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.

Data Dictionary for Market Segmentation:

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

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

Ans 1.1 **Reading the data with basic initial steps:**

	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

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

Checking the information of the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   spending                      210 non-null    float64
 1   advance_payments              210 non-null    float64
 2   probability_of_full_payment   210 non-null    float64
 3   current_balance               210 non-null    float64
 4   credit_limit                  210 non-null    float64
 5   min_payment_amt               210 non-null    float64
 6   max_spent_in_single_shopping  210 non-null    float64
dtypes: float64(7)
memory usage: 11.6 KB
```

<u>Checking the null values:</u>

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

<u>Observations:</u>

- There are 7 variables and 210 records.
- No missing record based on initial analysis.
- All the variables numeric type.
- Data looks good based on initial records seen in top 5 and bottom 5.

<u>Univariate Analysis:</u>

<u>Reading the summary statistics of the dataset:</u>

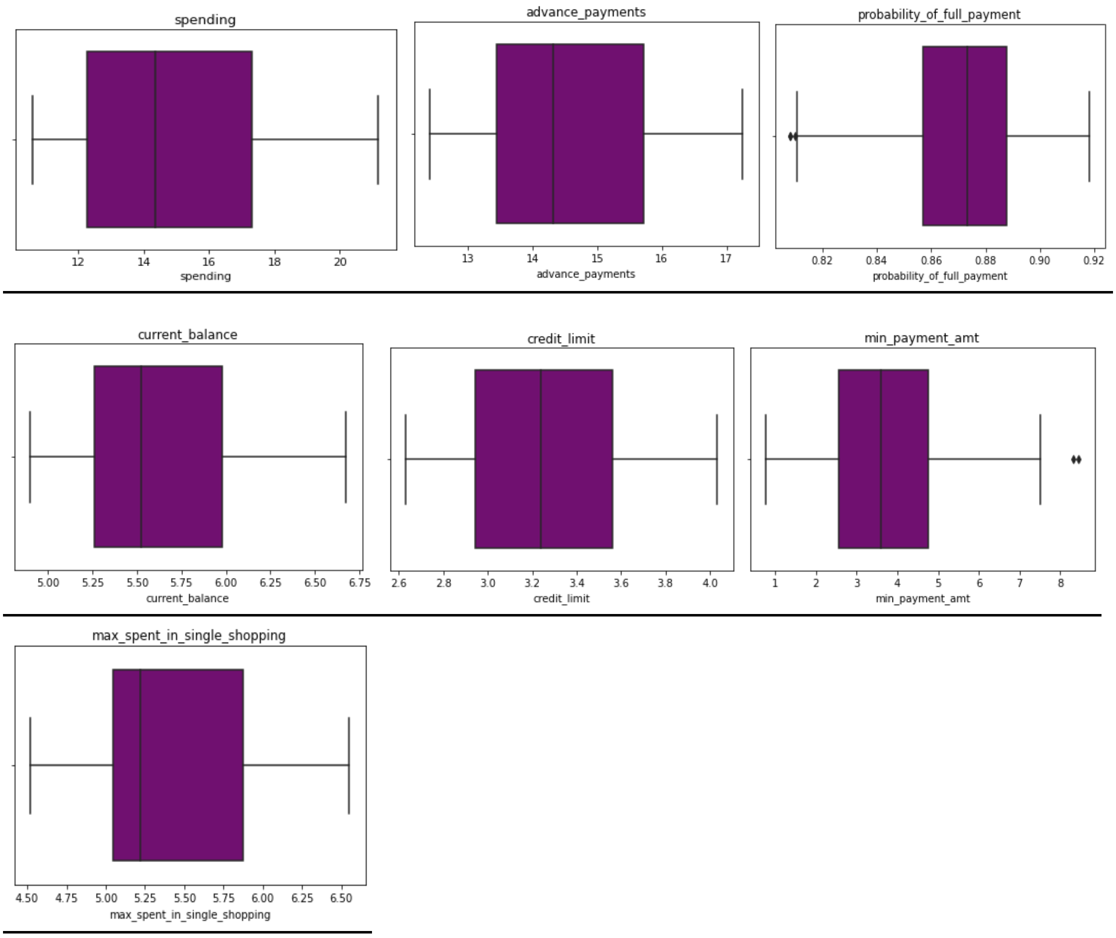
	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.847524	2.909699	10.5900	12.27000	14.35500	17.305000	21.1800
advance_payments	210.0	14.559286	1.305959	12.4100	13.45000	14.32000	15.715000	17.2500
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.85690	0.87345	0.887775	0.9183
current_balance	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

<u>Observations:</u>

Based on summary descriptive, the data looks good.

- We see for most of the variable, mean/medium are nearly equal
- Include a 90% to see variations and it looks distribute evenly
- Standard Deviation is high for spending variable.

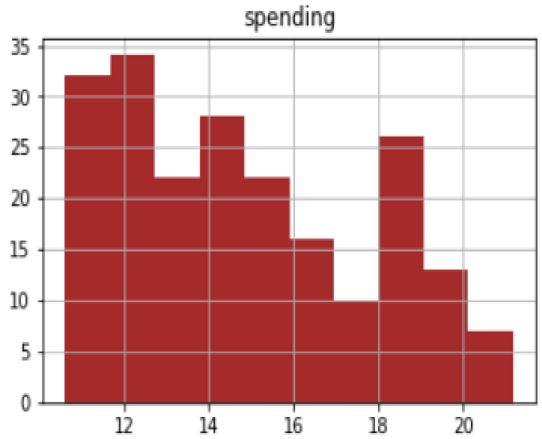
<u>Checking and plotting the box plot for outliers of all the features:</u>

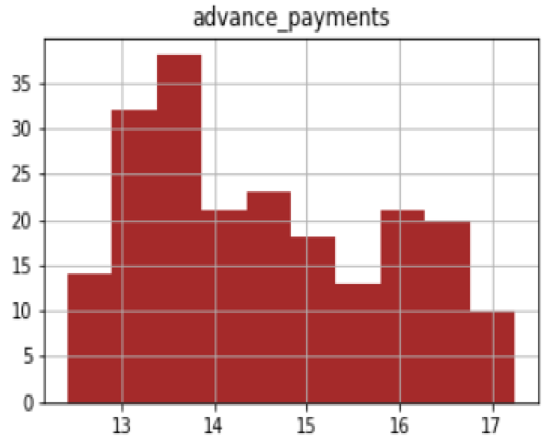


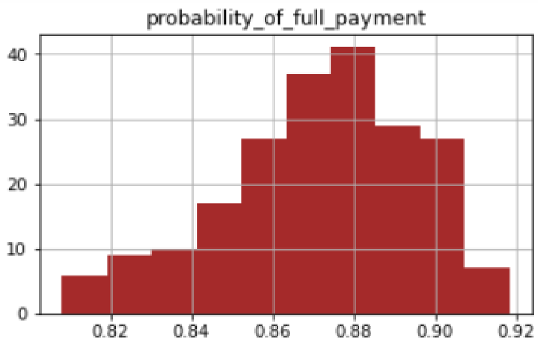
<u>Observations:</u>

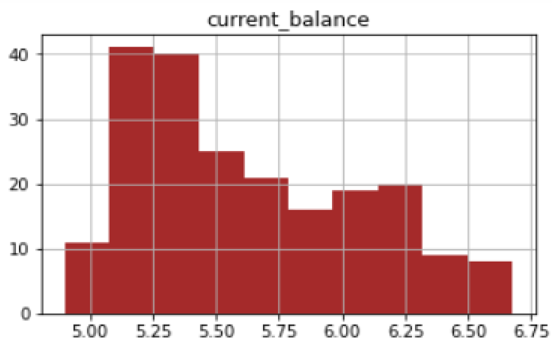
- Outliers found in 2 variables – Probability_of_full_amt & min_payment_amt

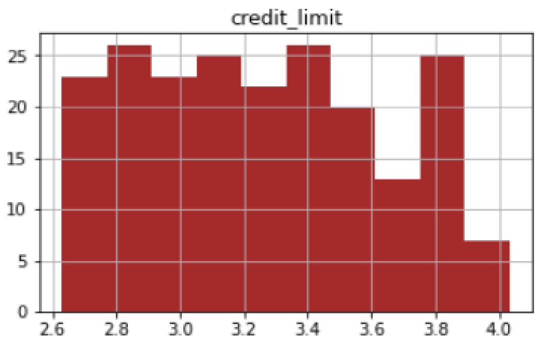
Plotting distribution using histogram of all individual variables:

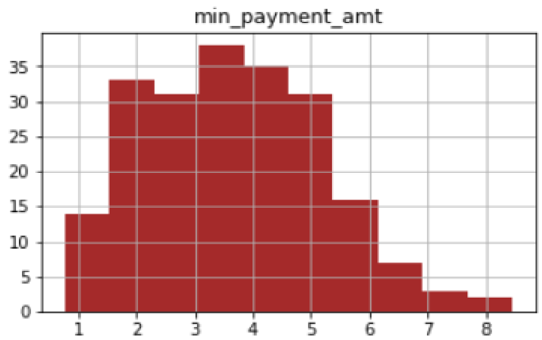


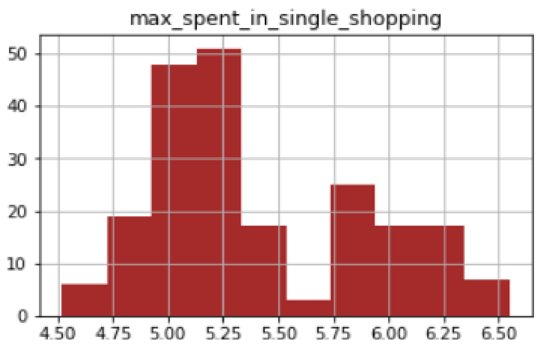










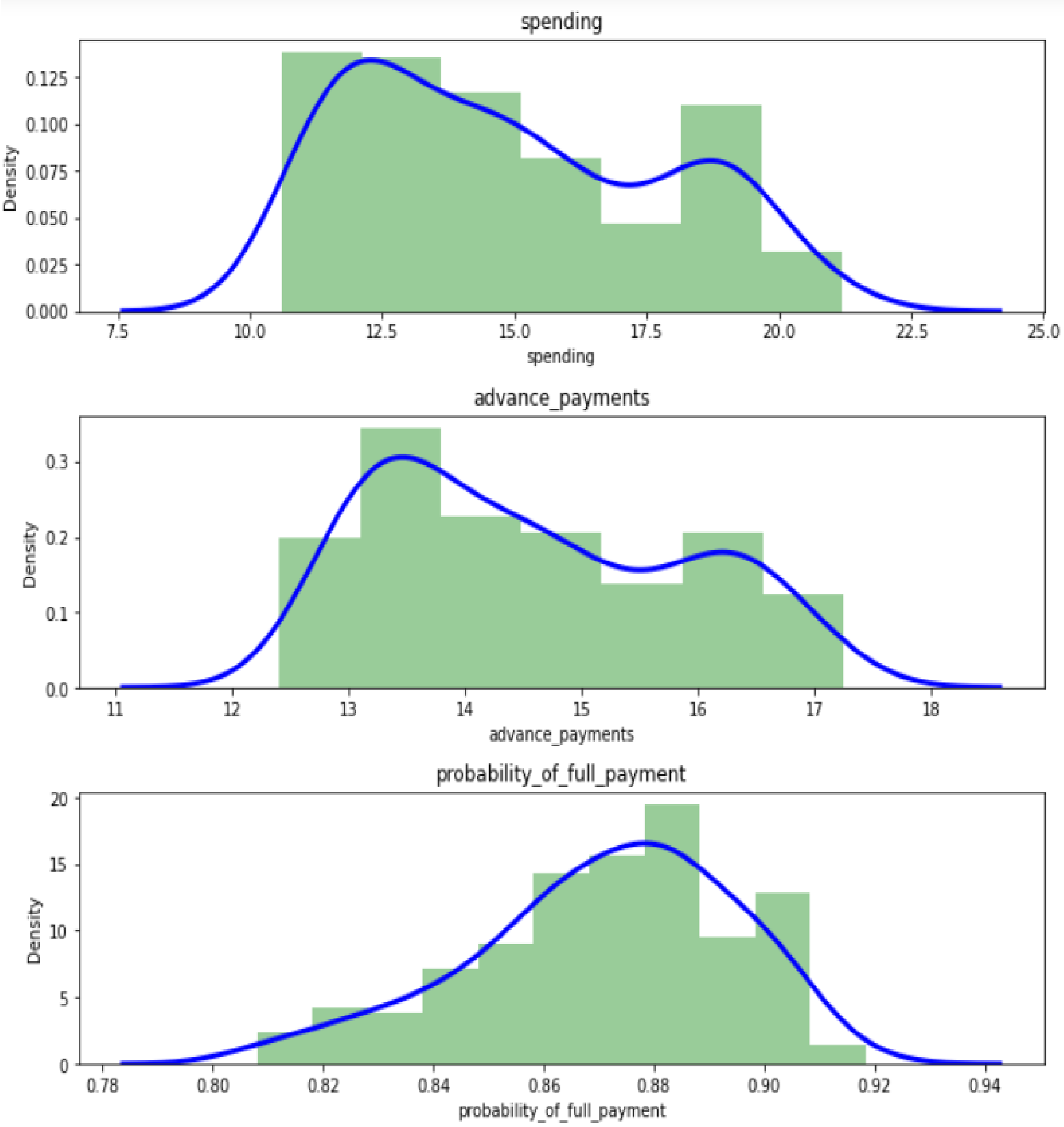


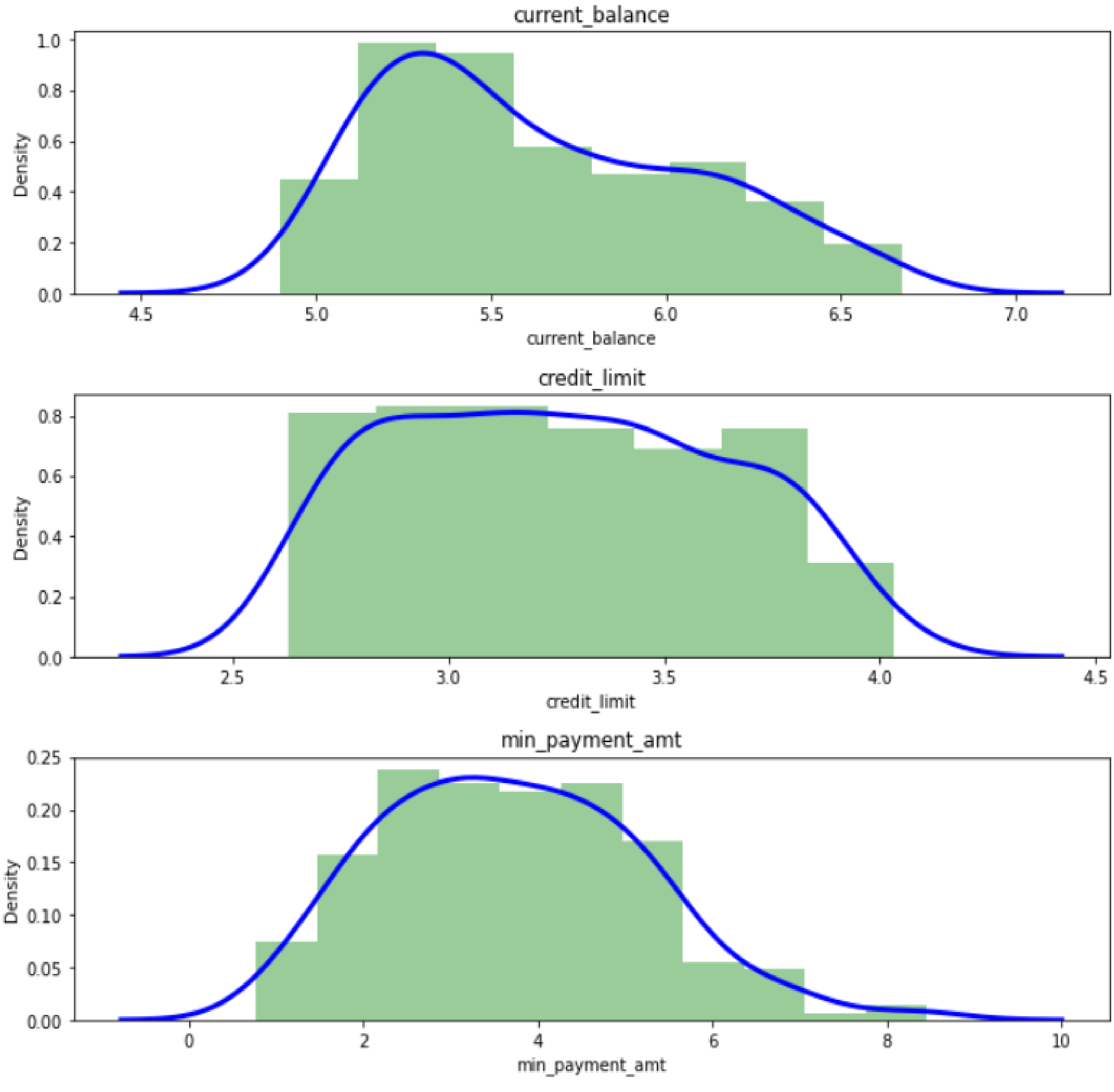
Checking the skewness values quantitatively:

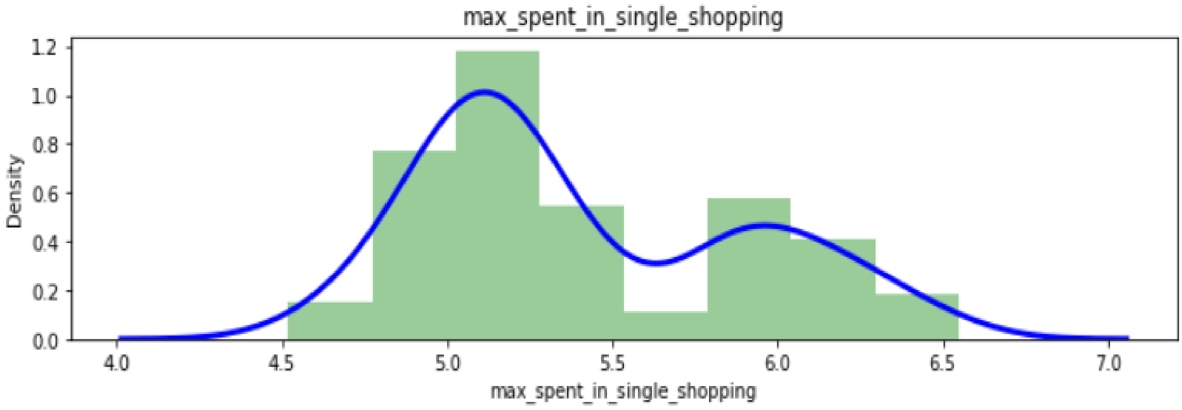
```
max_spent_in_single_shopping    0.561897
current_balance                 0.525482
min_payment_amt                 0.401667
spending                        0.399889
advance_payments                0.386573
credit_limit                    0.134378
probability_of_full_payment    -0.537954
dtype: float64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

**KDE is used for visualizing the Probability Density of a continuous variable.*

** KDE demonstrates the probability density at different values in a continuous variable.*




current_balance
Density
1.0
0.8
0.6
0.4
0.2
0.0
4.5
5.0
5.5
6.0
6.5
7.0
current_balance
credit_limit
Density
0.8
0.6
0.4
0.2
0.0
2.5
3.0
3.5
4.0
4.5
credit_limit
min_payment_amt
Density
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
10
min_payment_amt

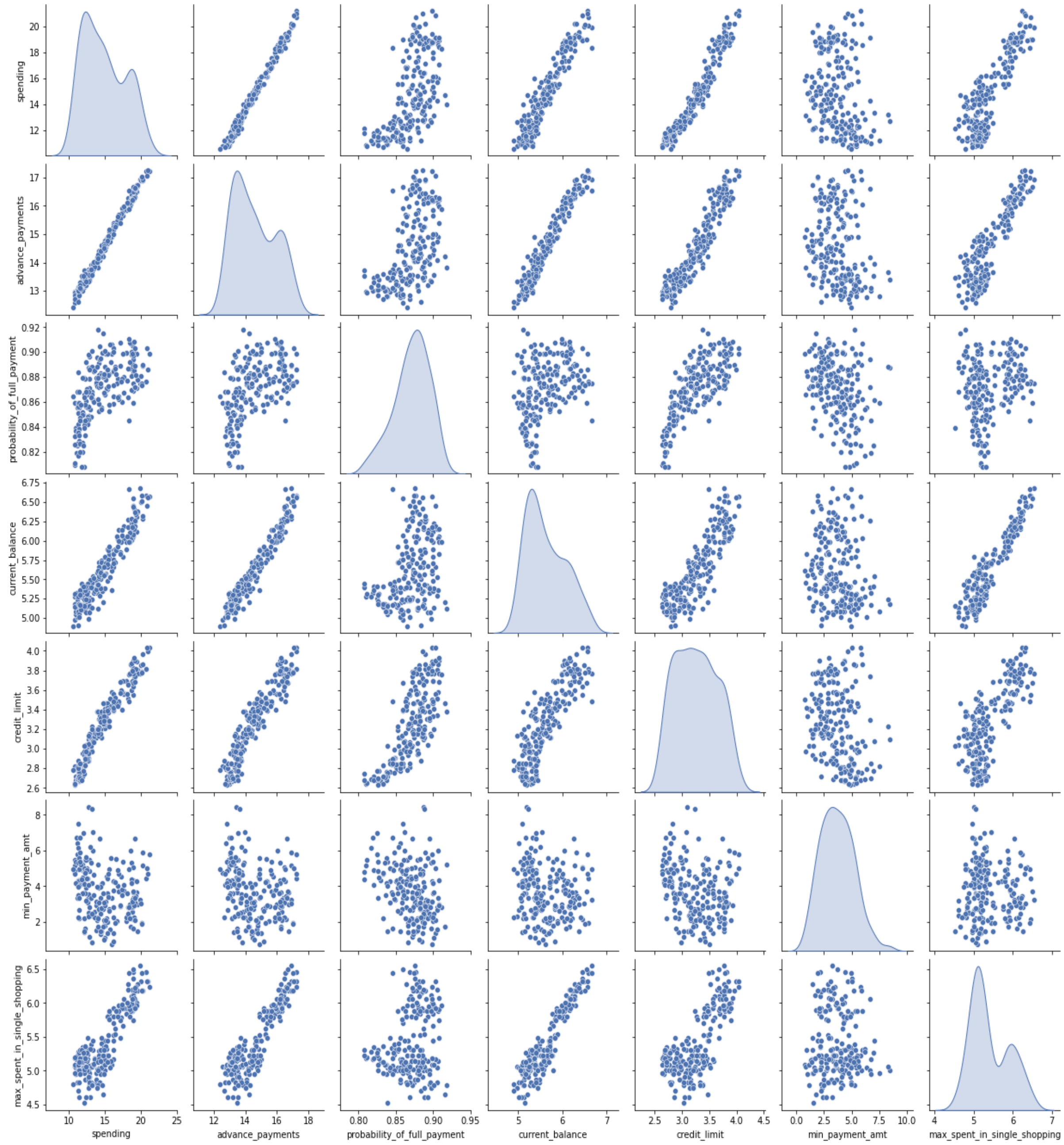


Observations:

- Credit limit average is around $3.258(10000s)
- Distribution is skewed to right tail for all the variable except probability_of_full_payment variable, which has left tail.

Multivariate Analysis:

Checking for multicollinearity:

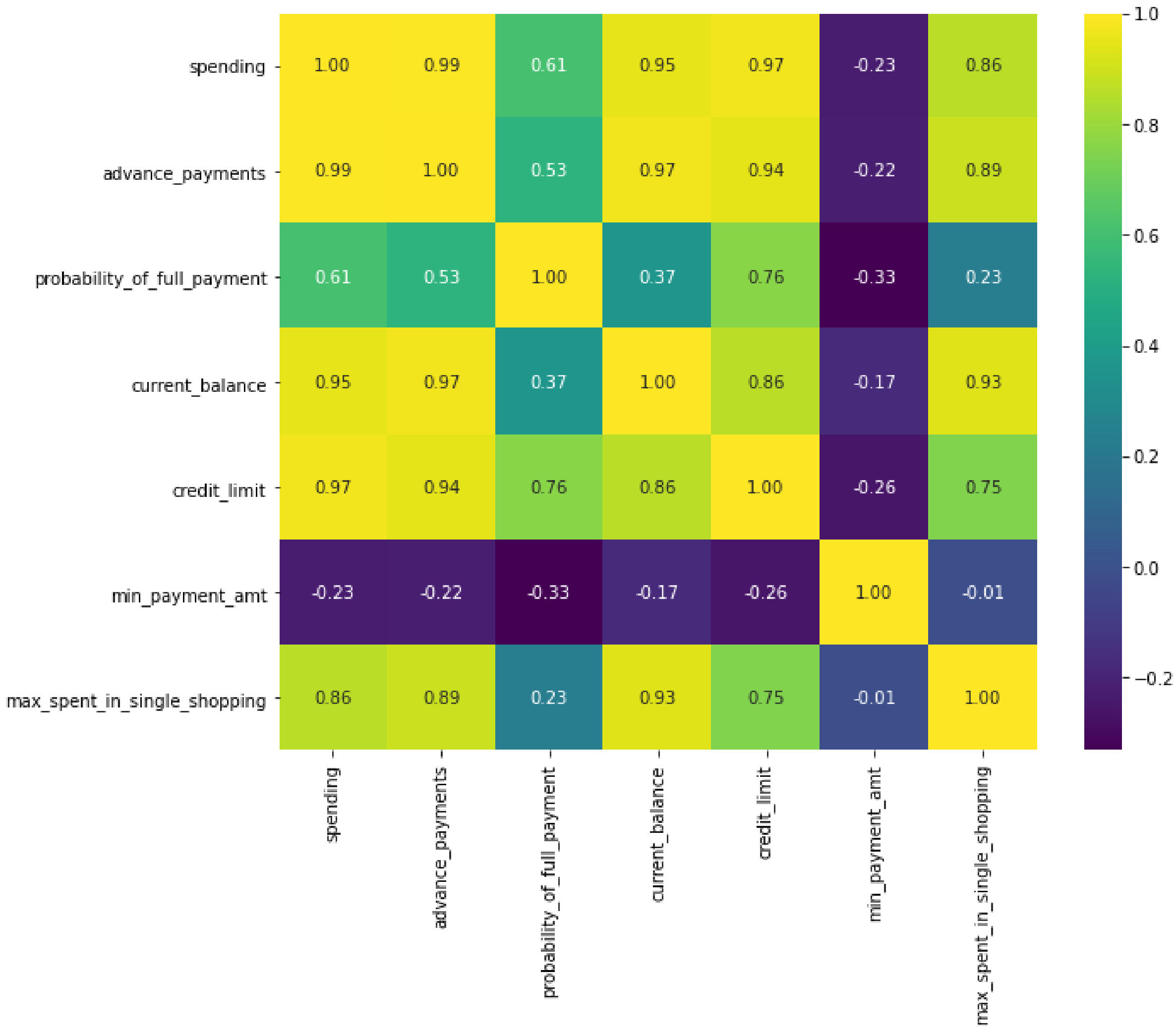


Observations:

- 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

Plotting heat map and correlation table for better visualisation and clear insights:

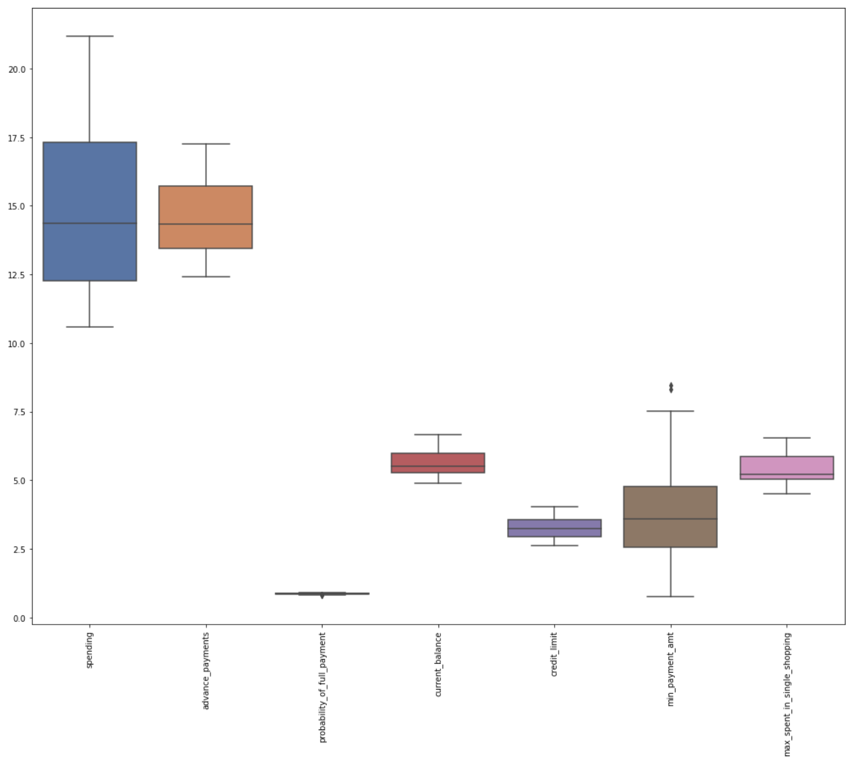
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
spending	1.000000	0.994341	0.608288	0.949985	0.970771	-0.229572	0.863693
advance_payments	0.994341	1.000000	0.529244	0.972422	0.944829	-0.217340	0.890784
ability_of_full_payment	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
nt_in_single_shopping	0.863693	0.890784	0.226825	0.932806	0.749131	-0.011079	1.000000



<u>Let us see the significant correlation either negative or positive among independent attributes:</u>

		correlation
spending	advance_payments	0.994341
advance_payments	current_balance	0.972422
credit_limit	spending	0.970771
spending	current_balance	0.949985
credit_limit	advance_payments	0.944829
max_spent_in_single_shopping	current_balance	0.932806
advance_payments	max_spent_in_single_shopping	0.890784
spending	max_spent_in_single_shopping	0.863693
current_balance	credit_limit	0.860415
probability_of_full_payment	credit_limit	0.761635
max_spent_in_single_shopping	credit_limit	0.749131
spending	probability_of_full_payment	0.608288
advance_payments	probability_of_full_payment	0.529244
current_balance	probability_of_full_payment	0.367915
probability_of_full_payment	min_payment_amt	0.331471

Treating Outliers and plotting on graph:



Observations:

- Most of the outlier has been treated and now we are good to go.
- Though we did treat the outlier, we still see one as per the boxplot, it is okay, as it is no extreme and on lower band.

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

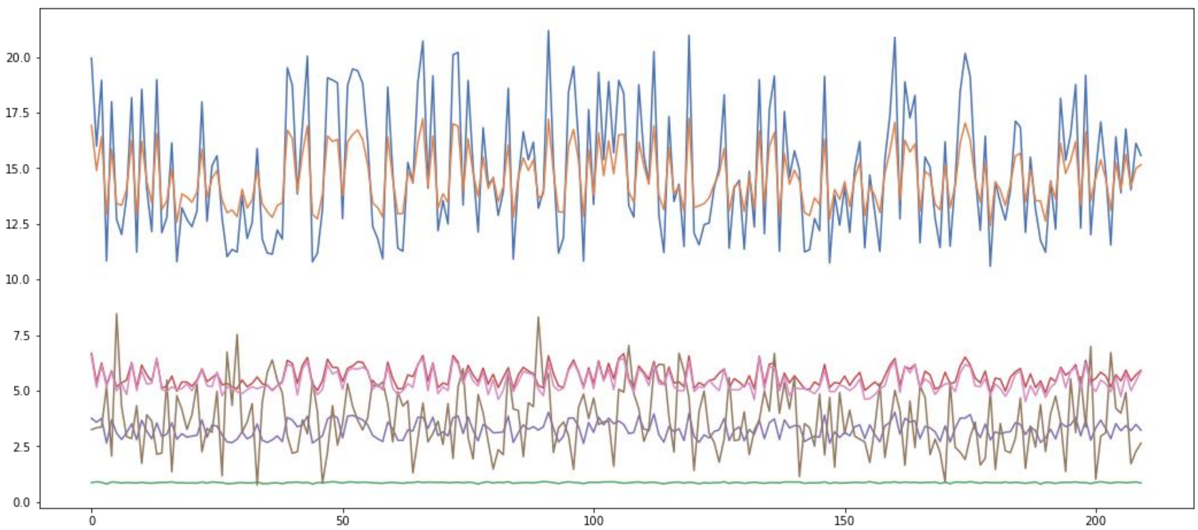
Ans 1.2 Scaling Concept - Feature scaling through standardization (or Z-score normalization) can be an important pre-processing step for many machine learning algorithms. Standardization involves rescaling the features such that they have the properties of a standard normal distribution with a mean of zero and a standard deviation of one.

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 weightage. Also have shown below the plot of the data prior and after scaling. Scaling will have all the values in the relative same range.

- I have used zscore to standardized the data to relative same scale -3 to +3.

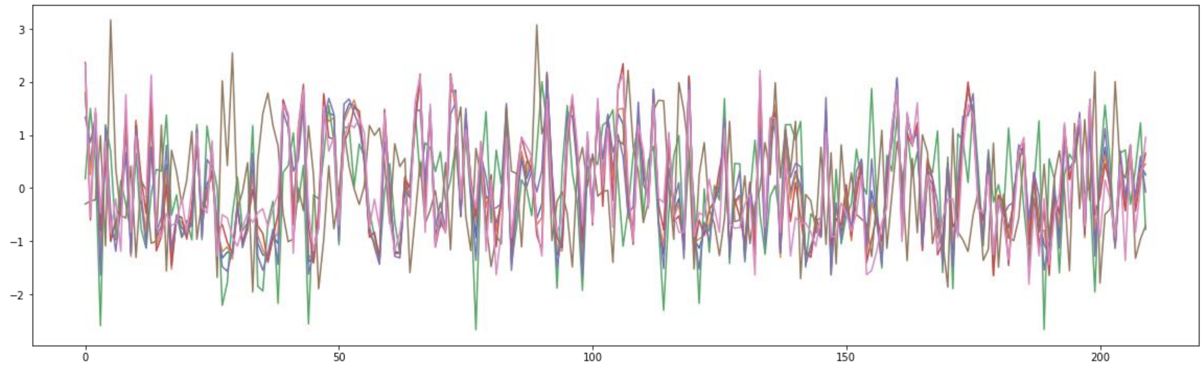
Plotting graph prior to scaling:



Scaling the attributes:

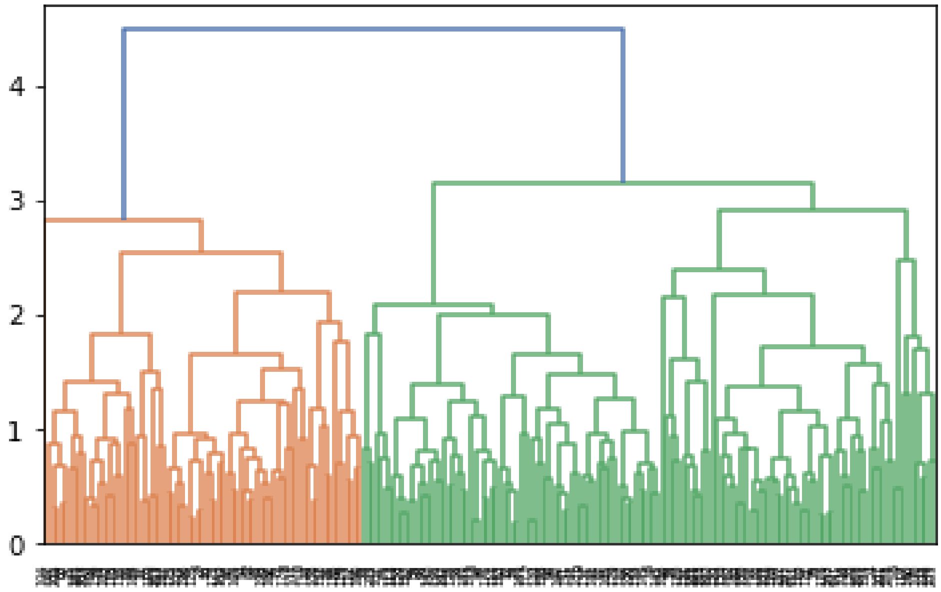
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754355	1.811968	0.178230	2.367533	1.338579	-0.298806	2.328998
1	0.393582	0.253840	1.501773	-0.600744	0.858236	-0.242805	-0.538582
2	1.413300	1.428192	0.504874	1.401485	1.317348	-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

Plotting graph after scaling:

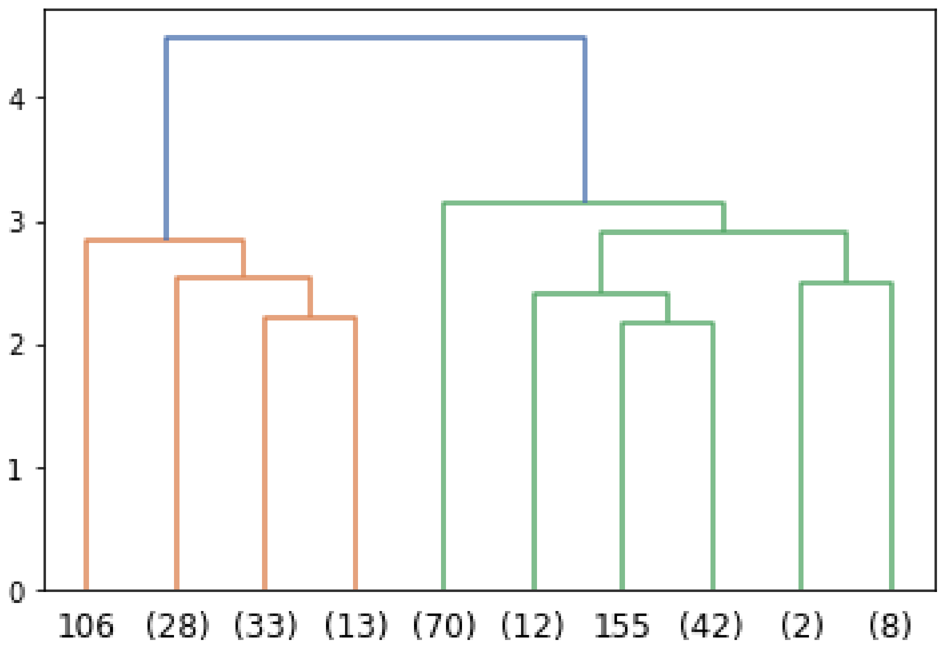


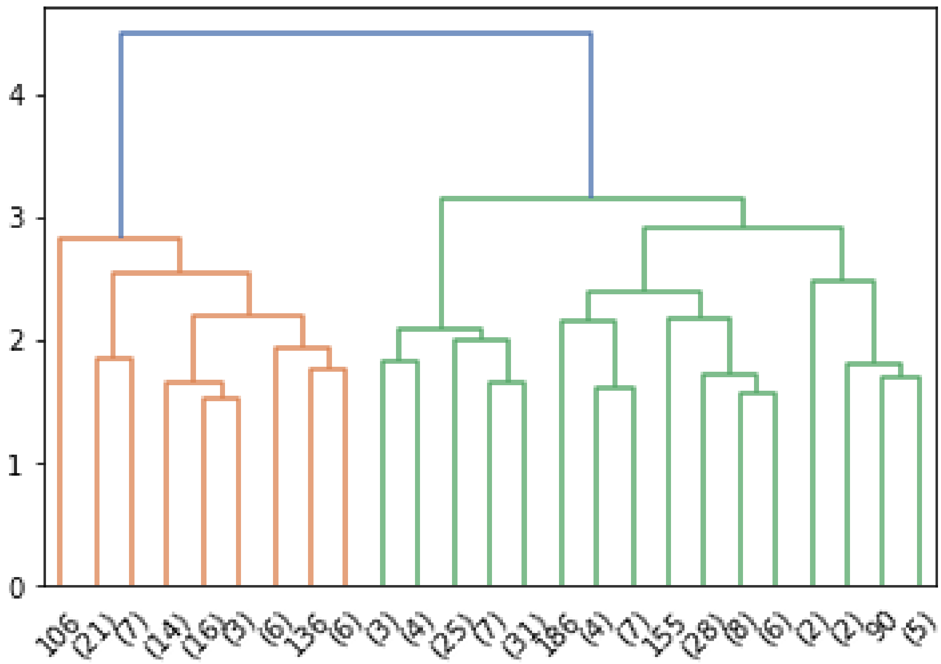
1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them.

Ans 1.3 Applying hierarchical clustering to scaled data using the ward linkage method:



Cutting the Dendrogram with suitable clusters:





Applying fCluster:

```
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)
```

	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

Cluster Profiles:

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

Observations:

- Both the method are almost similar means, minor variation, which we know it occurs.
- We for cluster grouping based on the dendrogram, 3 or 4 looks good. Did the further analysis, and based on the dataset had gone for 3 group cluster solution based on the hierarchical clustering
- Also in real time, there could have been more variables value captured - tenure, BALANCE_FREQUENCY, balance, purchase, instalment of purchase, others.
- And three group cluster solution gives a pattern based on high/medium/low spending with max_spent_in_single_shopping (high value item) and probability_of_full_payment (payment made)

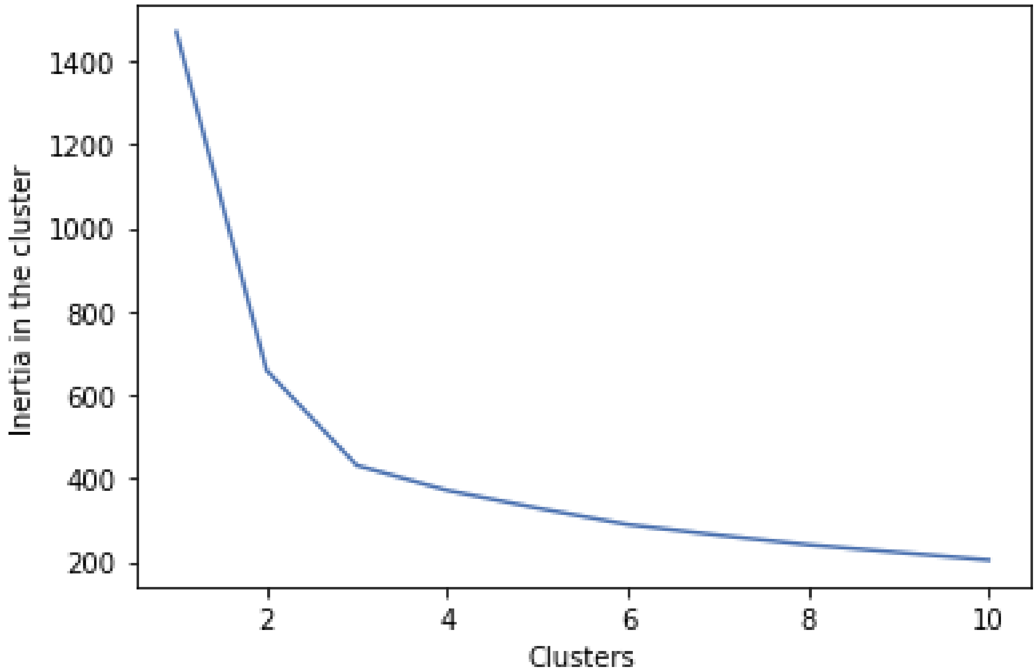
1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score.

Ans 1.4 <u>Applying the K-Means clustering on scaled data and the values are below:</u>

```
[1469.9999999999998,
 659.171754487041,
 430.6589731513006,
 371.30172127754196,
 328.61392616438127,
 289.215290274911,
 263.6557421107544,
 240.71443555253848,
 222.27596196077255,
 204.0231747492838]
```

Values of K-Means inertia value from cluster 1 to 10 using append function.

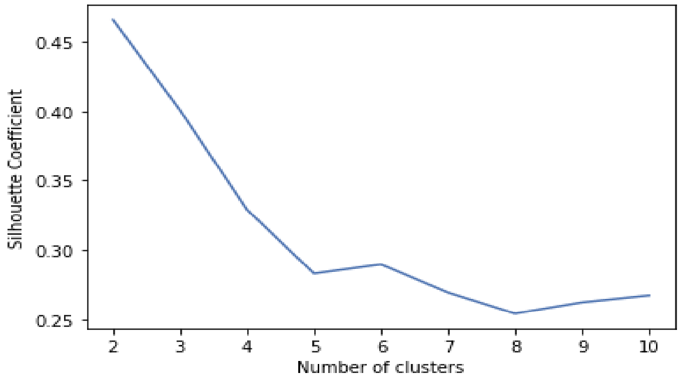
<u>Elbow Curve:</u>



As we can see that there is significant drop from cluster 0 to 2 and beyond the cluster point 3 curve is gradually decreasing. So, from the above information we can infer that optimum number of clusters = 3.

silhouette score:

```
[0.46577247686580914,
 0.4007270552751299,
 0.3291966792017613,
 0.28316654897654814,
 0.2897583830272518,
 0.269484435516853 5,
 0.25437316027505635,
 0.2623959398663564,
 0.2673980772529917]
```



Insights:

The smallest value of silhouette score is 0.009e and it is positive. We can infer that there is n o observation or no customer records that is incorrectly mapping.

From SC Score, we can infer that the number of optimal clusters could be 3 or 4.

3 Cluster Solution:

Clusters Value Counts
1 72
2 71
0 67

Plotting the mean of 3 clusters:

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

Observations:

- We can infer that maximum average spending is done by cluster 1.
- Probability of full payment is equal for cluster 1 & 3.
- Cluster 1 have highest credit limit & Max_spent_in_shopping.

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

Ans 1.5 **Clusters Profiles: 3 group cluster via Kmeans**

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

3 group cluster via hierarchical clustering:

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

Cluster Group Profiles

Group 1: High Spending

Group 3: Medium Spending

Group 2: Low Spending

Promotional strategies for each cluster

Group 1: High Spending Group

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

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 cars 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

Group 2: Low Spending Group

- customers should be given remainders for payments. Offers can be provided on early payments to improve their payment rate.
- Increase their spending habits by tying up with grocery stores, utilities (electricity, phone, gas, others)

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

Attribute Information:

1. Target: Claim Status (Claimed)
2. Code of tour firm (Agency_Code)
3. Type of tour insurance firms (Type)
4. Distribution channel of tour insurance agencies (Channel)
5. Name of the tour insurance products (Product)
6. Duration of the tour (Duration)
7. Destination of the tour (Destination)
8. Amount of sales of tour insurance policies (Sales)
9. The commission received for tour insurance firm (Commission)
10. Age of insured (Age)

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

Ans 2.1 Reading the data with initial steps:

	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

	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

Top 5 and the bottom 5 of the dataset looks good respectively.

Checking the shape, information of the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Age           3000 non-null   int64  
 1   Agency_Code   3000 non-null   object 
 2   Type          3000 non-null   object 
 3   Claimed       3000 non-null   object 
 4   Commision     3000 non-null   float64
 5   Channel       3000 non-null   object 
 6   Duration      3000 non-null   int64  
 7   Sales         3000 non-null   float64
 8   Product Name  3000 non-null   object 
 9   Destination   3000 non-null   object 
dtypes: float64(2), int64(2), object(6)
memory usage: 234.5+ KB
```

Observations:

- There are 10 variables in dataset out of which 4 variables are numeric data type and 6 variables are categorical data type.
- Age, Commission, Duration, Sales are numeric variables.
- Agency Code, Type, Claimed, Channel, Product Name and Destination are categorial variables.
- There are 3000 rows and 10 columns in dataset.
- 9 independent variable and one target variable – Claimed
- No Missing Values

Checking descriptive summary of Numeric data type:

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

Insights:

- Duration has negative value; it is not possible. Wrong entry.
- Commission & Sales- mean and median varies significantly.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	3000	NaN	NaN	NaN	38.091	10.4635	8	32	36	42	84
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	NaN	NaN	NaN	14.5292	25.4815	0	0	4.63	17.235	210.21
Channel	3000	2	Online	2954	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Duration	3000	NaN	NaN	NaN	70.0013	134.053	-1	11	26.5	63	4580
Sales	3000	NaN	NaN	NaN	60.2499	70.734	0	20	33	69	539
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

Insights:

- Categorial code variable maximum unique count is 5.

Getting unique counts of all Nominal Variables:

```
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
```

Checking for Duplicates:

Number of duplicate rows = 139

Removing Duplicates - Not removing them

no unique identifier, can be different customer.

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

Univariate Analysis

Descriptive statistics of all features:

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

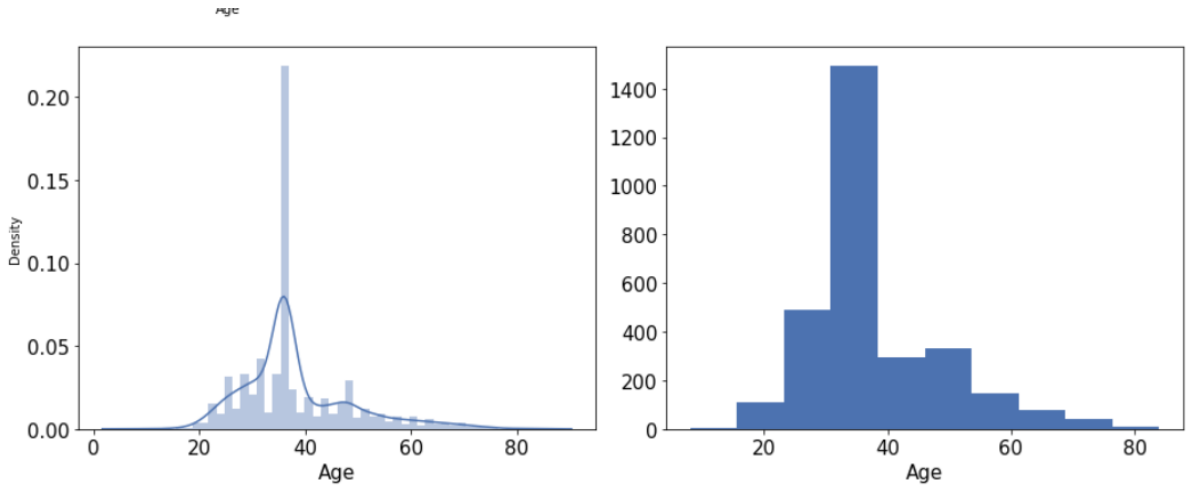
Individual Analysis of all the attributes:

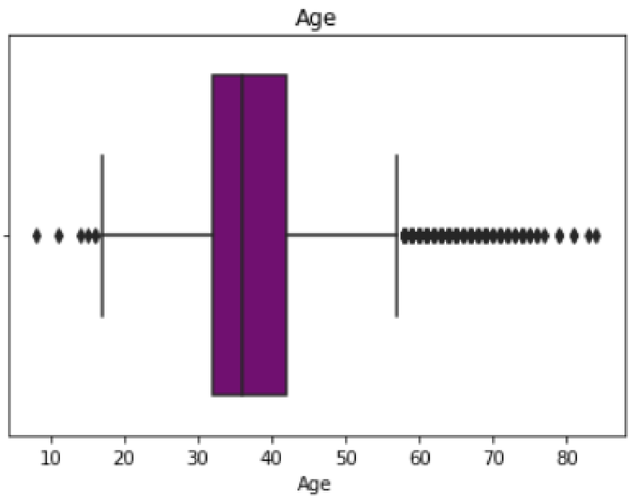
1. <u>Age Variables</u>

```
spending - 1st Quartile (Q1) is:  32.0
spending - 3st 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


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 %
```



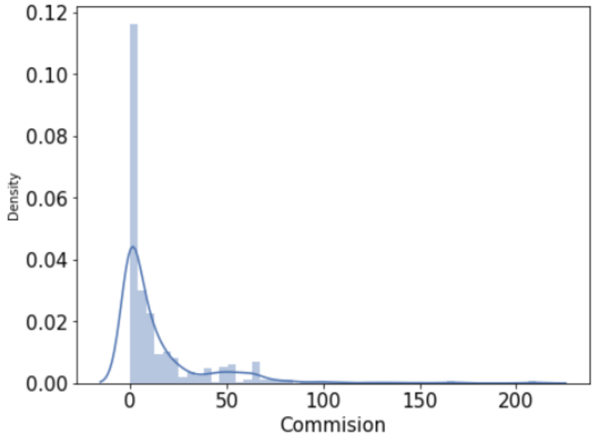


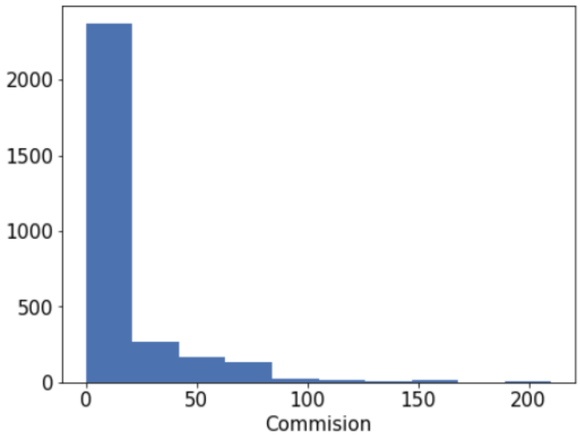
2. <u>Commisson Variable</u>

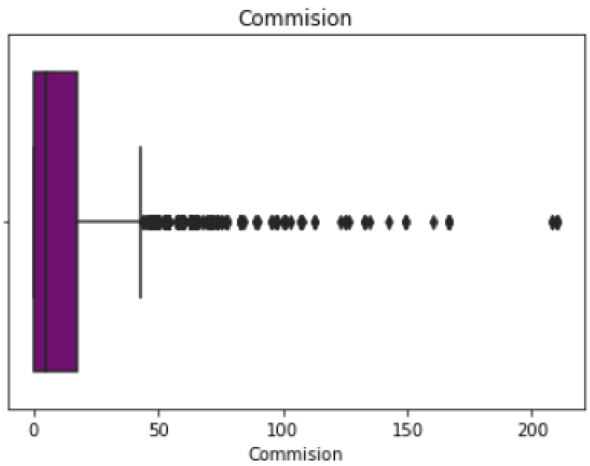
```
Commision - 1st Quartile (Q1) is:  0.0
Commision - 3st Quartile (Q3) is:  17.235
Interquartile range (IQR) of Commision is  17.235


Lower outliers in Commision:  -25.8525
Upper outliers in Commision:  43.0875


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





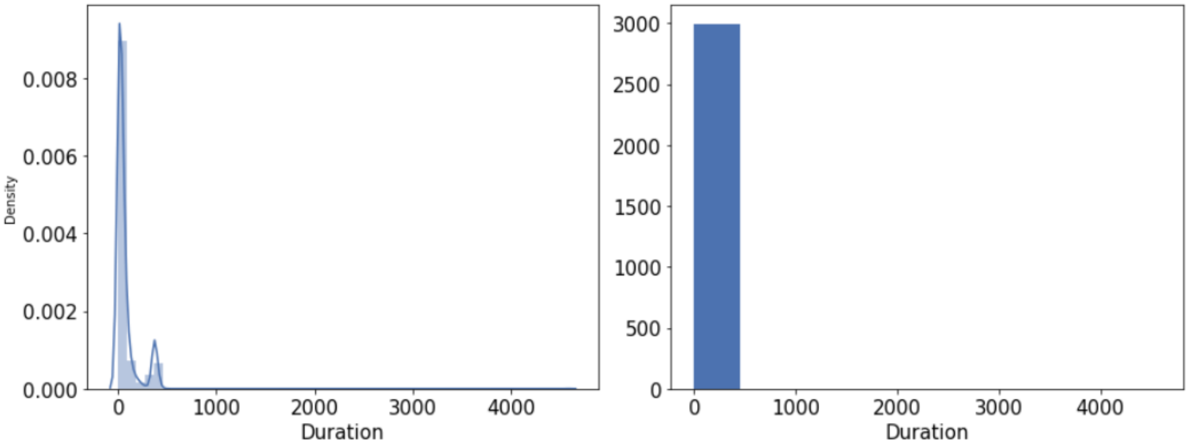


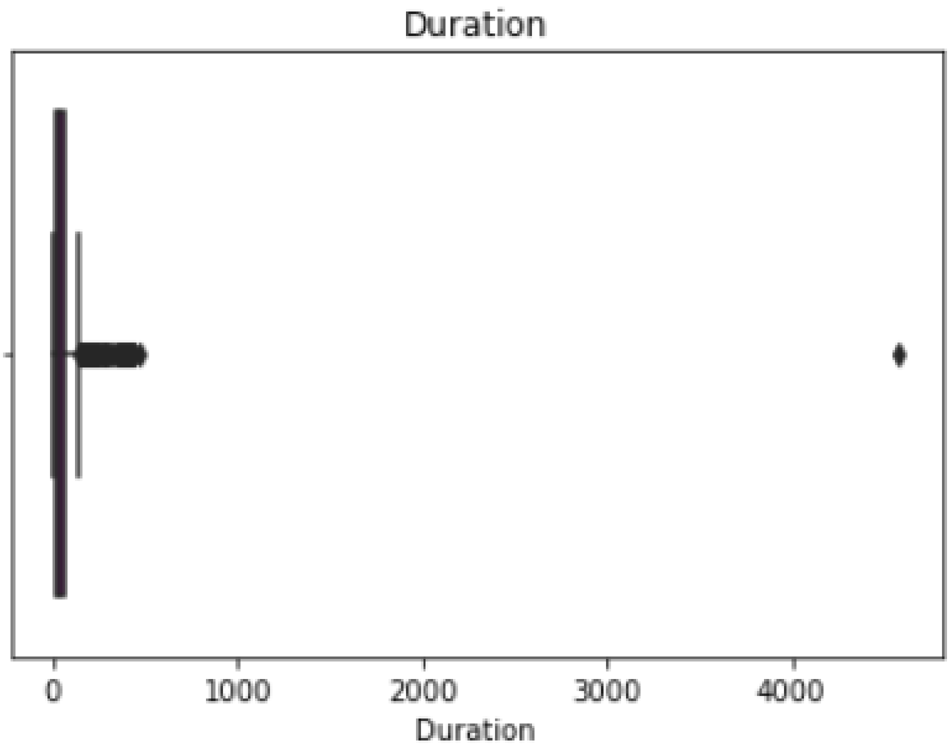
3. <u>Duration Variable</u>

```
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 %
```



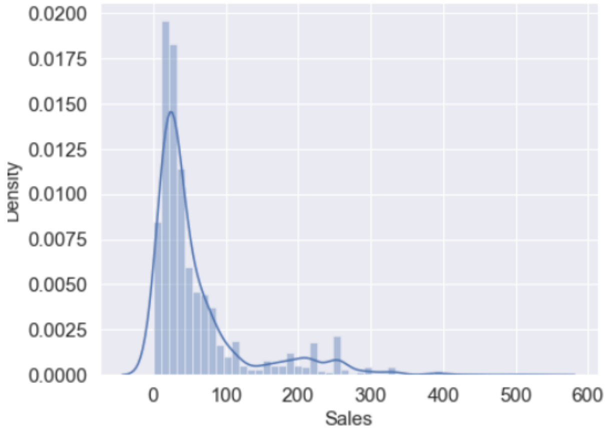


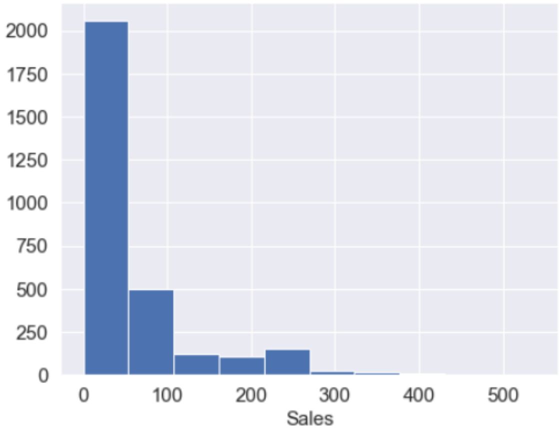
4. <u>Sales Variable</u>

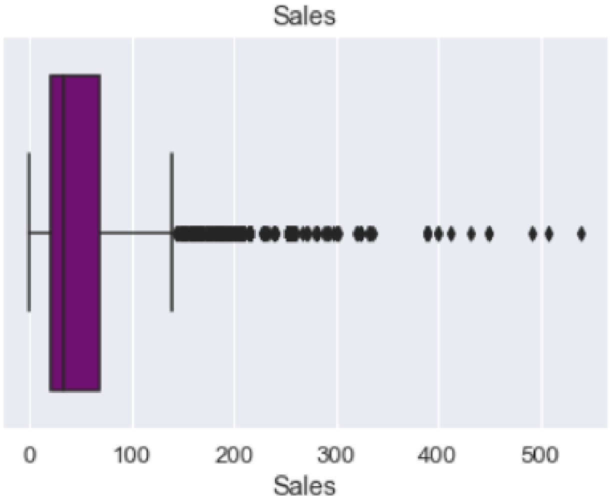
```
Sales - 1st Quartile (Q1) is:  20.0
Sales - 3st Quartile (Q3) is:  69.0
Interquartile range (IQR) of Sales is  49.0


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 %
```







Insights:

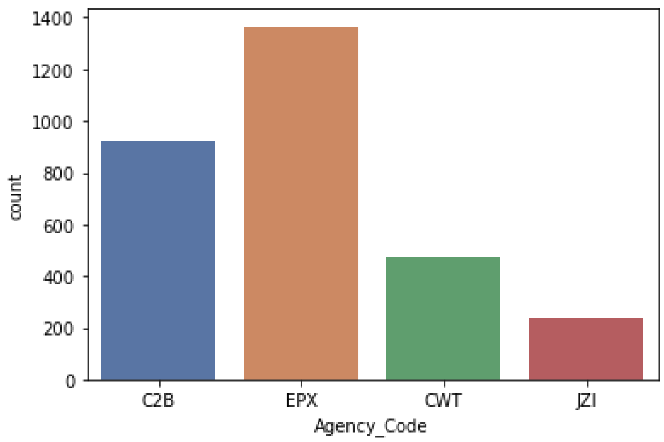
- There are outliers in all the variables, but the sales and commission can be a genius business value. Random Forest and CART can handle the outliers. Hence, Outliers are not treated for now, we will keep the data as it is.

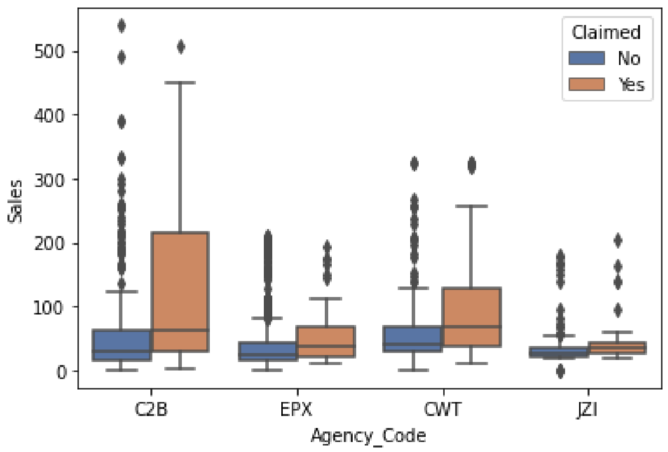
- I will treat the outliers for the ANN model to compare the same after the all the steps just for comparison.

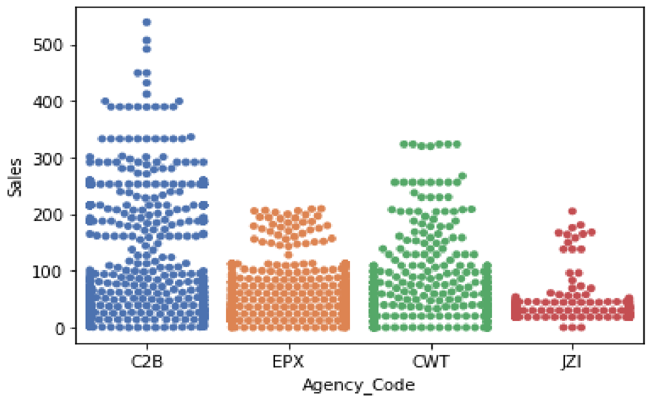
Categorical Variables

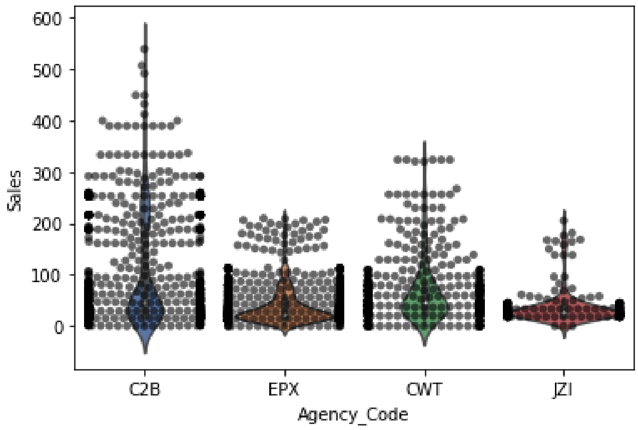
Plotting Cat variables using Box plot, Bar Plot, Swarm Plot and Violin plot.

1. Agency Code

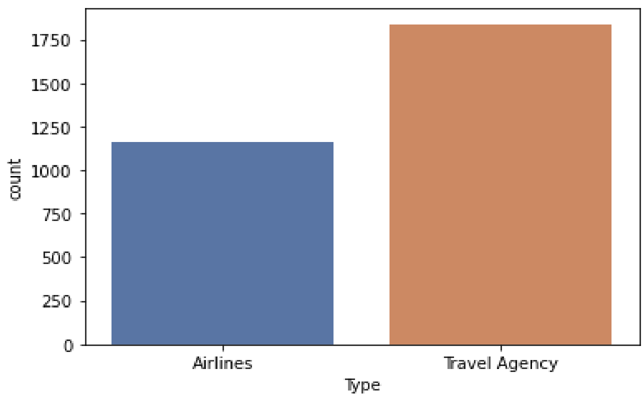


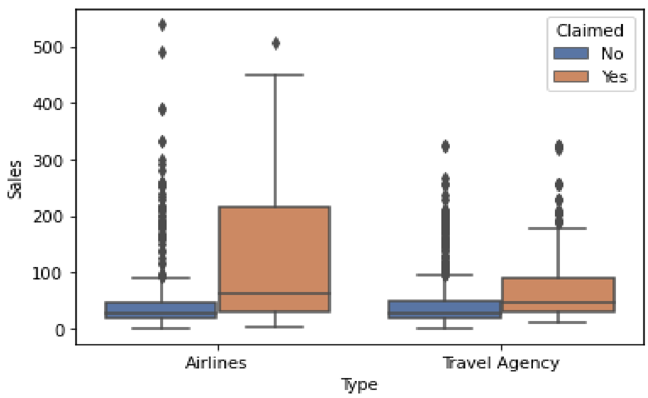


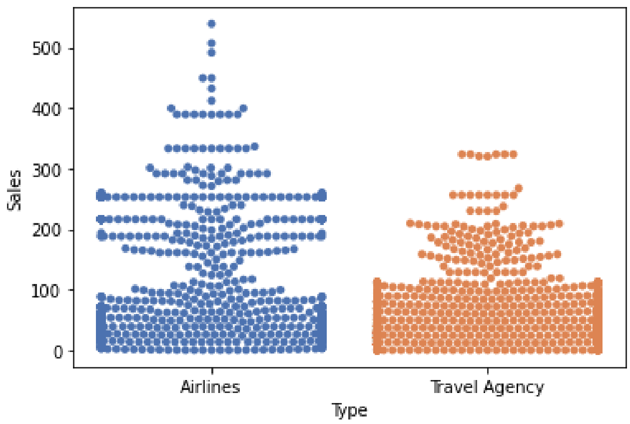




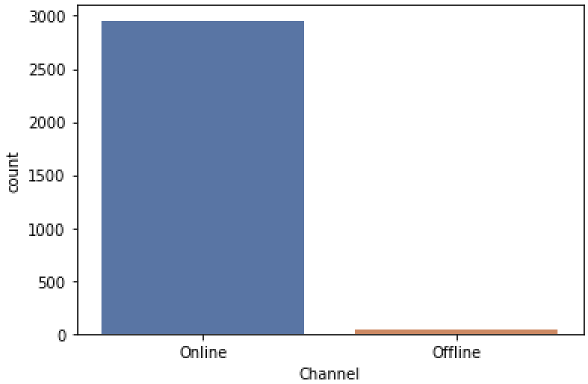
2. Type Variable

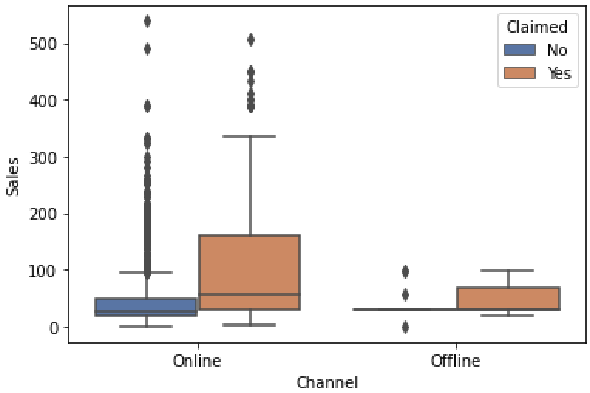


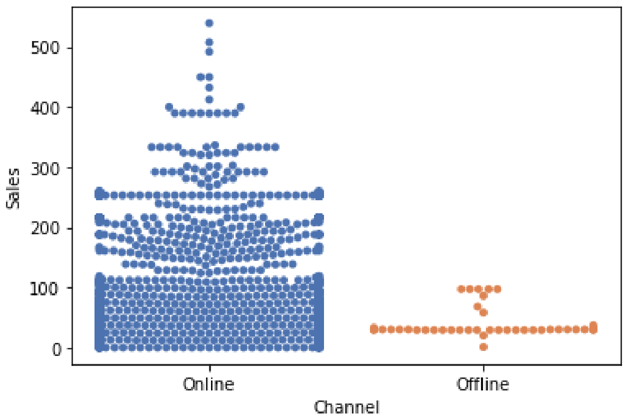




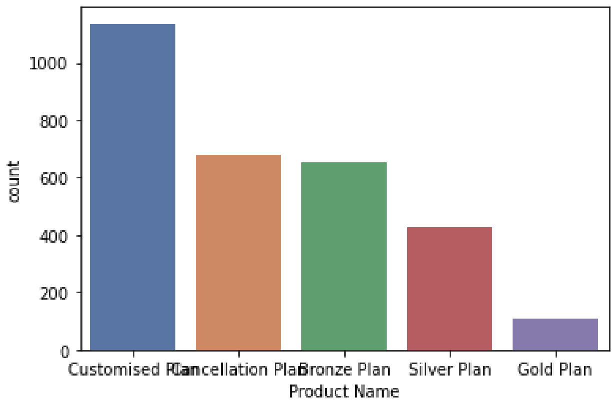
3. Channel Variable

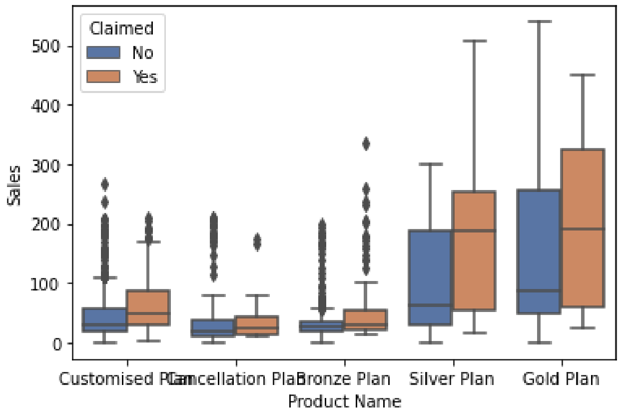


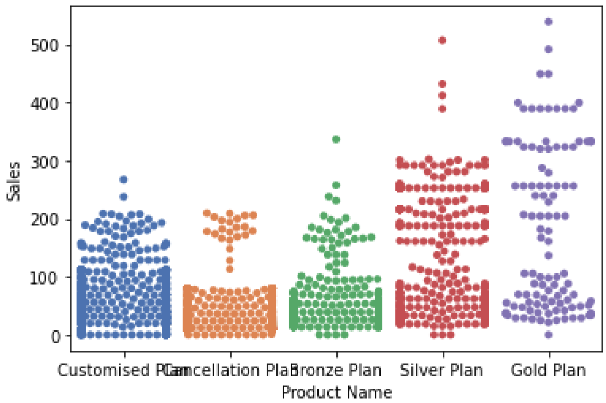


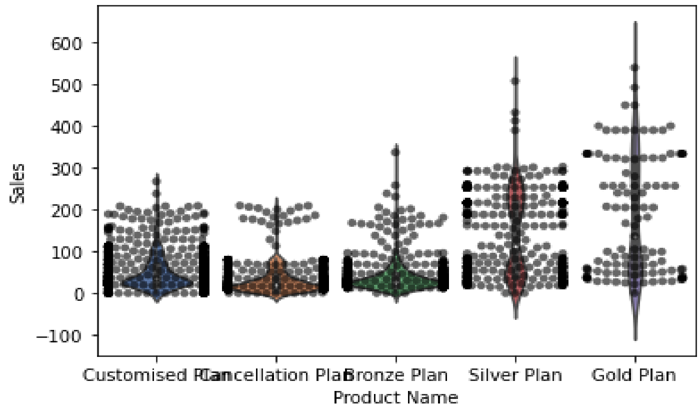


4. Product Name Variable

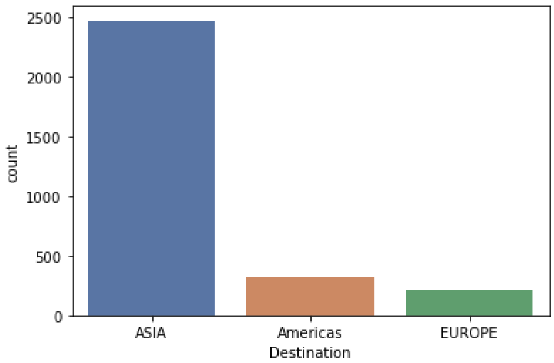


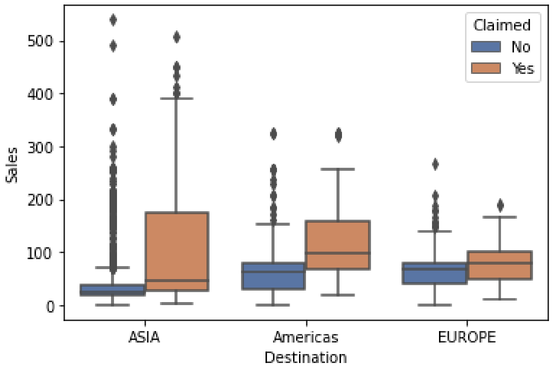


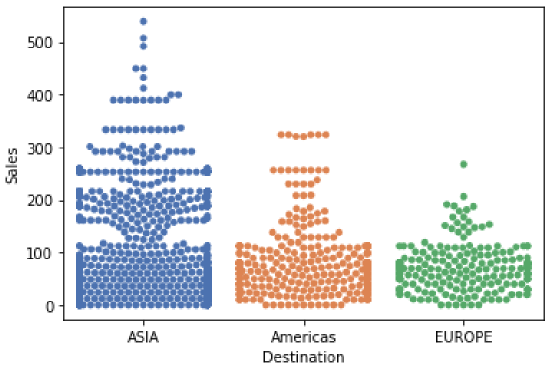


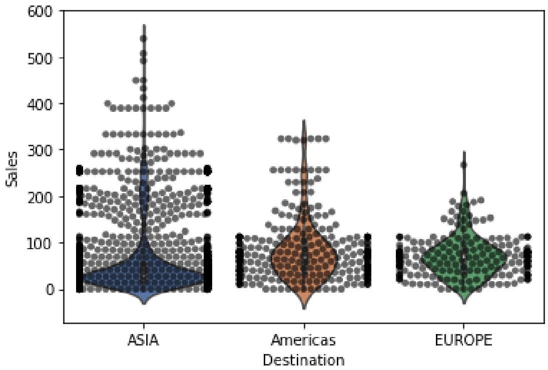


5. Destination Variable

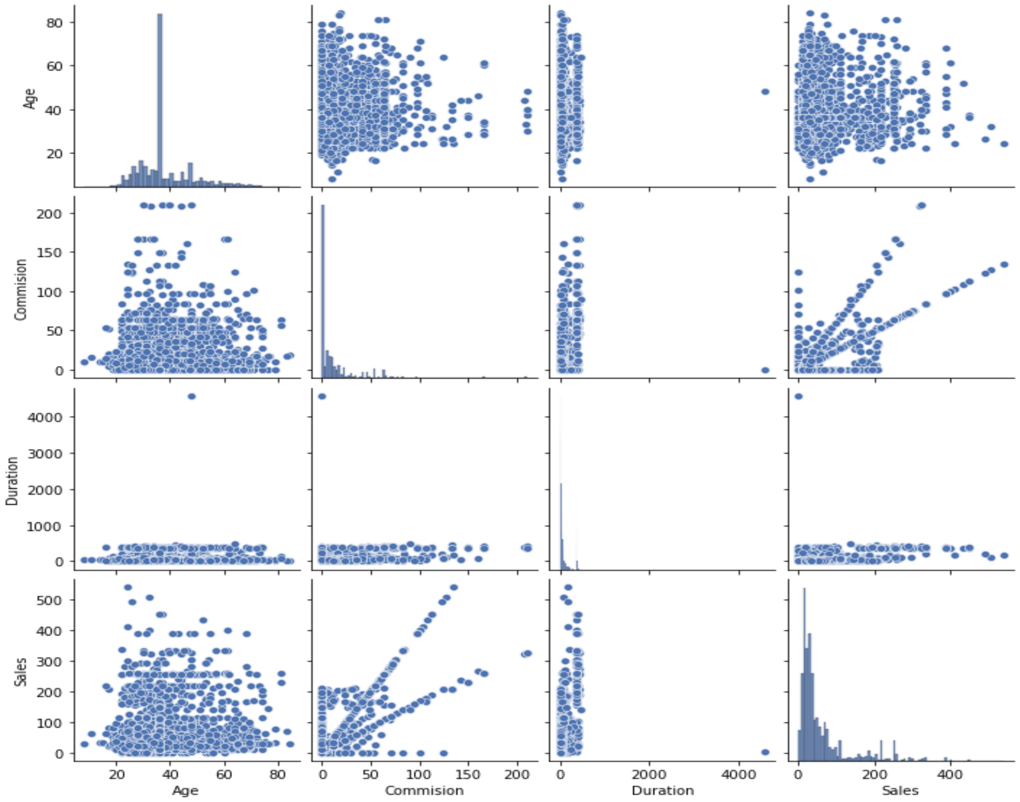








<u>**Multivariate Analysis**</u>



Checking for correlations:

Converting the Categorical data into codes and checking the information of the dataset:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Age           3000 non-null   int64  
 1   Agency_Code   3000 non-null   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

- Now, all the variables are numeric datatype.

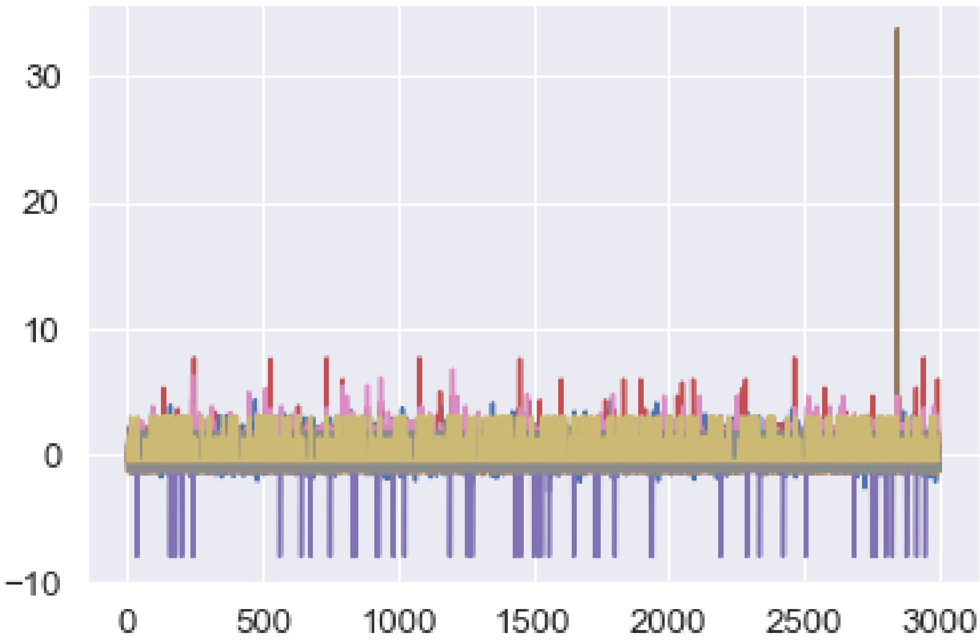
2.2 Data Split: Split the data into test and train, build classification model CART, Random Forest, Artificial Neural Network

Ans 2.2 Drooping the variable “Claimed” before splitting the data into train and test data. It also requires scaling before splitting.

	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

Scaling the data and plotting it on graph for better data visualization using Z score:

	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



Checking the dimensions of the training and test data:
X_train (2100, 9)
X_test (900, 9)
train_labels (2100,)
test_labels (900,)

Building a Decision Tree Classifier

{'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)

Variable Importance - DTCL

	Imp
Agency_Code	0.634112
Sales	0.220899

Product Name 0.086632
Commision   0.021881
Age   0.019940
Duration   0.016536
Type   0.000000
Channel   0.000000
Destination   0.000000

Getting the Predicted Classes and Probs

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

Building a Random Forest Classifier

{'max_depth': 6, 'max_features': 3, 'min_samples_leaf': 8, 'min_samples_split': 46, 'n_estima
tors': 350}

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

Getting the Predicted Classes and Probs

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

Variable Importance via RF

	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

Building a Neural Network Classifier

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

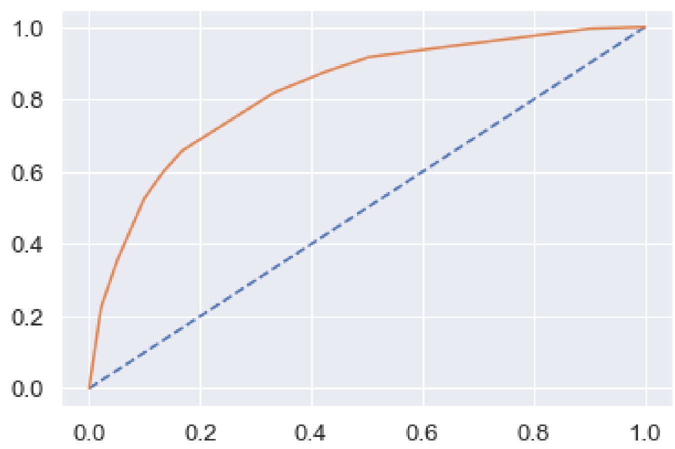
Getting the Predicted Classes and Probs

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

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

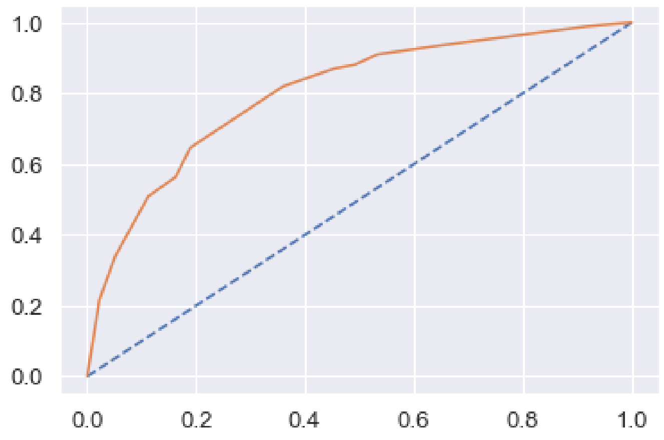
Ans 2.3 CART - AUC and ROC for the training data

AUC : 0.823



CART -AUC and ROC for the test data

AUC: 0.801



CART Confusion Matrix and Classification Report for the training data

Confusion Matrix :

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

Classificaton Report:

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

```
cart_train_precision  0.7
cart_train_recall  0.53
cart_train_f1  0.6
```

CART Confusion Matrix and Classification Report for the testing data

Confusion Matrix:

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

Classification Report:

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

```
cart_test_precision  0.67
cart_test_recall  0.51
cart_test_f1  0.58
```

Cart Conclusion

Train Data:

- AUC: 82%
- Accuracy: 79%
- Precision: 70%
- f1-Score: 60%

Test Data:

- AUC: 80%
- Accuracy: 77%
- Precision: 80%
- f1-Score: 84%

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

Change is the most important variable for predicting diabetes

RF Model Performance Evaluation on Training Data

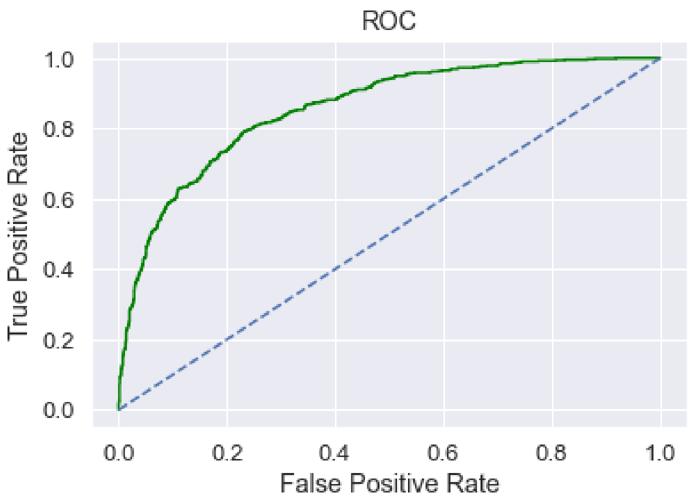
Confusion Matrix

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

Classification Report

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

Area under Curve is 0.8563713512840778
RF Accuracy score on Training Data: 0.8042857142857143



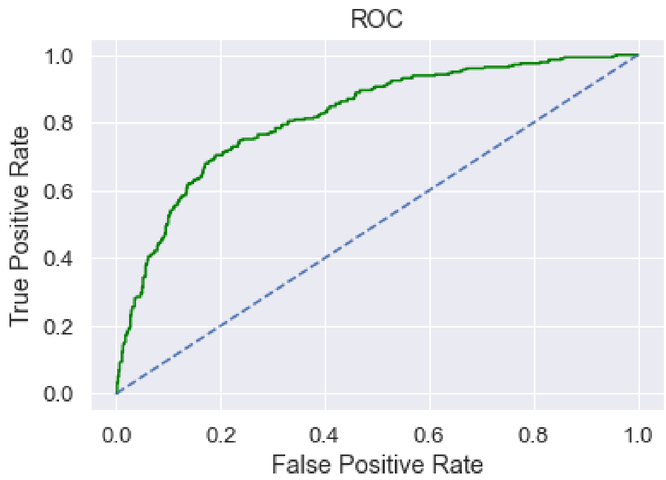
RF Model Performance Evaluation on Test data

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

Classification Report

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

Area under Curve is 0.8181994657271499
RF Accuracy score on Test Data: 0.7844444444444445



Random Forest Conclusion

Train Data:

- AUC: 86%
- Accuracy: 80%
- Precision: 72%
- f1-Score: 66%

Test Data:

- AUC: 82%
- Accuracy: 78%
- Precision: 68%
- f1-Score: 62

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

Change is again the most important variable for predicting diabetes

NN Model Performance Evaluation on Training data

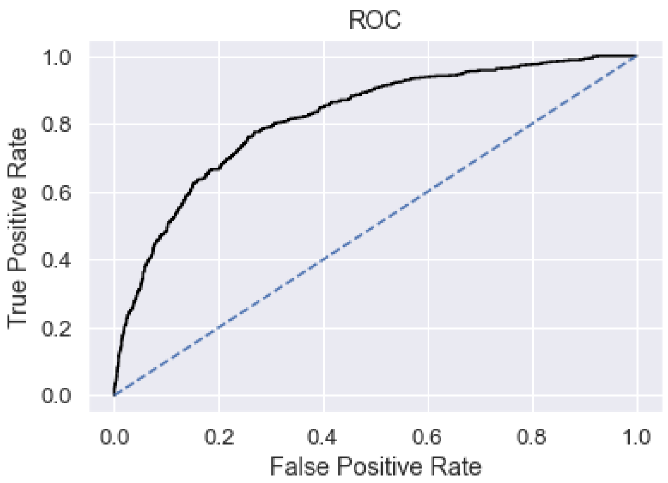
Confusion Matrix

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

Classification Report

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

Area under Curve is 0.8166831721609928
Accuracy Score - 0.7761904761904762



NN Model Performance Evaluation on Test data

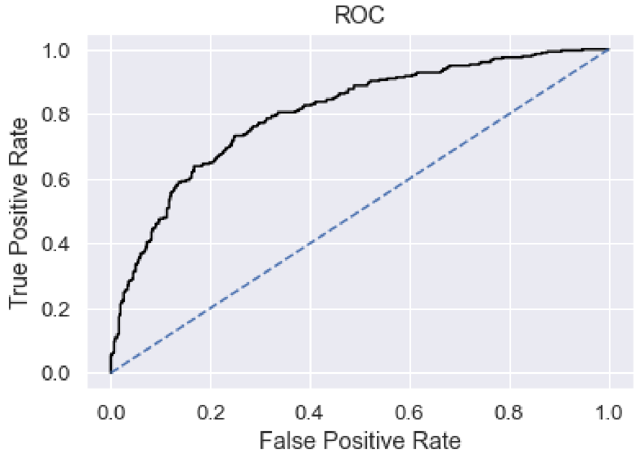
Confusion Matrix

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

Classification report

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

Area under Curve is 0.8044225275393896
Accuracy Score - 0.7688888888888888



Neural Network Conclusion

Train Data:

- AUC: 82%
- Accuracy: 78%
- Precision: 68%
- f1-Score: 59

Test Data:

- AUC: 80%
- Accuracy: 77%
- Precision: 67%
- f1-Score: 57%

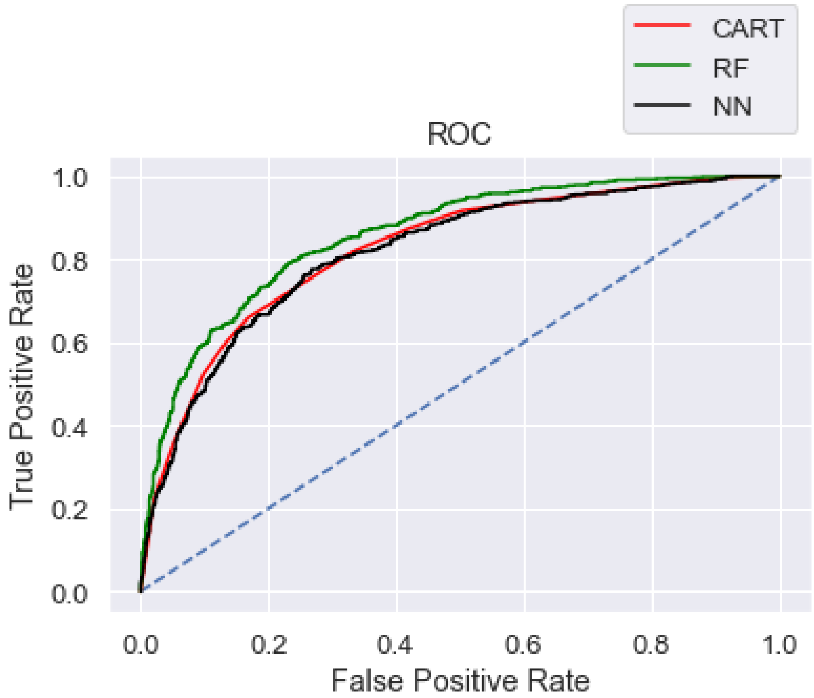
Training and Test set results are almost similar, and with the overall measures high, the model is a good model.

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

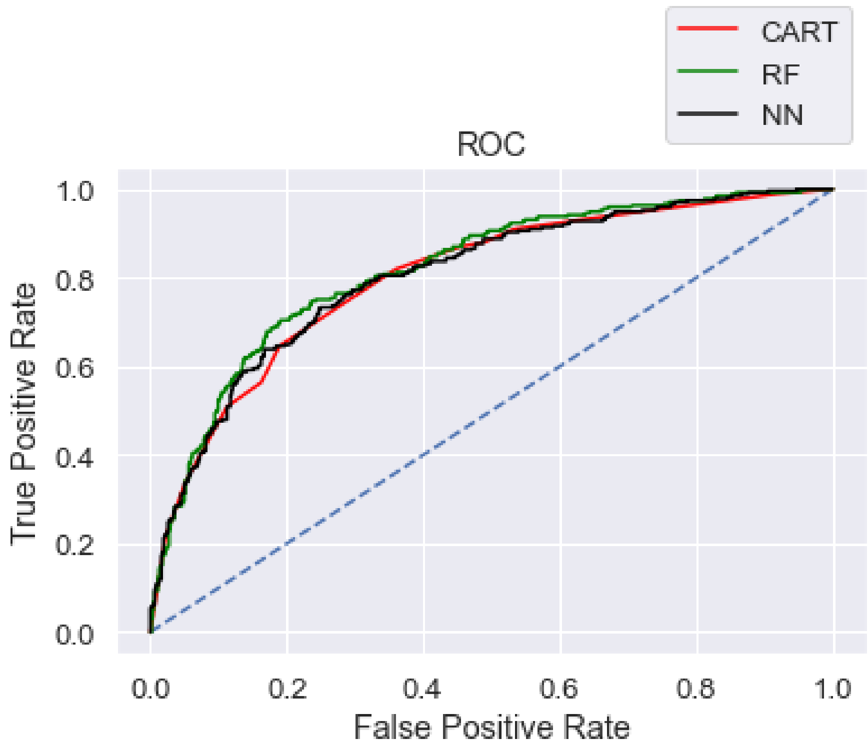
Ans 2.4 Comparison of the performance metrics from the 3 models

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

ROC Curve for the 3 models on the Training data



ROC Curve for the 3 models on the Test data



CONCLUSION:

I am selecting the RF model, as it has better accuracy, precision, recall, f1 score better than other two CART & NN

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

Ans 2.5

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, behaviour 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.

The End.