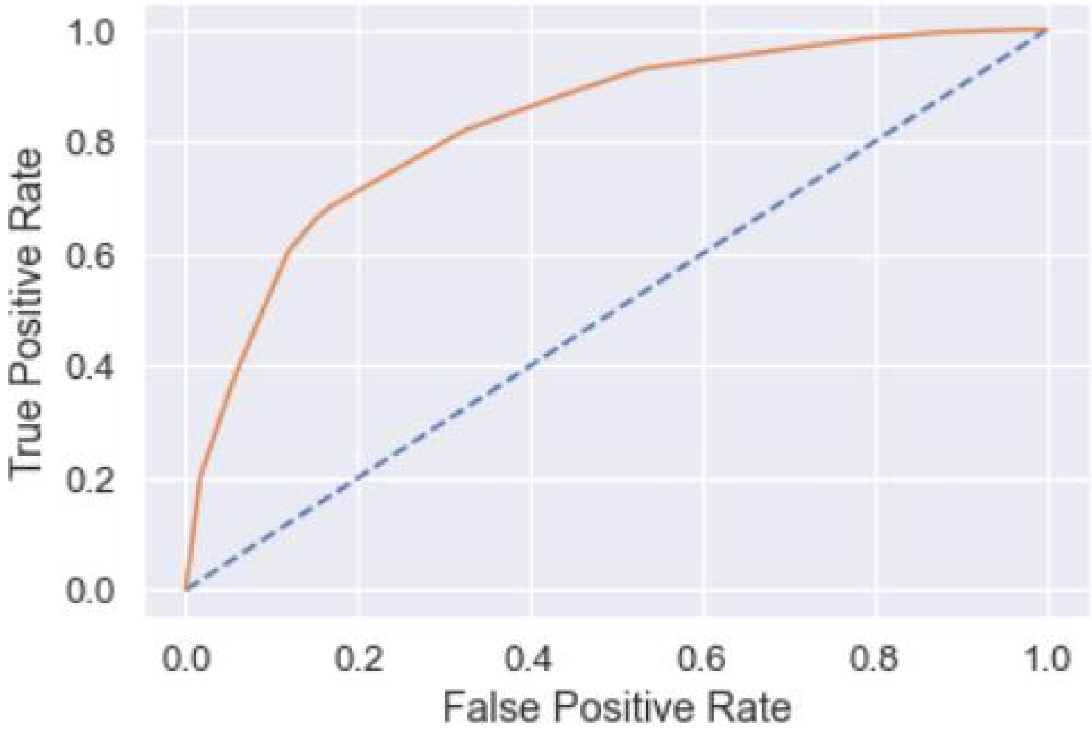
```
              precision    recall  f1-score   support

           0       0.83      0.88      0.86      1453
           1       0.69      0.60      0.65       647

    accuracy                           0.80      2100
   macro avg       0.76      0.74      0.75      2100
weighted avg       0.79      0.80      0.79      2100
```

Train Data:
- AUC: 83%
- Accuracy: 79%
- Precision: 69%
- f1-Score: 65%
- Recall: 60%

- ROC curve:

```
AUC: 0.832
```

```
[<matplotlib.lines.Line2D at 0x22790e0fee0>]
```



<u>TEST DATA</u>

- Confusion matrix and Classification Report

```
confusion_matrix(test_labels, ytest_predict_dtcl)
```

```
array([[540,  83],
       [115, 162]], dtype=int64)
```

```
print(classification_report(test_labels, ytest_predict_dtcl))
```

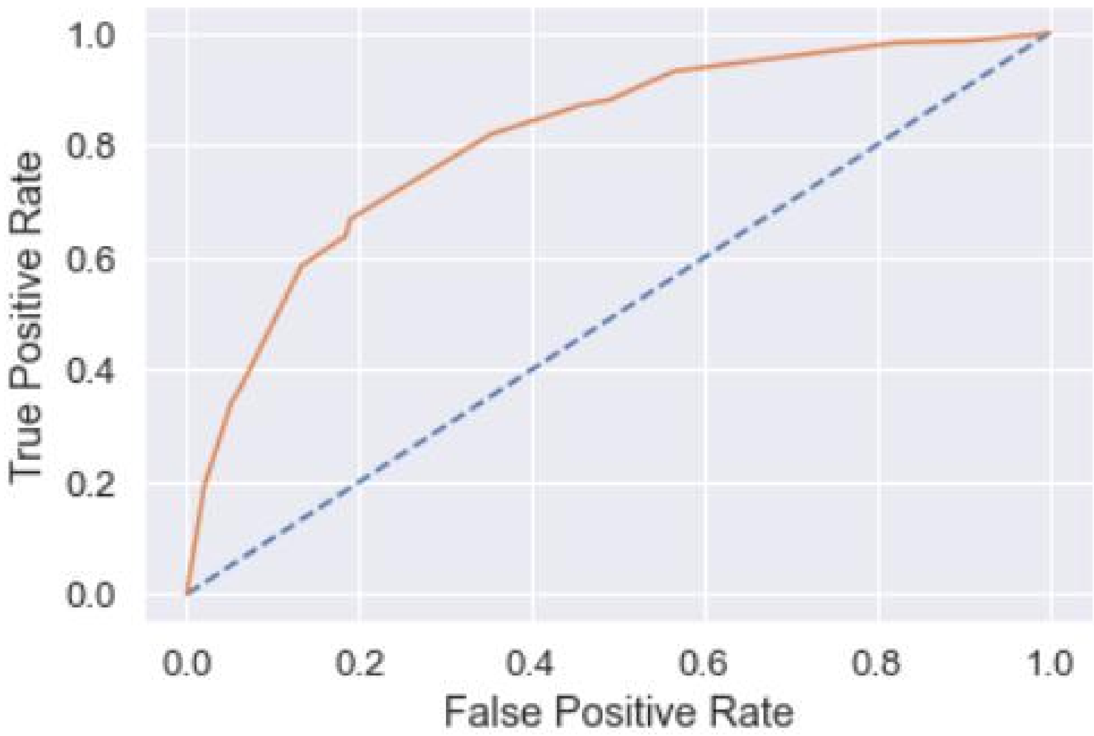
```
              precision    recall  f1-score   support

           0       0.82      0.87      0.85       623
           1       0.66      0.58      0.62       277

    accuracy                           0.78       900
   macro avg       0.74      0.73      0.73       900
weighted avg       0.77      0.78      0.78       900
```

Test Data:
- AUC: 81%
- Accuracy: 78%
- Precision: 66%
- f1-Score: 62%
- Recall: 58%

- ROC Curve

```
AUC: 0.811
```

```
[<matplotlib.lines.Line2D at 0x22790dc3520>]
```



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

MODEL 2: RANDOM FOREST

TRAINING DATA

- Confusion Matrix and Classification Report

```
confusion_matrix(train_labels,ytrain_predict_rfcl)
```

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

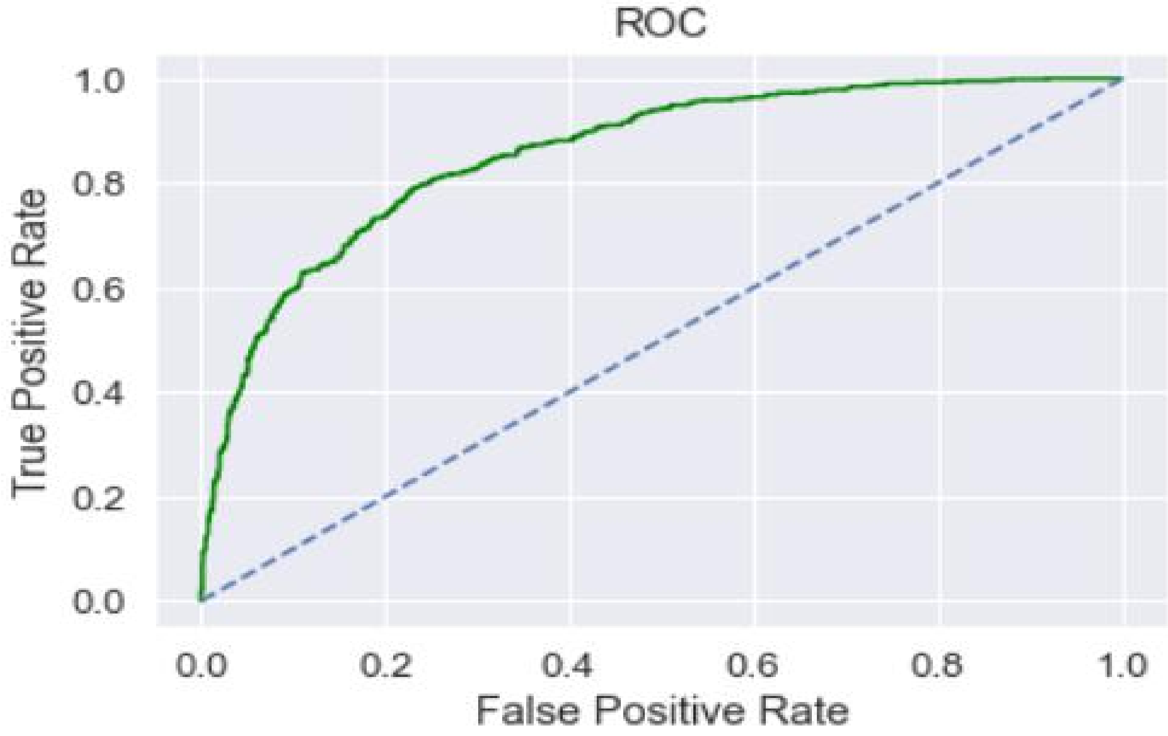
```
print(classification_report(train_labels,ytrain_predict_rfcl))
```

```
              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
```

Train Data:
- AUC: 86%
- Accuracy: 80%
- Precision: 72%
- f1-Score: 66%
- Recall: 61%

- ROC Curve

```
Area under Curve is 0.8563713512840778
```



<u>TEST DATA</u>

- Confusion Matrix and Classification Report

```
confusion_matrix(test_labels,ytest_predict_rfcl)
```

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

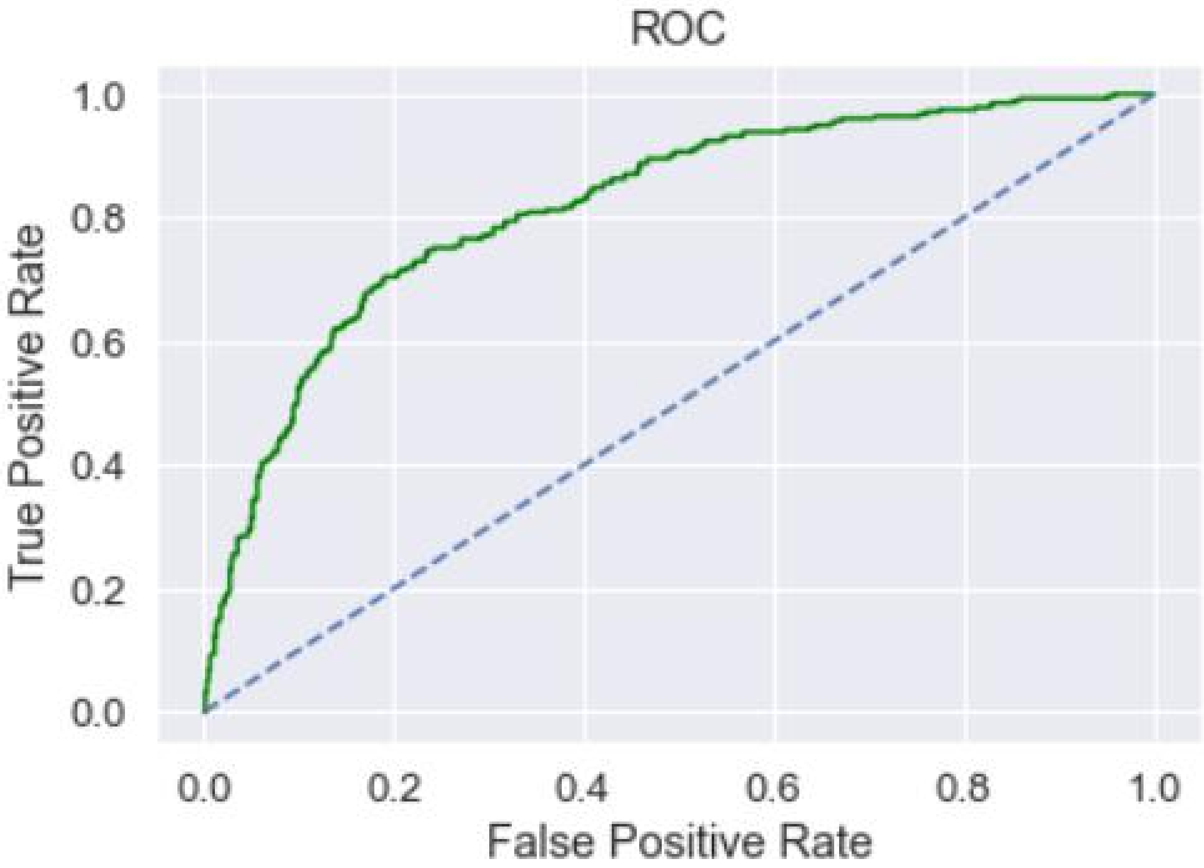
```
print(classification_report(test_labels,ytest_predict_rfcl))
```

	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

Test Data:
- AUC: 82%
- Accuracy: 78%
- Precision: 68%
- f1-Score: 62%
- Recall: 56%

- ROC Curve

```
Area under Curve is 0.8181994657271499
```



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

MODEL 3: Neural Network

TRAINING DATA

- Confusion Matrix and Classification Report

```
confusion_matrix(train_labels,ytrain_predict_nncl)
```

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

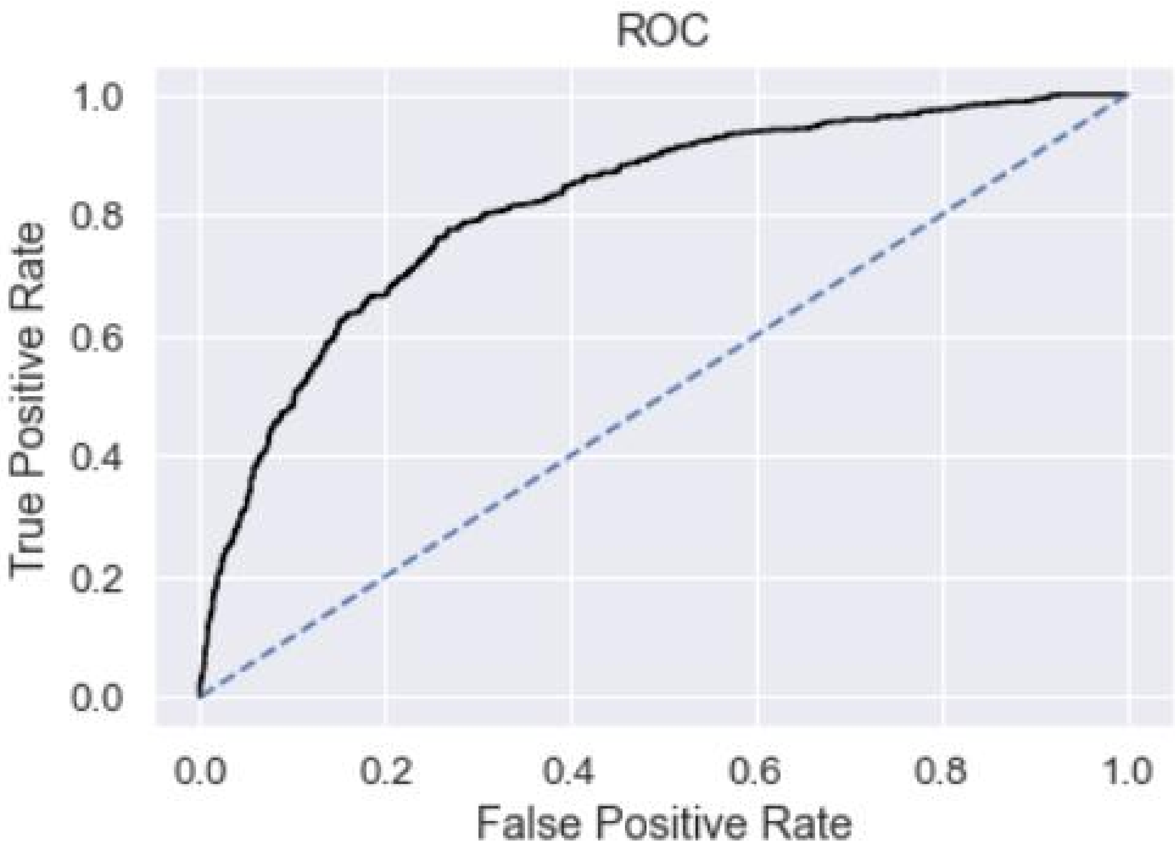
```
print(classification_report(train_labels,ytrain_predict_nncl))
```

```
              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
```

Train Data:
- AUC: 82%
- Accuracy: 78%
- Precision: 68%
- f1-Score: 59%
- Recall: 51%

-ROC Curve

```
Area under Curve is 0.8166831721609928
```



<u>TEST DATA</u>

- Confusion Matrix and Classification Report

```
confusion_matrix(test_labels,ytest_predict_nncl)
```

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

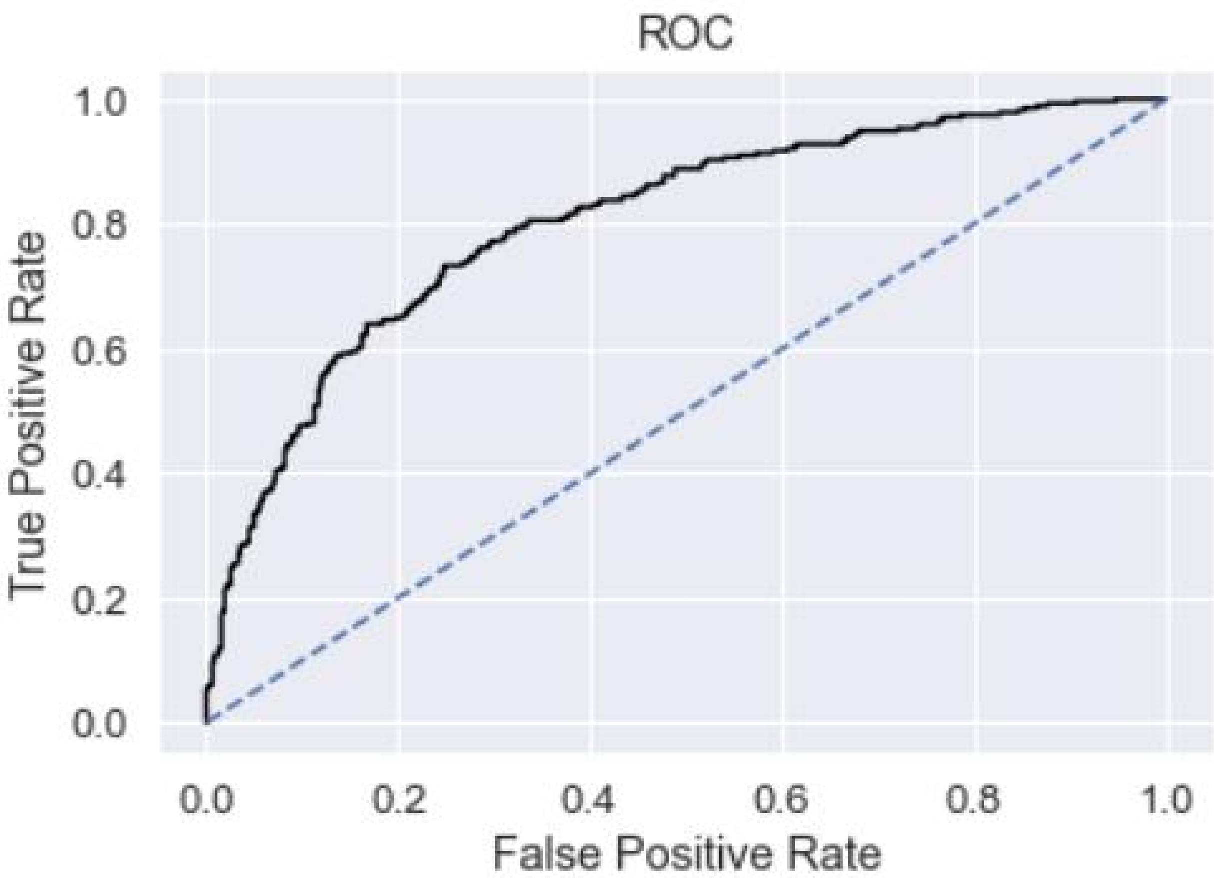
```
print(classification_report(test_labels,ytest_predict_nncl))
```

	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

Test Data:
- AUC: 80%
- Accuracy: 77%
- Precision: 67%
- f1-Score: 57%
- Recall: 50%

- ROC Curve

```
Area under Curve is 0.8044225275393896
```



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

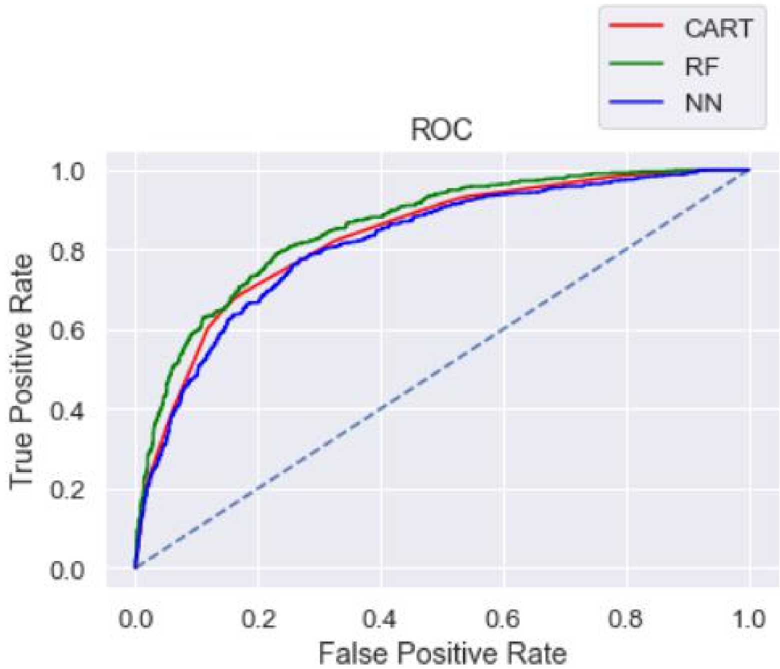
2.4 Final Model - Compare all models on the basis of the performance metrics in a structured tabular manner (2.5 pts). Describe on which model is best/optimized (1.5 pts). A table containing all the values of accuracies, precision, recall, auc_roc_score, f1 score. Comparison between the different models(final) on the basis of above table values. After comparison which model suits the best for the problem in hand on the basis of different measures. Comment on the final model.

<u>ANSWER:</u>

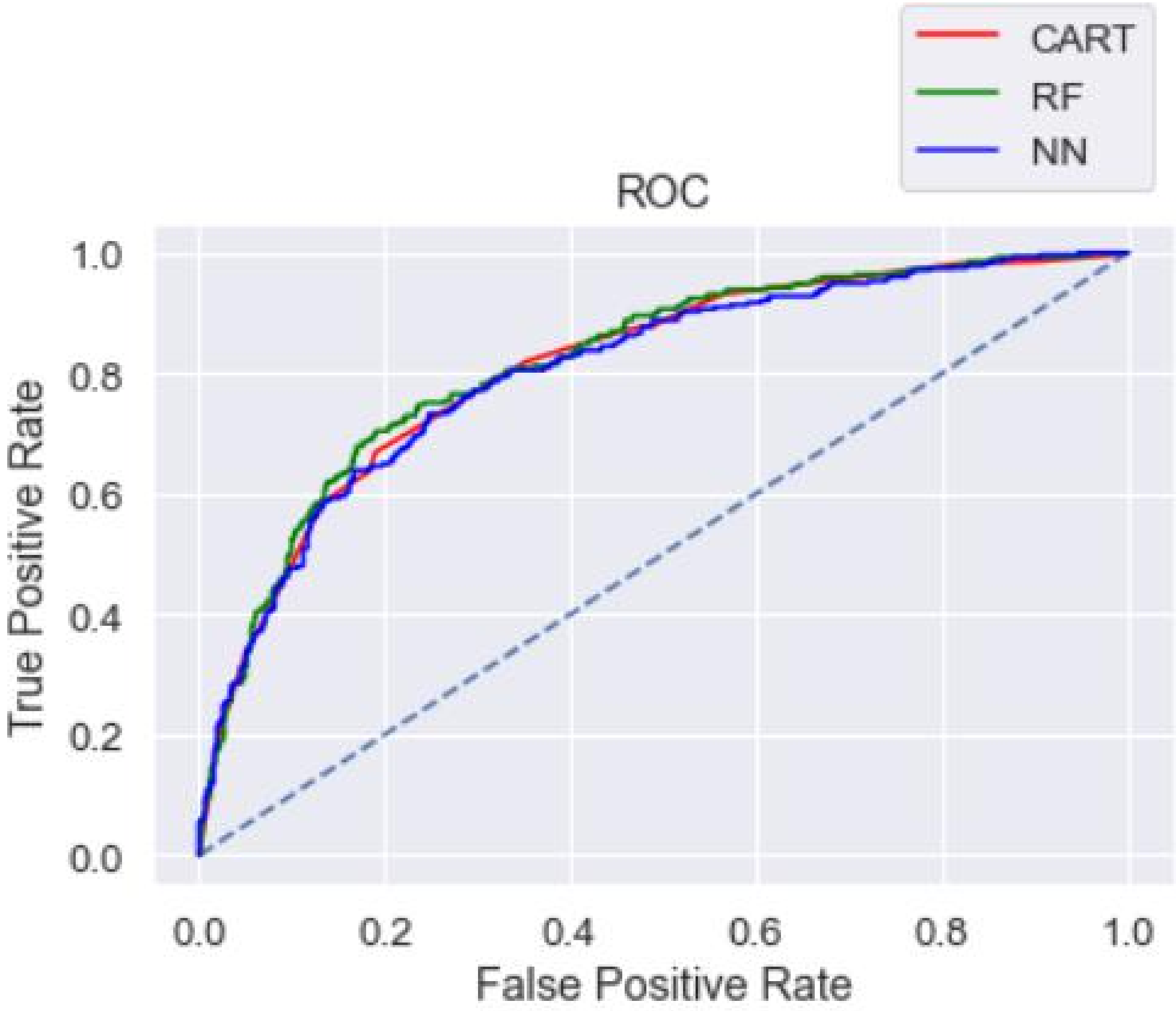
Comparing all performance metrics in a tabular form:

	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.80	0.78	0.80	0.78	0.78	0.77
AUC	0.83	0.81	0.86	0.82	0.82	0.80
Recall	0.60	0.58	0.61	0.56	0.51	0.50
Precision	0.69	0.66	0.72	0.68	0.68	0.67
F1 Score	0.65	0.62	0.66	0.62	0.59	0.57

ROC For training data of all models:



ROC for test data for all models:



CONCLUSION :
I am selecting the RF model, as it has better accuracy, precision, recall and f1-score better than other two CART & Neural Network.

2.5 Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective. There should be at least 3-4 Recommendations and insights in total. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks should only be allotted if the recommendations are correct and business specific.

ANSWER:

Looking at the model, more data will help us understand and predict models better.Streamlining online experiences benefited customers, leading to an increase in conversions,which subsequently raised profits. As per the data 90% of insurance is done by online channel.

1. Other interesting fact, is almost all the offline business has a claimed associated.

2.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.

3.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.

4. 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.

The Key performance indicator's of insurance claims are:

A. Increase customer satisfaction which in fact will give more revenue.
B. Combat fraud transactions, deploy measures to avoid fraudulent transactions at earliest.
C.Optimize claims recovery method.
D. Reduce claim handling costs.

End of Business Report