BUSINESS REPORT 2021

DATA MINING

SUBJECT

Name: Harsh Alkesh Pandya

Batch: DSBA_FEB_A-2021

Problem1: CLUSTERING

Bank Marketing

Problem 2: CART-RF-ANN

Insurance

TABLE OF CONTENTS

SUMMARY 6

Probleam 1: CLUSTERING 7

1.1 Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis). 7
1.2 Do you think scaling is necessary for clustering in this case? Justify 15
1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them. 16
1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Explain the results properly. Interpret and write inferences on the finalized clusters. 19
1.5 Describe cluster profiles for the clusters defined. Recommend different promotional strategies for different clusters. 22

Probleam 2: CART-RF-ANN 23

2.1 Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis). 23
2.2 Data Split: Split the data into test and train, build classification model CART, Random Forest, Artificial Neural Network 37
2.3 Performance Metrics: Comment and Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score, classification reports for each model. 40
2.4 Final Model: Compare all the models and write an inference which model is best/optimized. 46
2.5 Inference: Based on the whole Analysis, what are the business insights and recommendations 48

Conclusion 49

LIST OF TABLES:

Probleam 1

1.Data Head------------------7

2. Describe-----------------------8

3. Scaled Data----------------------15

4. Cluster Frequency Data Head-------17

5. Cluster 3 Data Head-------------------------17

6. Ward Claustering Data--------------------------18

7. Cluster Profiling--------------18

8. Shihotal Width Score----------------20

9. Frequency Data----------------------------21

Probleam 2

10. Data Head----------------23

11. Describe Data----------------25

12. Duplicate List---------------------26

13. All Data List----------------------------46

LIST OF FIGURE:

Probleam 1

1. Import Library------------7

2. Information Output------------8

3. Spending----------------------------9

4. Advance Pay----------------------------10

5. Probability----------------------------------10

6. Current Balance--------------------------------11

7. Credit Limit---------------11

8. Min Spend---------------------12

9. Max Spend-----------------------12

10. Multivariates.----------------------13

11. Heat Map---14

12. Average Linkage Method--------------16

13. Ward Linkage Method-------------------18

14. Clustering Output------------------------------19

15. WSS Output---19

16. Shiotal Output--20

17. Clustering Map--20

Probleam 2

18. Import Data------------23

19. Information-----------------24

20. Missing Value-------------------24

21. Outlayers Check----------------------26

22. Age Destribution-----------------------28

23. Comission Skwed----------------------------28

24. Duration Skwed----------------------29

25. Seles Skwed-------------------------------29

26. Agency Countplot------------------------------30

27. Agency Boxplot-----------30

28. Type Countploy------------31

29. Type Boxplot----------------31

30. Channel Countplot----------------32

31. Channel Boxplot-----------------32

32. Product Name Countplo-------------33

33. Product Name Boxplot-----------------33

34. Description Countplot----------------34

35. Description Boxplot--------------------34

36. Multivariate--------------35

37. Heat Map----------------------35

38. Desicition Tree----------------------38

39.Cart Train ROC----------------------------43

40.Cart Test ROC-------------------------------------43

41. RF Train ROC ---43

42. RF Test ROC---------------------44

43. ANN Train ROC-----------------------44

44. ANN Test ROC------------------------------44

45. Fainal Train ROC---------------------------------46

46. Final Test ROC---47

REPORT

SUMMARY

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)

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.

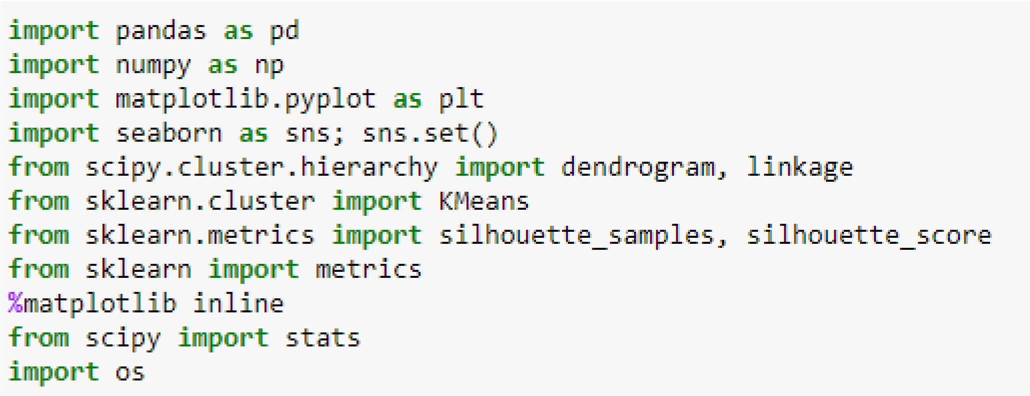
Attribute Information:

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)

PROBLEAM 1: CLUSTERING

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

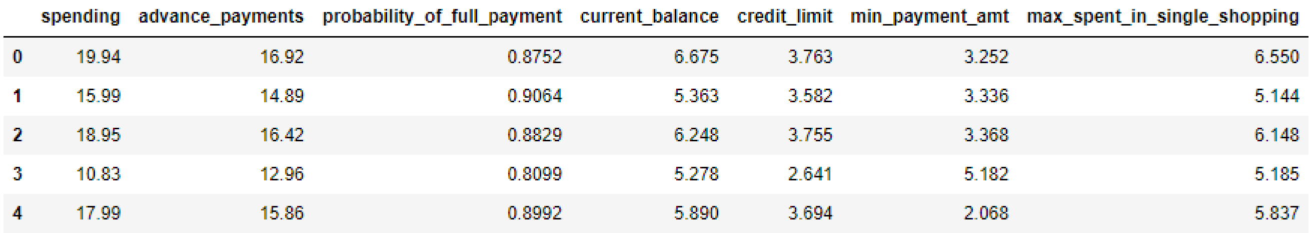
So, we will import all the necessary libraries for cluster analysis,



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples, silhouette_score
from sklearn import metrics
%matplotlib inline
from scipy import stats
import os
```

(Fig: 1 Import)

Reading the data,

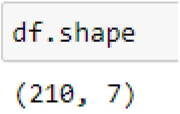


	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148
3	10.83	12.96	0.8099	5.278	2.641	5.182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

(Tab: 1 Head)

- The data seems to be perfect
- The shape of the data is (210, 7)
- The info of the data indicates that all values are float
- No Null values in the data
- No missing values in the data

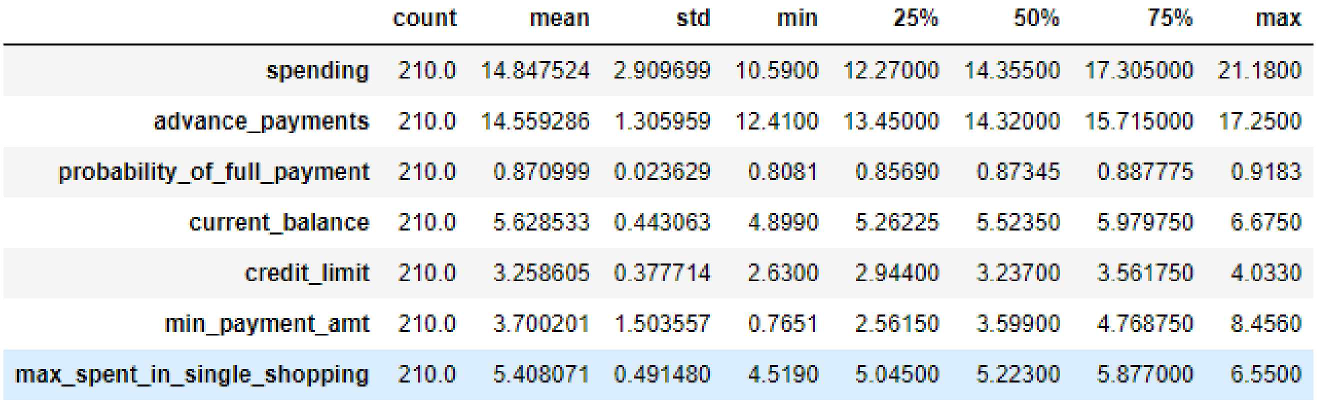
The data has 7 columns and 210 rows.



```
df.shape
```

```
(210, 7)
```

Description of the 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

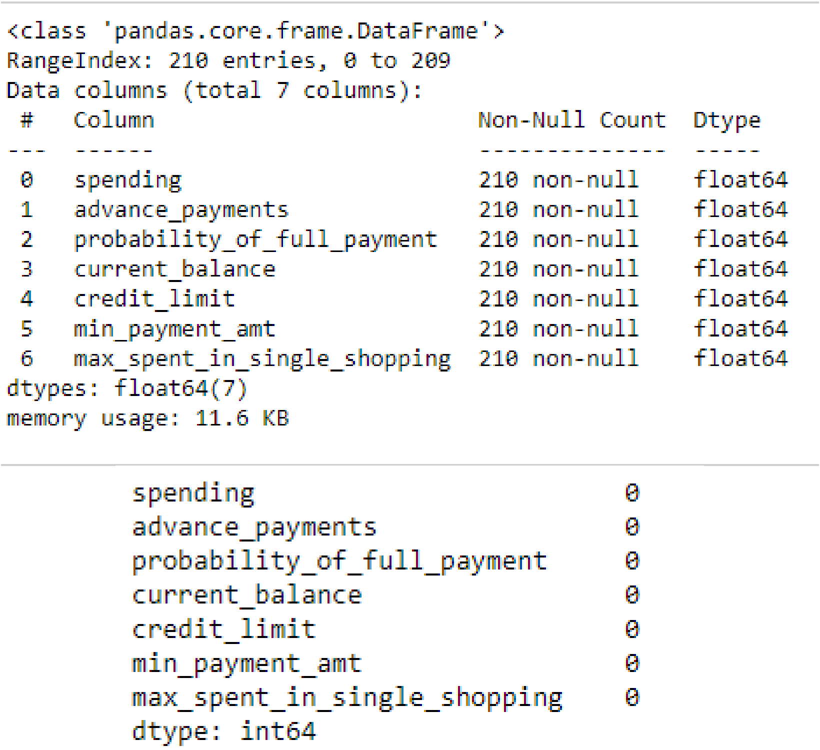
(Tab: 2 Describe)

We have 7 variables,

- No null values present in any variables.
- The mean and median values seem to be almost equal.
- The standard deviation for spending is high when compared to other variables.
- No duplicates in the dataset

All the columns are in float64 data type.

- As we can see in the information below, all the columns have 210 records of non-null values.



```
<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
```

```
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
```

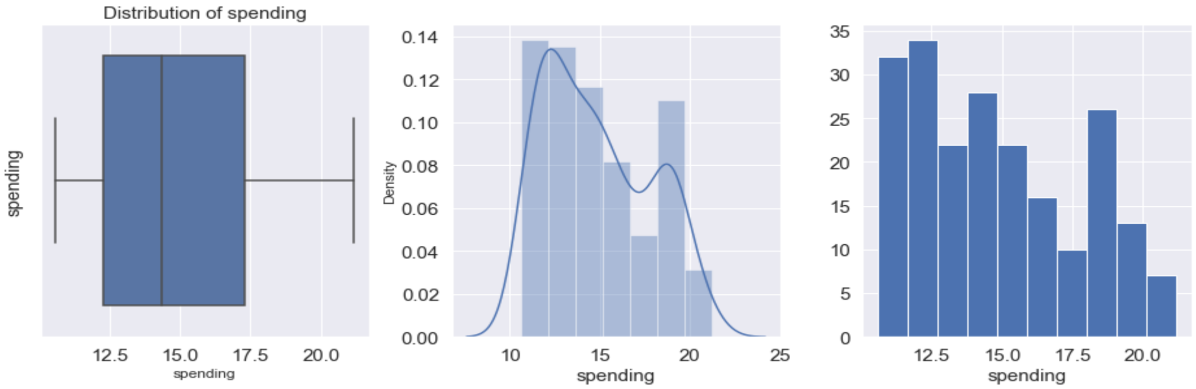
(Fig: 2 info Out nput)

There is no null values

Exploratory Data Analysis (Univariate / Bivariate analysis)

Helps us to understand the distribution of data in the dataset. With univariate analysis we can find patterns and we can summarize the data and have understanding about the data to solve our business problem.

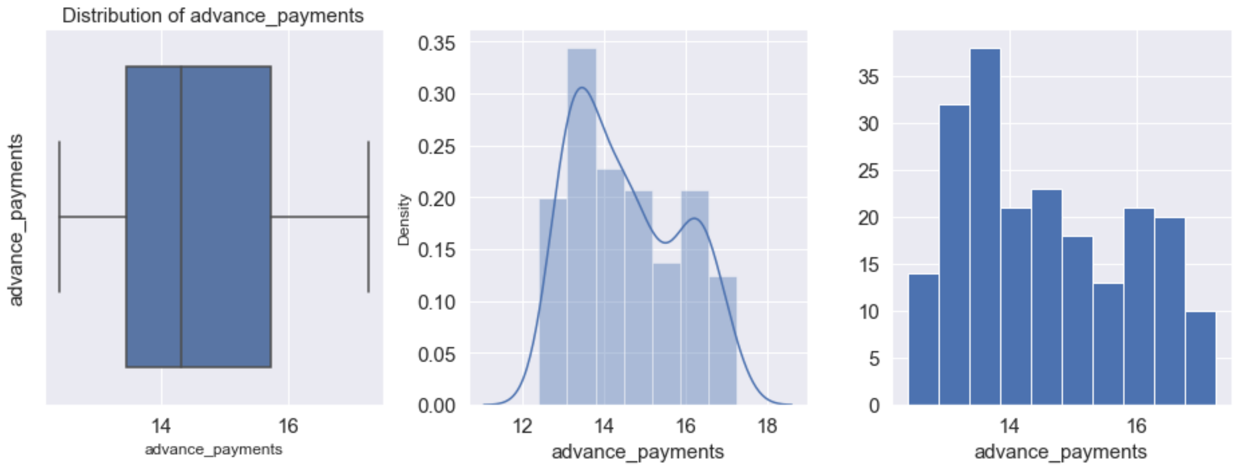
- The plots provide information about the distribution of the observations in the single data variable.
- Here we consider the distplot and violinplot (from sns package) to know the distribution of the individual variables from the dataset.
- The right skewed (or positively skewed) distribution has large occurrence in the left side and few in the right side. Here, mean is greater than the median.
- The left skewed (or negatively skewed) distribution has large occurrence in the right side and few in the left side. Here, mean is less than the median
- The symmetric distribution is the bell-shaped or normal distribution.
- Here, the mean is slightly greater than the median. All the variables are Right Skewed (Positively Skewed) distribution.



(Fig: 3 Spending)

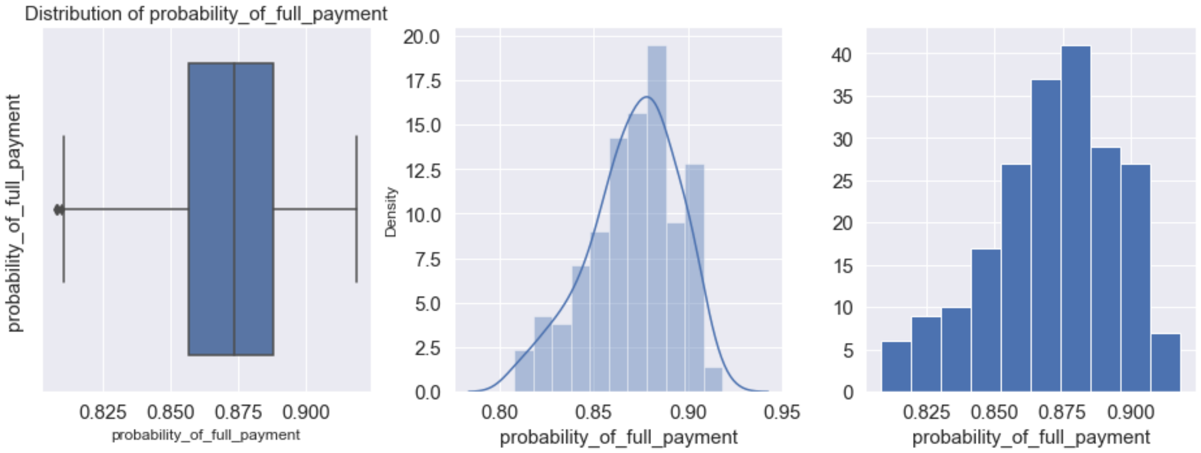
The box plot of the spending variable shows no outliers.

Spending is positively skewed - 0.399889.
We could also understand there could be chance of multi modes in the dataset. The dist. plot shows the distribution of data from 10 to 22



(Fig: 4 advance payment)

The box plot of the advance payment’s variable shows no outliers.
Advance payments are positively skewed - 0.386573.
We could also understand there could be chance of multi modes in the dataset. The dist. plot shows the distribution of data from 12 to 17

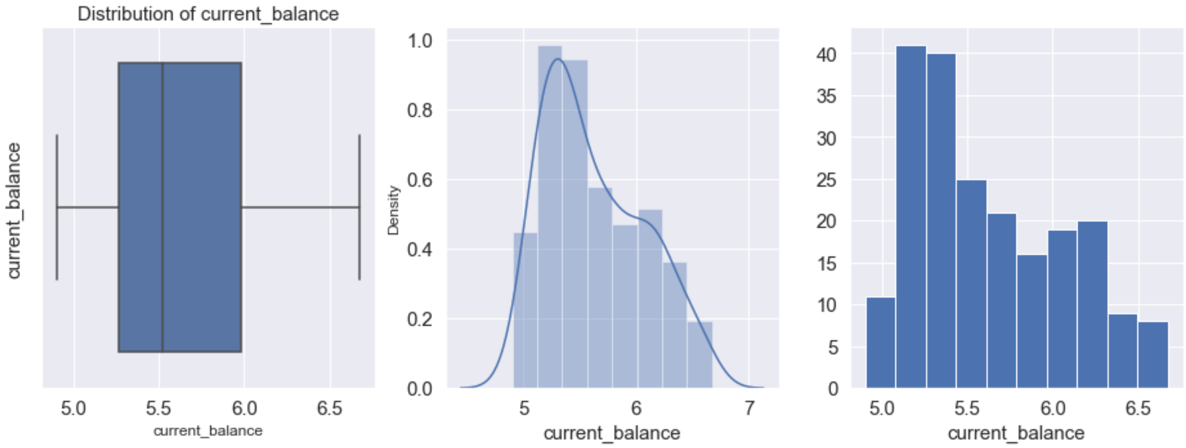


(Fig: 5 probability of full payment)

The box plot of the probability of full payment variable shows few outliers.

Probability of full payment is negatively skewed -0.537954

The dist. plot shows the distribution of data from 0.80 to
0.92. The Probability values is good above 80%

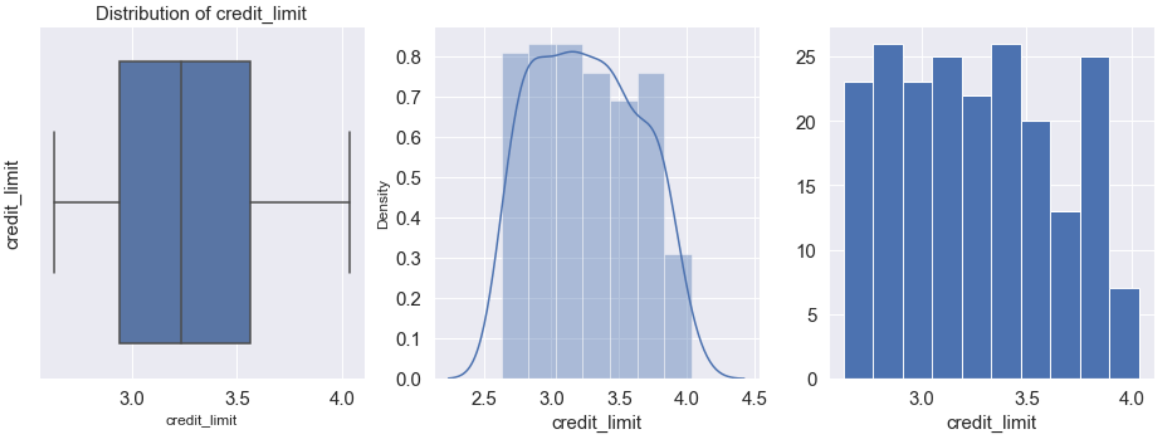


(Fig: 6 current balance)

The box plot of the current balance variable shows no outliers.

Current balance is positively skewed - 0.525482

The dist. plot shows the distribution of data from 5.0 to 6.5

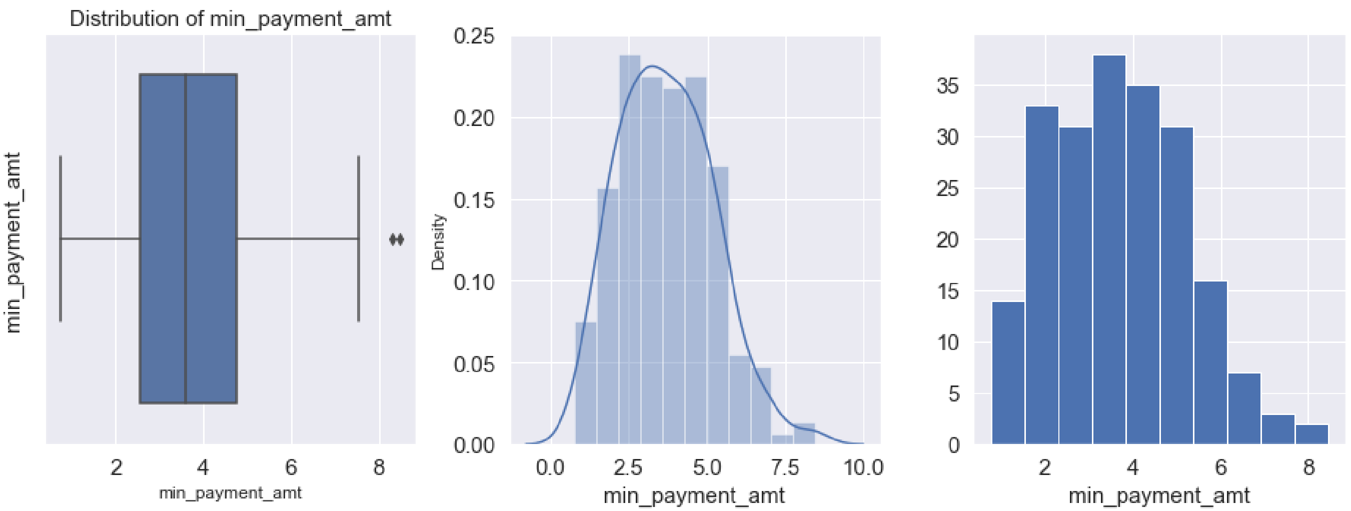


(Fig: 7 credit limit)

The box plot of the credit limit variable shows no outliers.

Credit limit is positively skewed - 0.134378

The dist. plot shows the distribution of data from 2.5 to 4.0

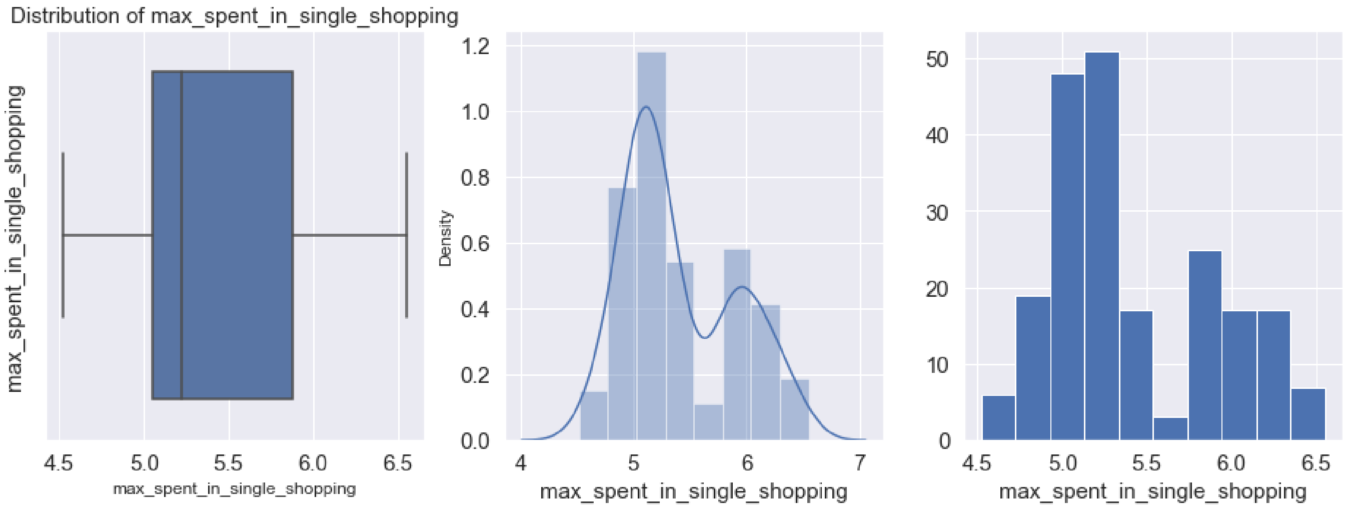


(Fig: 8 min spent)

The box plot of the min payment amount variable shows few outliers.

Min payment amount is positively skewed - 0.401667

The dist. plot shows the distribution of data from 2 to 8



(Fig: 9 max spent)

The box plot of the max spent in single shopping variable shows no outliers.

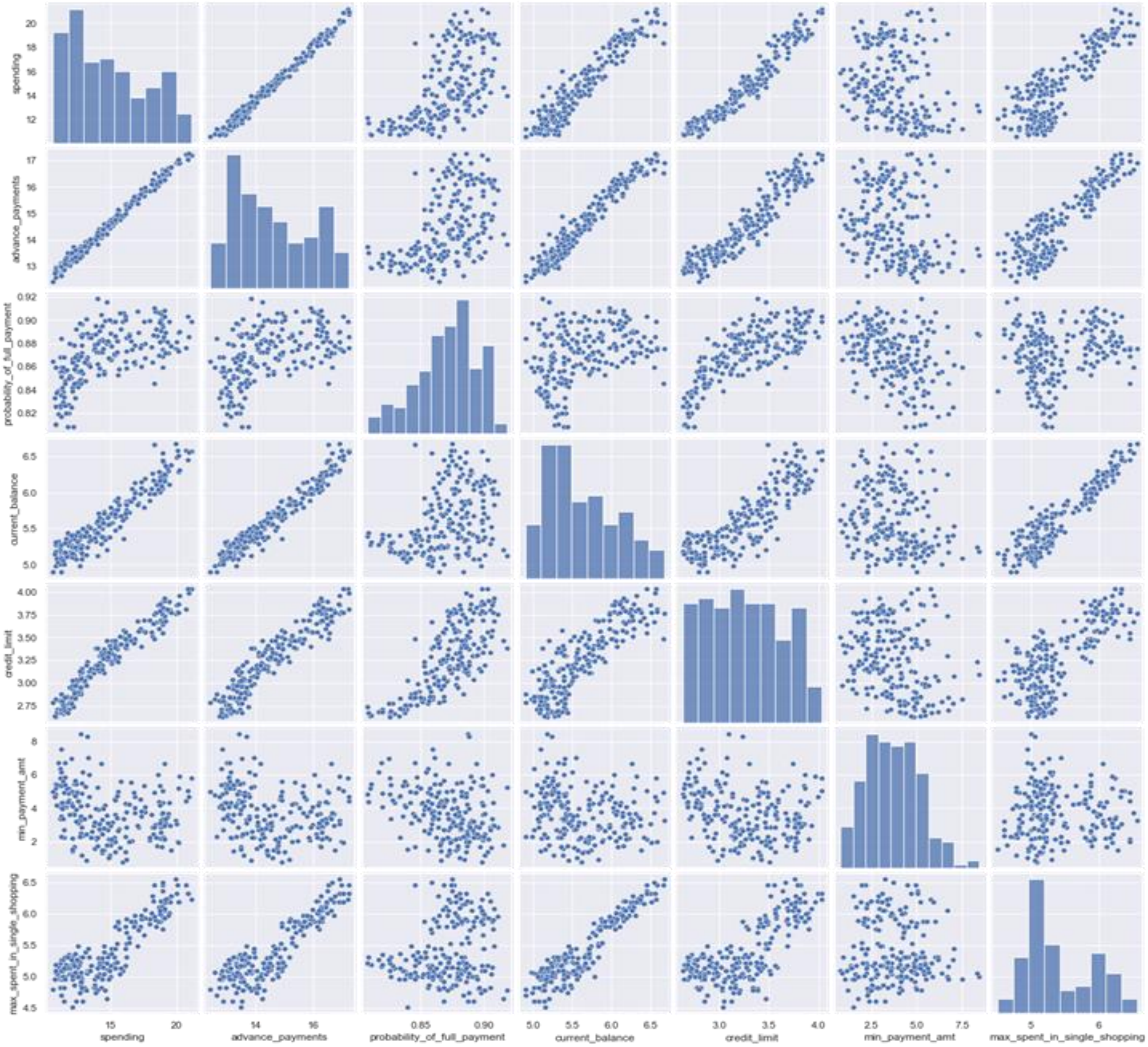
Max spent in single shopping is positively skewed - 0.561897

The dist. plot shows the distribution of data from 4.5 to 6.5

No outlier treatment – only 3 to 4 values re observed has outlier we are treating them

Exploratory Data Analysis (Multivariate)

Check for multicollinearity

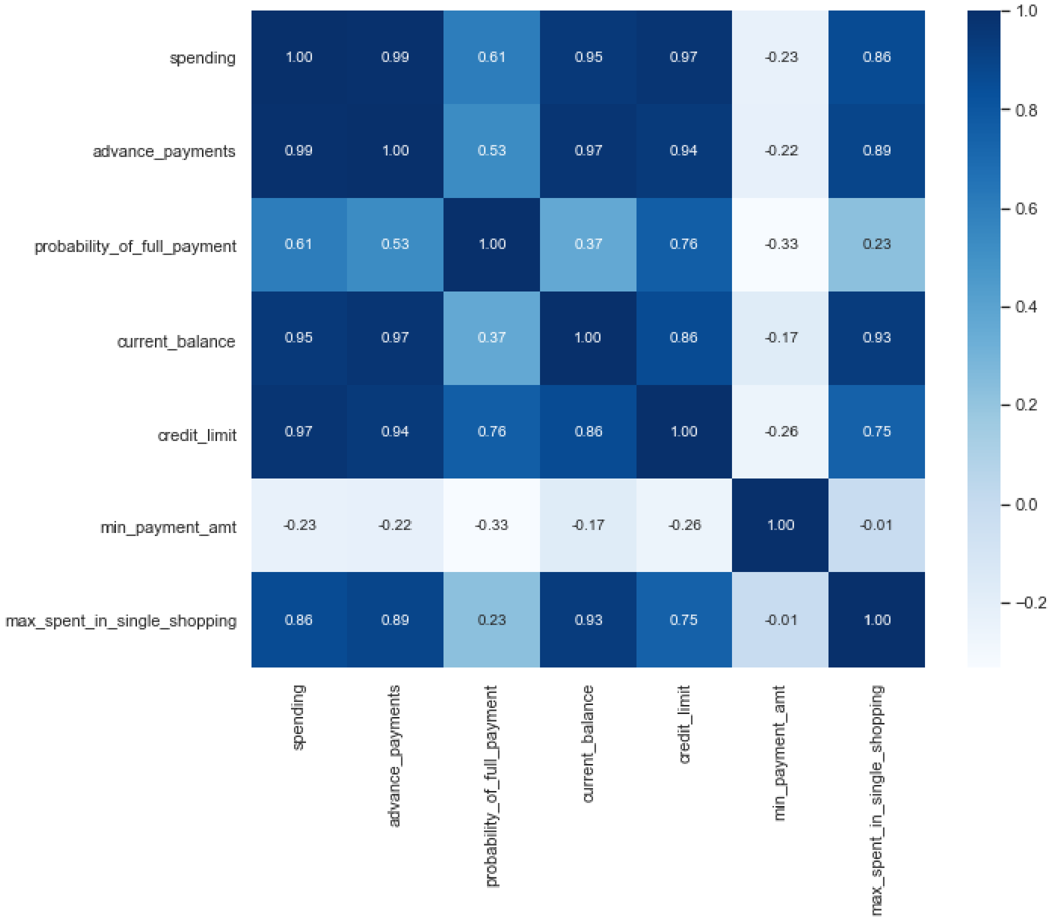


(Fig: 10 Multivariate)

- The correlation across the variables can be found using corr() function in a matrix form.
- To indicate and visualize the clusters within the data using the heatmap (from sns package).
- There is good correlation (0.99) between spending and advance_payments. The customers paid advance amount nearly equal to their spending amount.
- Next good correlation (0.97) is between advance_payments and current_balance. If the customers had no spending for that month, the advance amount is stored as current balance in his account.
- The next good correlation (0.95) is between spending and current_balance. The customer plans for the spending based on the current balance amount available.

- The min_payment_amt is very less correlated with all other variables. As, only few customers have paid minimum amount while making payments for purchases made monthly.
- We have to look carefully into min_payment_amt variable , as it will be the major deciding factor whether to provide promotional offers to the customers or not.
- The pairplot (from sns package) is used for visualizing the pair wise relationship across entire dataframe.
- There is a linear relationship between almost all the variables, mainly between spending and advance_payments.
- There is sparse relation between min_payment_amt and all other variables.

HEAT MAP Analysis



(Fig: 11 Heat Map)

Heat map for Better Visualization

- 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

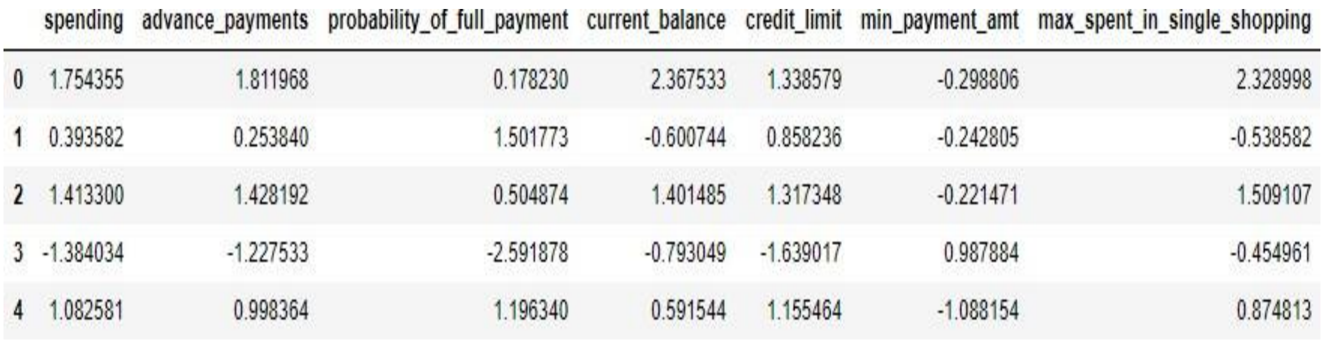
1.2 Do you think scaling is necessary for clustering in this case? Justify

Yes, scaling is very important as the model works based on the distance-based computations scaling is necessary for unscaled data.

Scaling needs to be done as the values of the variables are in different scales. Spending, advance payments are in different values and this may get more weightage. Scaling will have all the values in the relative same range.

I have used standard scalar for scaling

Below is the snapshot of scaled 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

(Tab: 3 Scaled data)

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

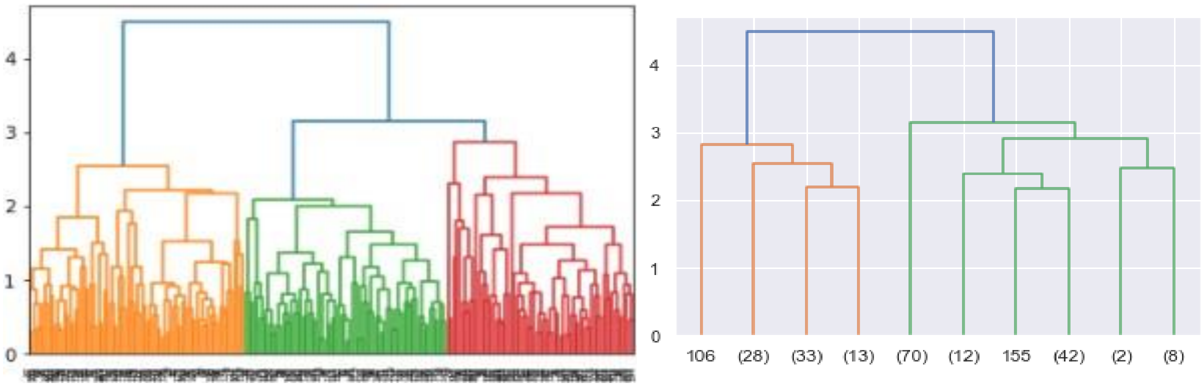
In Hierarchical clustering, the records are sequentially grouped to create clusters, based on the distance between the records and distance between the clusters.

- The 2 types of Hierarchical clustering are,
  - Agglomerative Clustering
  - Divisive Clustering
- The similar records can be grouped based on the various Distance measures:
  - Euclidean Distance
  - Manhattan Distance
  - Minkowski Distance
  - Chebyshev Distance
  - Hamming Distance
- The similar clusters can be grouped based on the following Linkage types:
  - Single Linkage
  - Complete Linkage
  - Average Linkage
  - Centroid Linkage
  - Wards Linkage
- Using the Dendogram, we can produce the graphical display of clustering process and decide a suitable cut-off.
- Here, we apply the Agglomerative clustering.

AVERAGE LINKAGE METHOD:

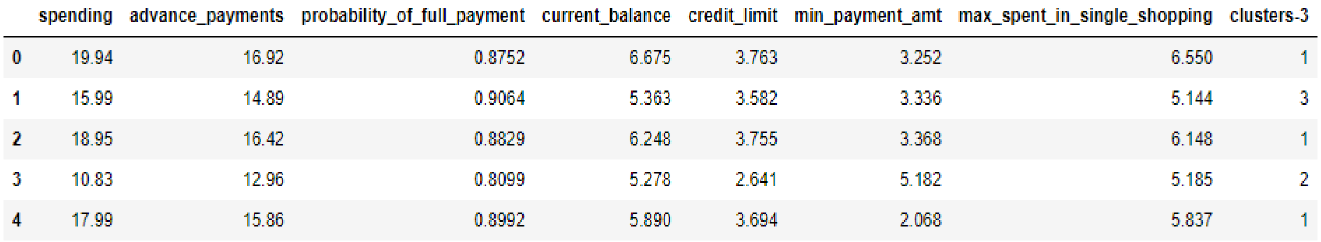
USING DENDOGRAM:

- With the help of scipy.cluster.hierarchy package, we use dendogram module.



(Fig: 12 Average Linkage method)

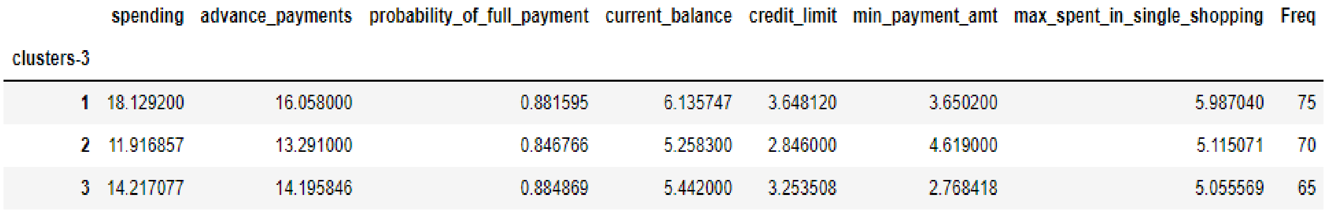
- We get the above Dendogram, using the **Average Linkage method** on the scaled dataset.
- We can make the cut-off at Y-axis between 2-3 units, to get 3 clusters.



	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

(Tab: 4 Cluster Frequency head)

- By adding the clusters to our original dataset, we can see that every record falls under a cluster.
- We can group the data by clusters and calculate the frequency of the clusters to understand the pattern of the grouped clusters.
- The similar records can be grouped based on the various Distance measures:
  - Euclidean Distance
  - Manhattan Distance
  - Minkowski Distance
  - Chebyshev Distance
  - Hamming Distance
- The similar clusters can be grouped based on the following Linkage types:
  - Single Linkage
  - Complete Linkage
  - Average Linkage
  - Centroid Linkage
  - Wards Linkage
- Using the Dendogram, we can produce the graphical display of clustering process and decide a suitable cut-off.
- Here, we apply the Agglomerative clustering.



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
clusters-3								
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

(Tab: 5 Cluster3 data)

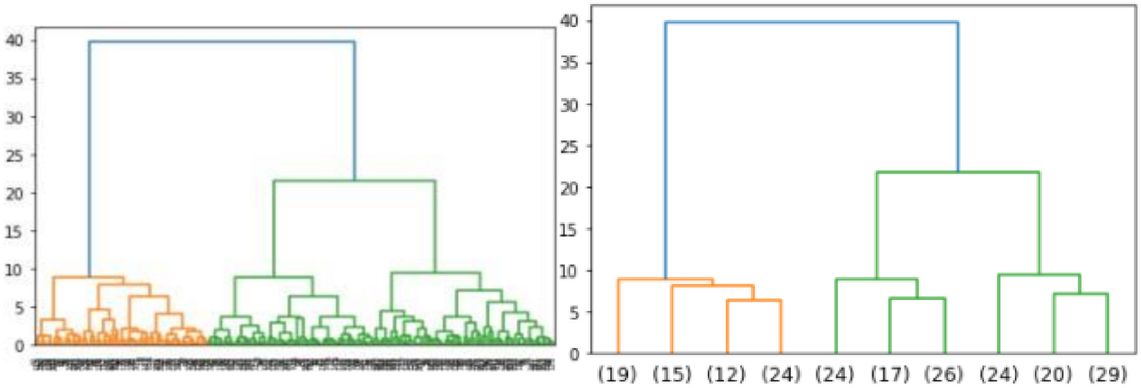
- The Frequencies seems closer and not much significant with each other.

OBSERVATION:

Both the method are almost similar means, minor variation, which we know it occurs. There was not too much variations from both methods 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 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).

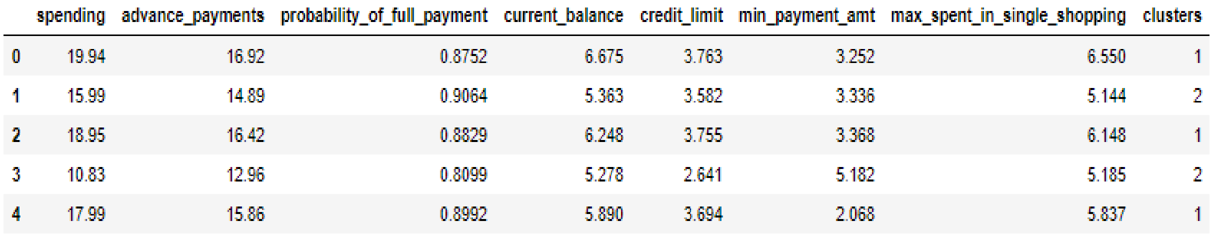
WARD LINKAGE METHOD:

- We try using the Wards Linkage method on the scaled dataset to get the dendogram.



(Fig: 13 Ward method)

- The combinations of 2 lines are not joined on the Y-axis from 20 to 40, for about 20 units.
- So, the optimal number of clusters will be 2 for hierarchical clustering.

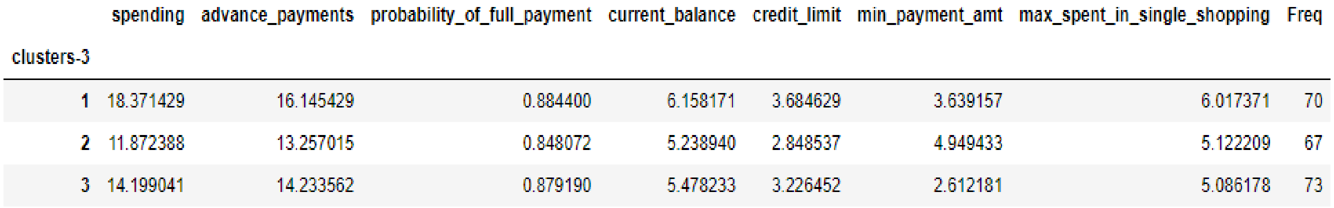


	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	clusters
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	1
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	2
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1
3	10.83	12.96	0.8099	5.278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

(Tab: 6 Ward cluster data)

- Adding the clusters as a column to the original data, every record is either in cluster 1 or cluster 2.
- By grouping the clusters according to their frequency, to understand the pattern of grouped records in each cluster.

CLUSTER PROFILES:



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

(Tab: 7 Cluster Profiling)

- From the above cluster profiling, we can see the customers are grouped based on the patterns of the credit card usage as 2 clusters.
- The Inference we get is:
    - Cluster 1 : Moderate Credit Card Usage
    - Cluster 2 : High Credit Card Usage

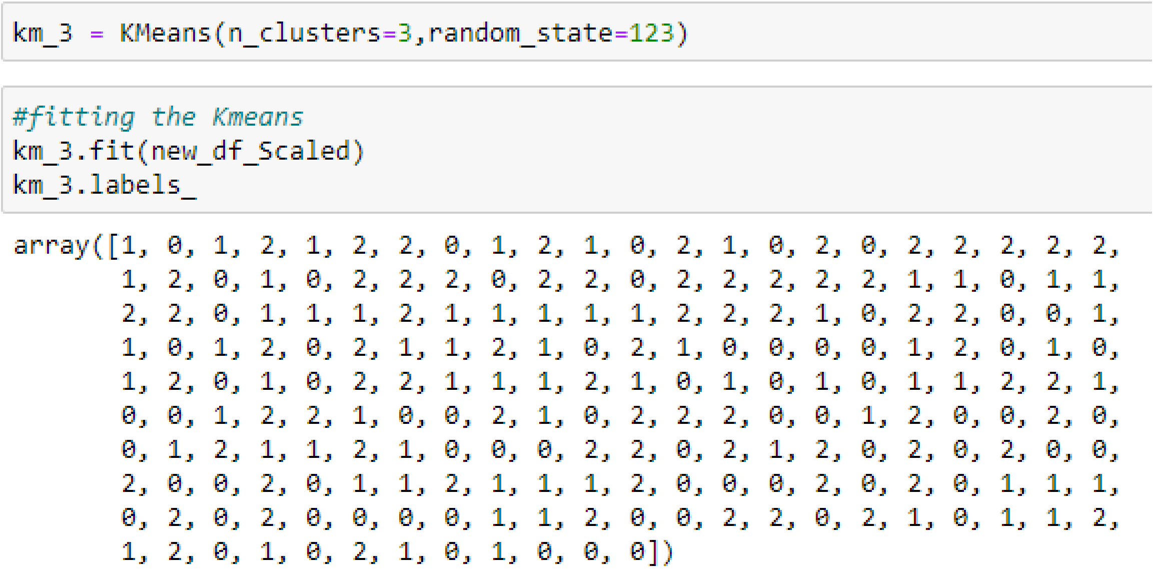
1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score.

K-means clustering,

Randomly we decide to give n_clusters = 3 and we look at the distribution of clusters according to the n_clusters.

We apply K-means technique to the scaled data.

Cluster output for all the observations in the dataset,



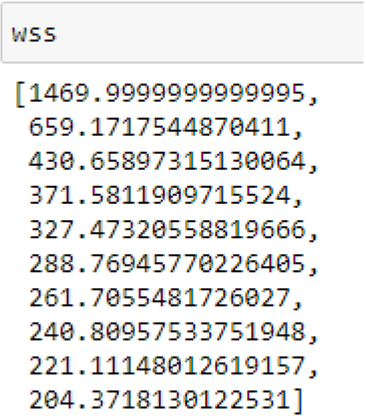
```
km_3 = KMeans(n_clusters=3,random_state=123)
```

```
#fitting the Kmeans
km_3.fit(new_df_Scaled)
km_3.labels_
```

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

(Fig: 14 Cluster Output)

- We have 3 clusters 0, 1, 2
- To find the optimal number of clusters, we can use k-elbow method

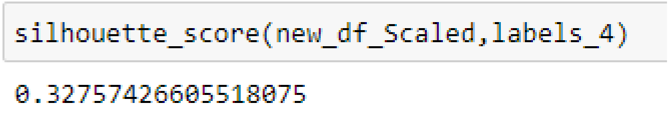


```
wss
```

```
[1469.9999999999995,
 659.1717544870411,
 430.65897315130064,
 371.5811909715524,
 327.47320558819666,
 288.76945770226405,
 261.7055481726027,
 240.80957533751948,
 221.11148012619157,
 204.3718130122531]
```

(Fig: 15 Wss Output)

- To find the inertia value for all the clusters from 1 to 11, I used for loop to find the optimal number of clusters.
- The silhouette score for 4 clusters is good

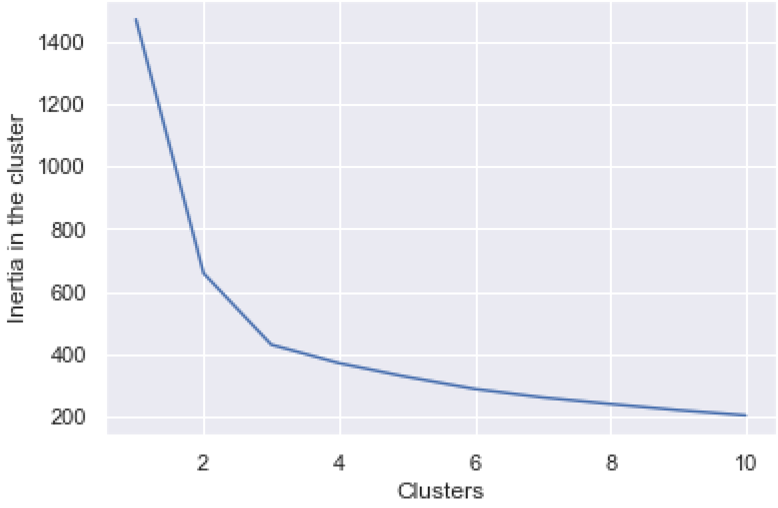


```
silhouette_score(new_df_Scaled,labels_4)
```

```
0.32757426605518075
```

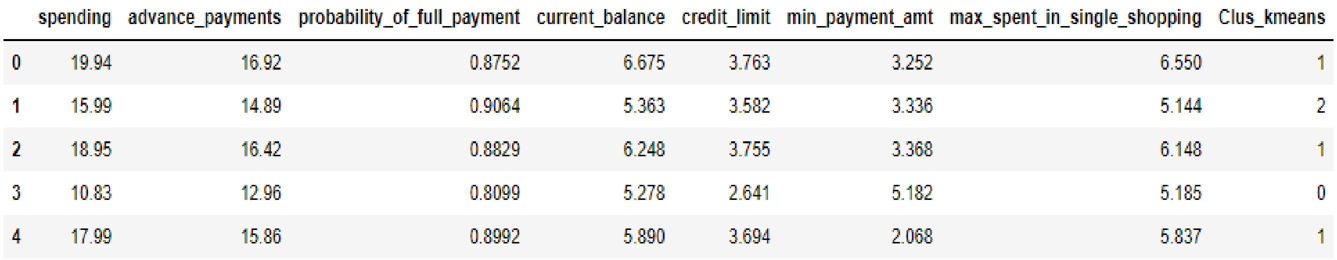
(Fig: 16 Silhouette Score Output)

- The elbow curve seen here also shows us after 3 clusters there is no huge drop in the values, so we select 3 clusters.



(Fig: 17 Cluster Map)

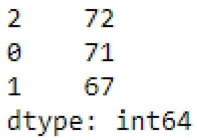
- So adding the cluster results to our dataset to solve our business objective.



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

(Tab: 8 silhouette width score)

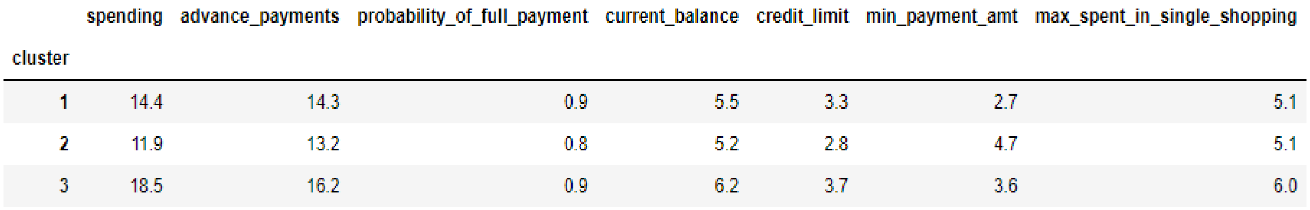
- This table shows the clusters to the dataset and also individual silhouette width score. Cluster frequency



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

K-Means Clustering & Cluster Information

This frequency shows frequency of clusters to the dataset.



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

(Tab: 9 Frequency data)

3-Group clusters via K- Means has equal split of percentage of results.

Cluster 0 – Medium
Cluster 1 – low
Cluster 2 – High

Observation

By K- Mean’s method we can at cluster 3 we find it optimal after there is no huge drop in inertia values. Also, the elbow curve seems to show similar results.
The silhouette width score of the K – means also seems to very less value that indicates all the data points are properly clustered to the cluster. There is no mismatch in the data points with regards to clustering

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.

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)**.**

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

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 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)

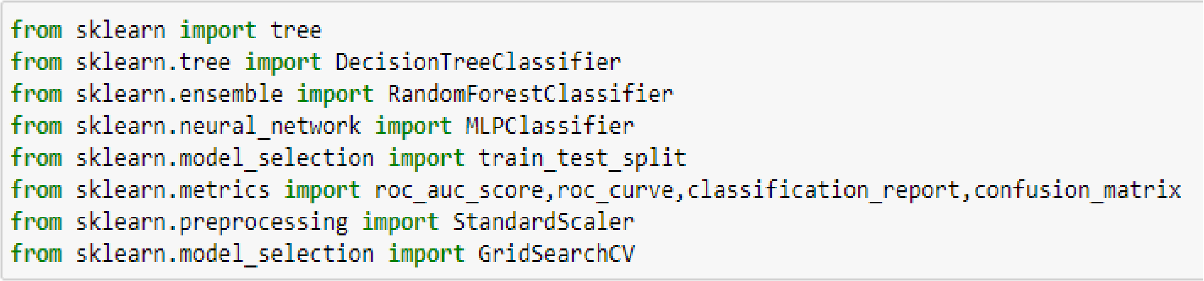
Group 3: Medium Spending Group

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.

PROBLEAM 2: CART-RF-ANN

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

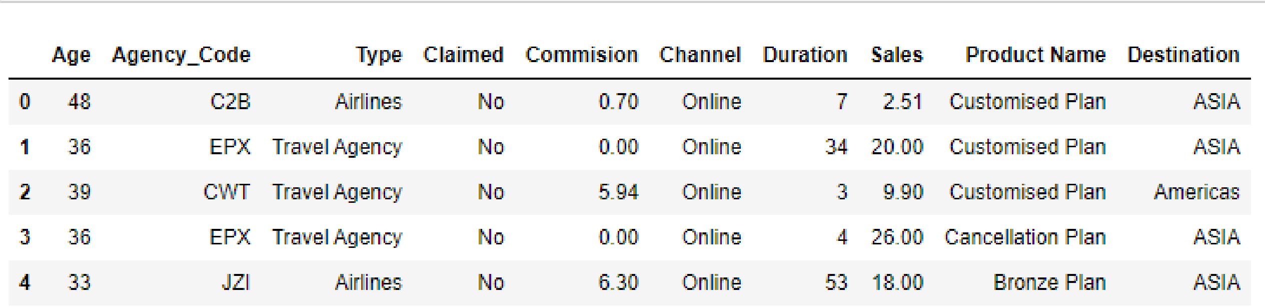
IMPORT LIBRARY



```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score,roc_curve,classification_report,confusion_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV
```

(Fig: 18 Import 2)

Reading the dataset,



	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	C2B	Airlines	No	0.70	Online	7	2.51	Customised Plan	ASIA
1	36	EPX	Travel Agency	No	0.00	Online	34	20.00	Customised Plan	ASIA
2	39	CWT	Travel Agency	No	5.94	Online	3	9.90	Customised Plan	Americas
3	36	EPX	Travel Agency	No	0.00	Online	4	26.00	Cancellation Plan	ASIA
4	33	JZI	Airlines	No	6.30	Online	53	18.00	Bronze Plan	ASIA

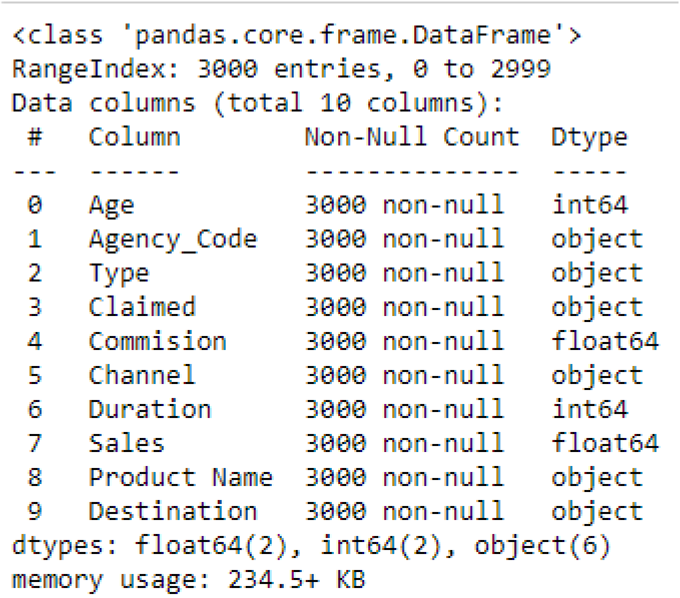
(Tab: 10 Data Head)

The data has read successfully,

The shape of the dataset is (3000, 10)

Info function clearly indicates the dataset has object, integer and float so we have to change the object data type to numeric value.

INFORMATION



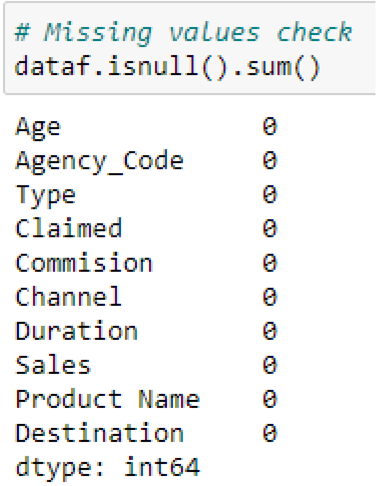
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Age           3000 non-null   int64  
 1   Agency_Code   3000 non-null   object 
 2   Type        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

(Fig: 19 Info output)

Observation

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

Missing value Check



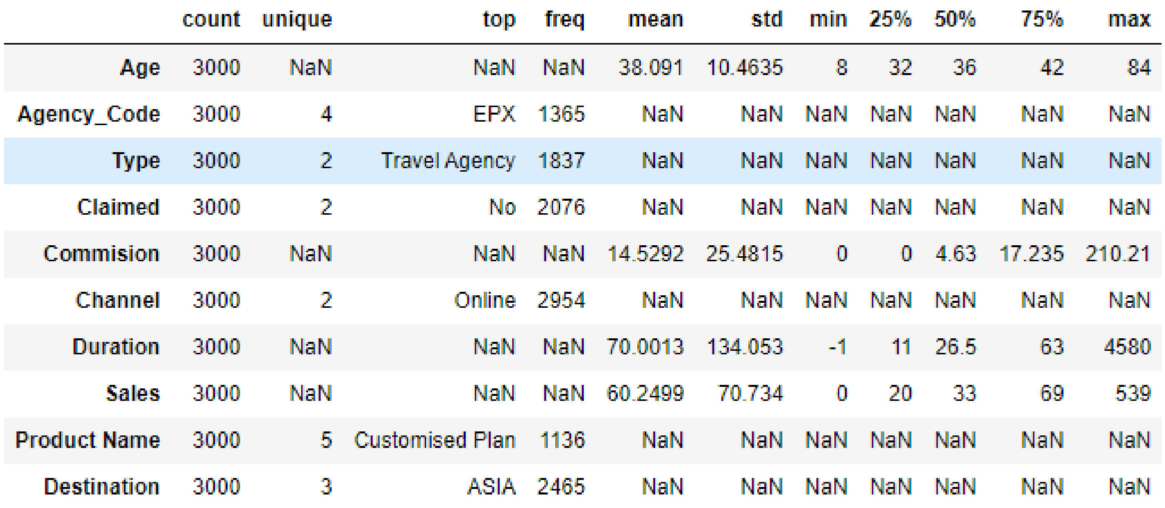
```
# Missing values check
dataf.isnull().sum()
```

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

(Fig: 20 Missing value)

No missing value available in dataset.

Describe Data,



	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

(Tab: 11 Describe Data)

We have 4 numeric values and 6 categorical values, Agency code EPX has a frequency 1365,

The most preferred type seems to be travel agency Channel is online, customized plan is the most sought plan by customers

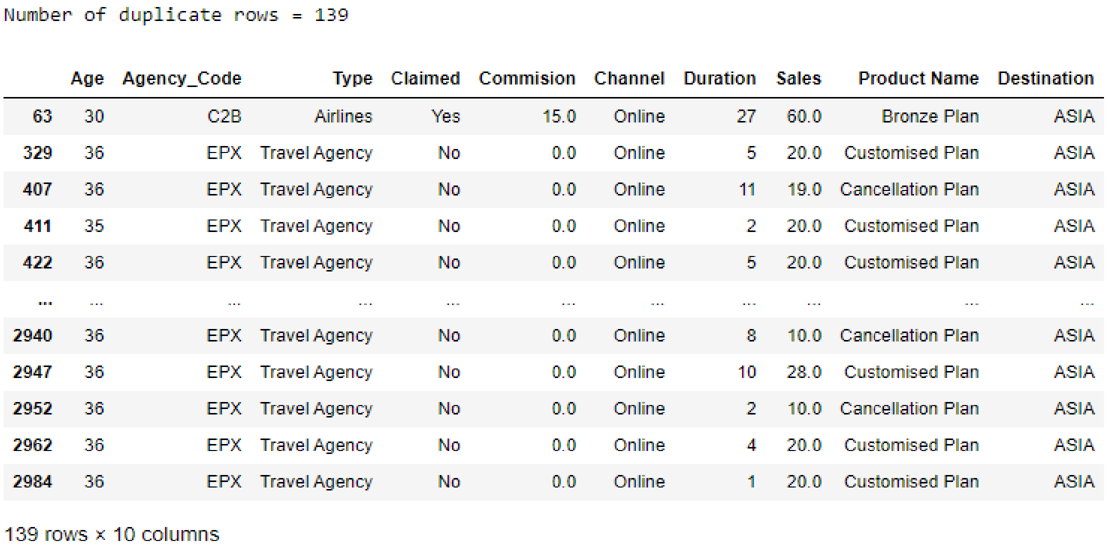
Destination ASIA seems to be most sought destination place by customers.

We will further look at the distribution of dataset in univariate and bivariate analysis checking for duplicates in the dataset,

Observation

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

Check duplicate data,



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

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

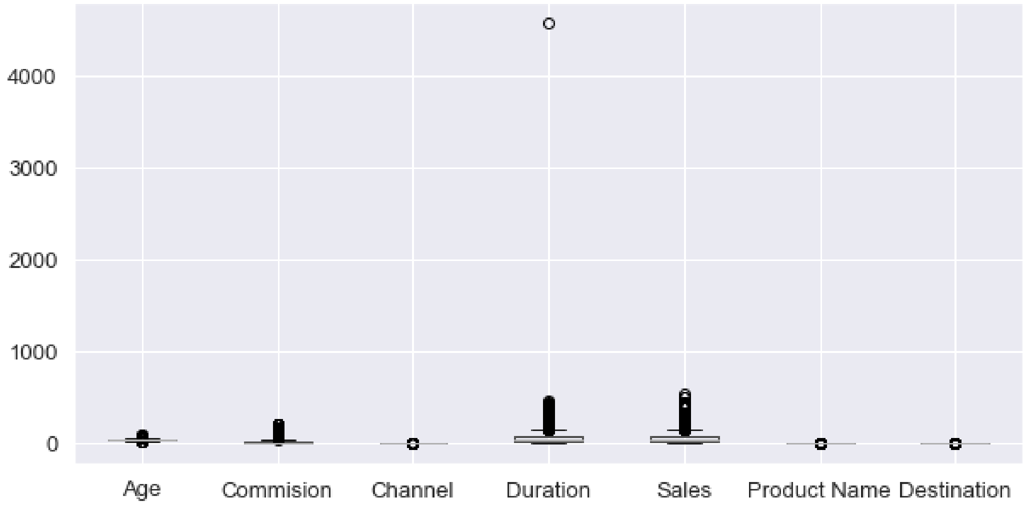
139 rows × 10 columns

(Tab: 12 Duplicate list)

Observation

- 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.

Checking Out layers



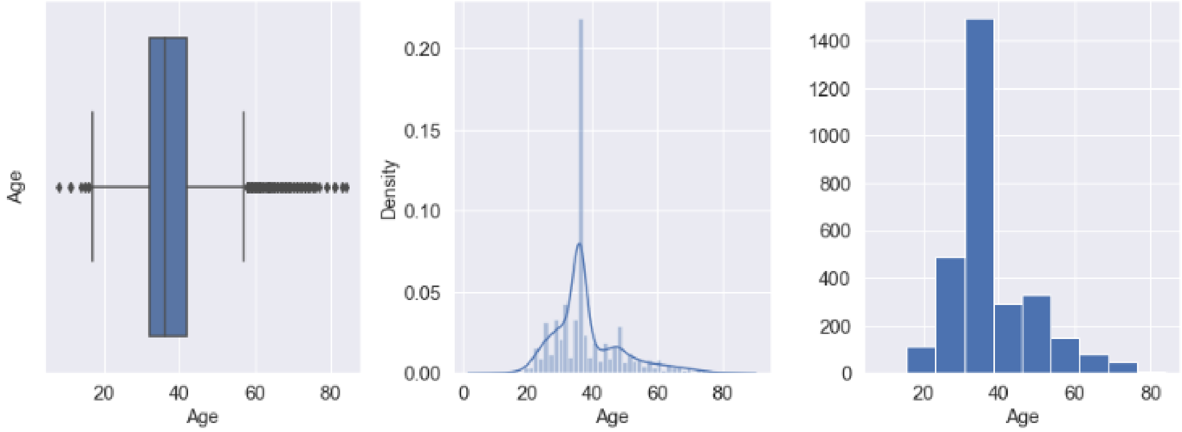
(Fig: 21 out layers)

- Outliers exist in almost all the numeric values.
- We can treat outliers in random forest classification.

Let’s see unique value count of all nominal variables,

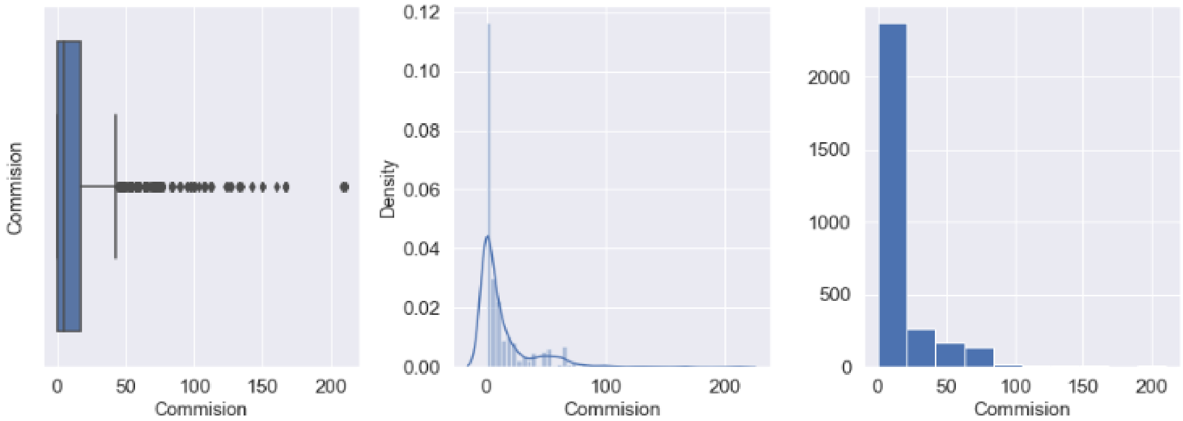
```
AGENCY_CODE :  4
JZI     239
CWT     472
C2B     924
EPX    1365
Name: Agency_Code, dtype: int64


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
```

Univariate / Bivariate analysis



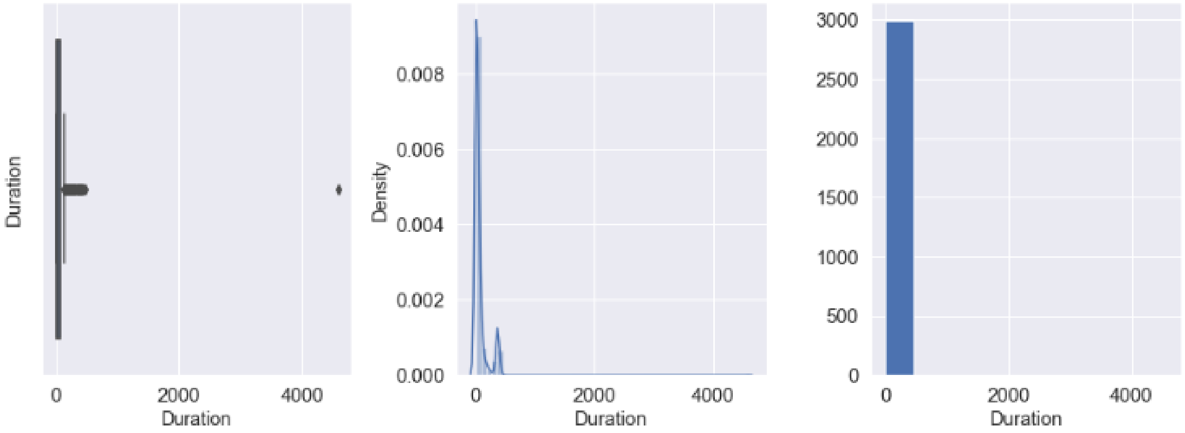
(Fig: 22 Age skewed)

- Spending is positively skewed - 1.149713
- The box plot of the Age variable shows outliers.
- The dist. plot shows the distribution of data from 20 to 80
- In the range of 30 to 40 is where the majority of the distribution lies.



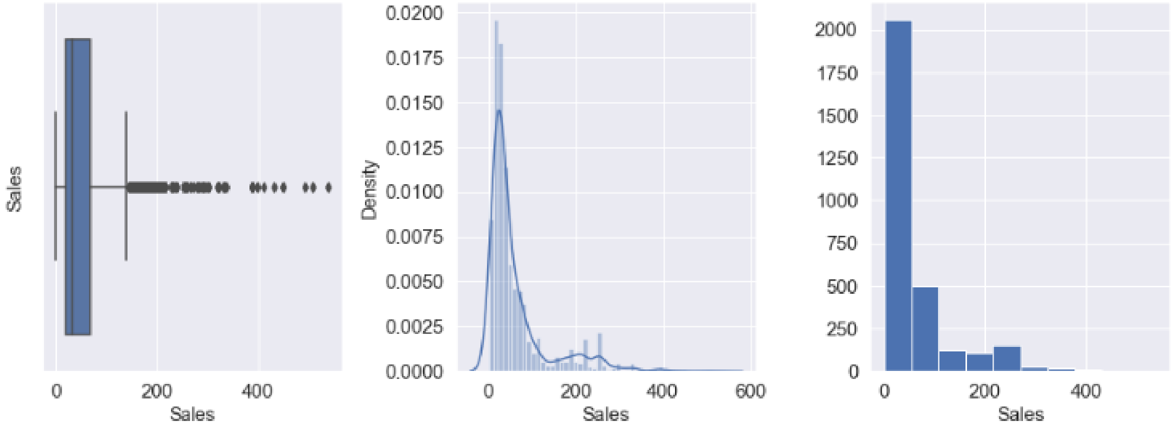
(Fig: 23 Commission skewed)

- The box plot of the Commission variable shows outliers.
- Spending is positively skewed - 3.148858
- The dist. plot shows the distribution of data from 0 to 30



(Fig: 24 Duration skewed)

- The box plot of the Duration shows outliers.
- Spending is positively skewed - 13.784681
- The dist. plot shows the distribution of data from 0 to 100

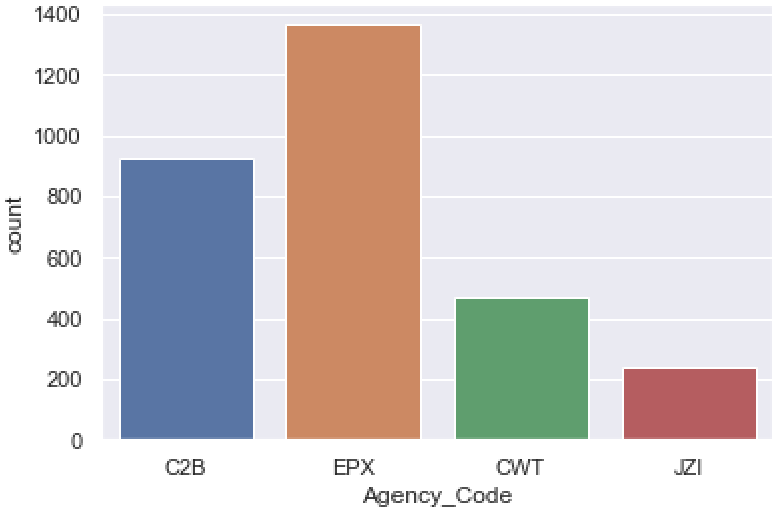


(Fig: 25 Sales skewed)

- The box plot of the Sales variable shows outliers.
- Spending is positively skewed - 2.381148
- The dist. plot shows the distribution of data from 0 to 300

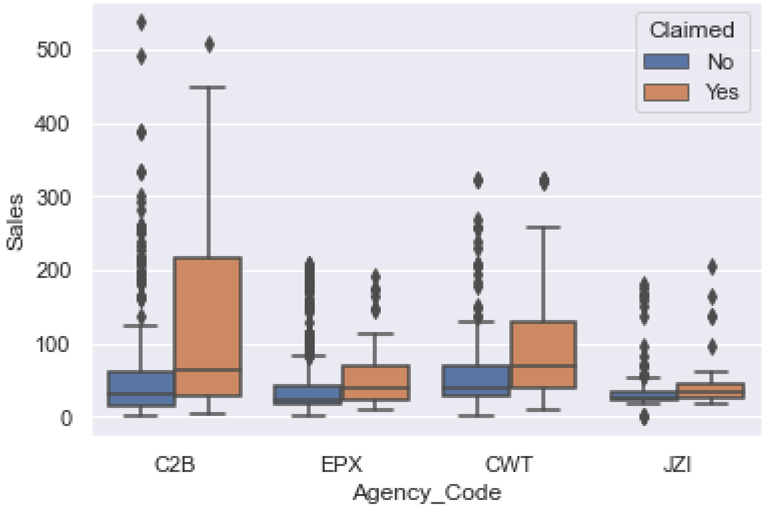
CATEGORICAL VARIABLES

Agency Code



(Fig: 26 Agency code count plot)

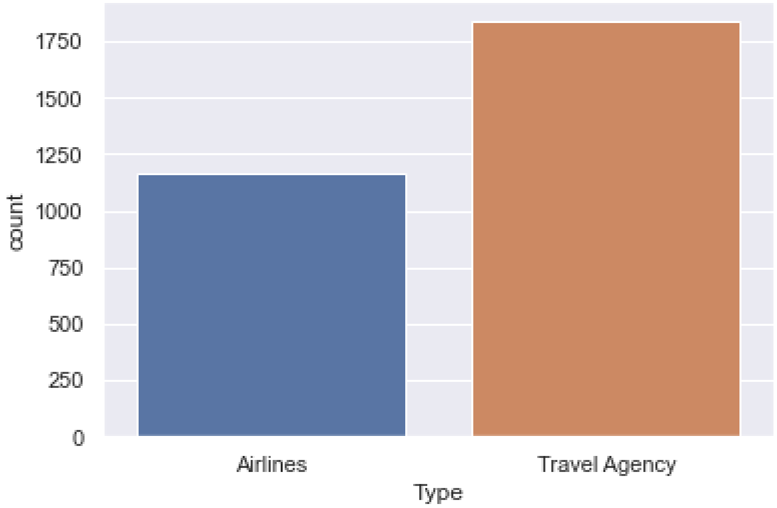
- The distribution of the agency code, shows us EPX with maximum frequency



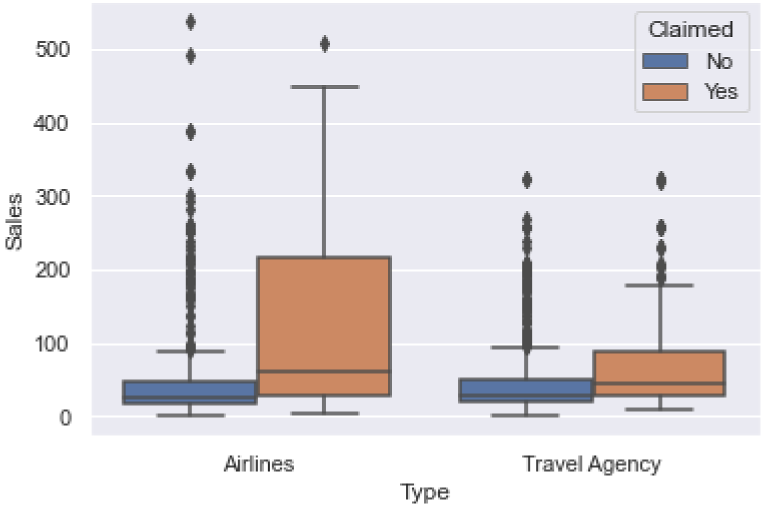
(Fig: 27 Agency code box plot)

- The box plot shows the split of sales with different agency code and also hue having claimed column.
- It seems that C2B have claimed more claims than other agency.

TYPE



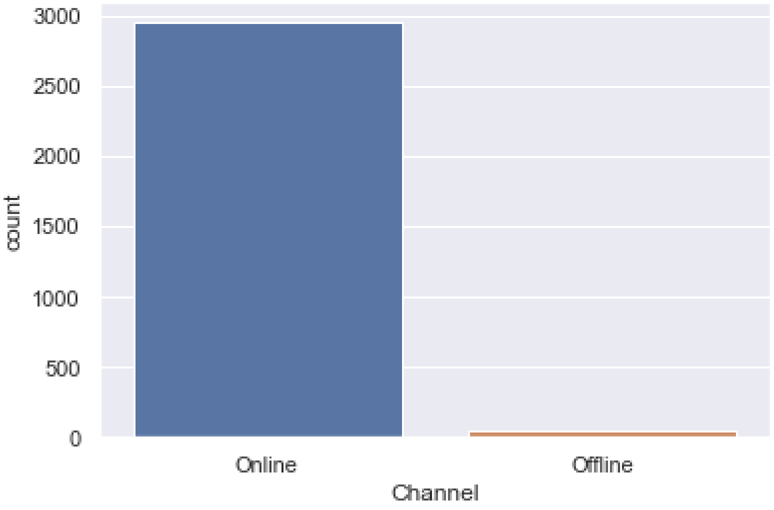
(Fig: 28 Type count plot)



(Fig: 29 Type box plot)

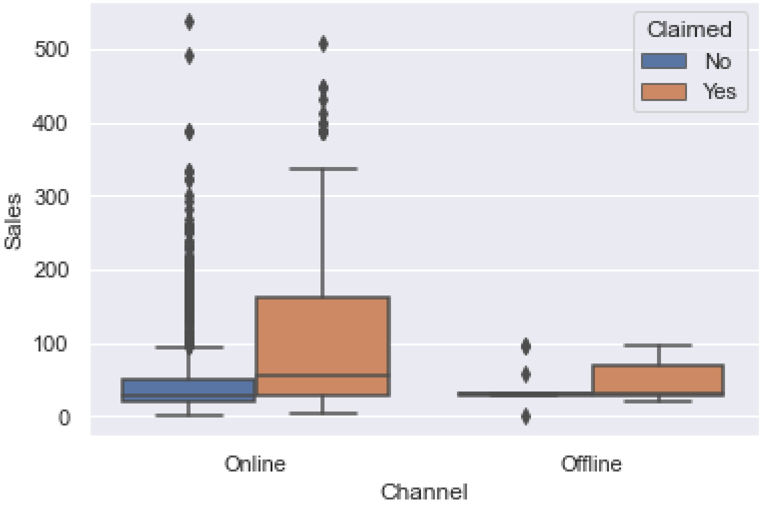
- The box plot shows the split of sales with different type and also hue having claimed column. We could understand airlines type has more claims.

CHANNEL



(Fig: 30 Channel count plot)

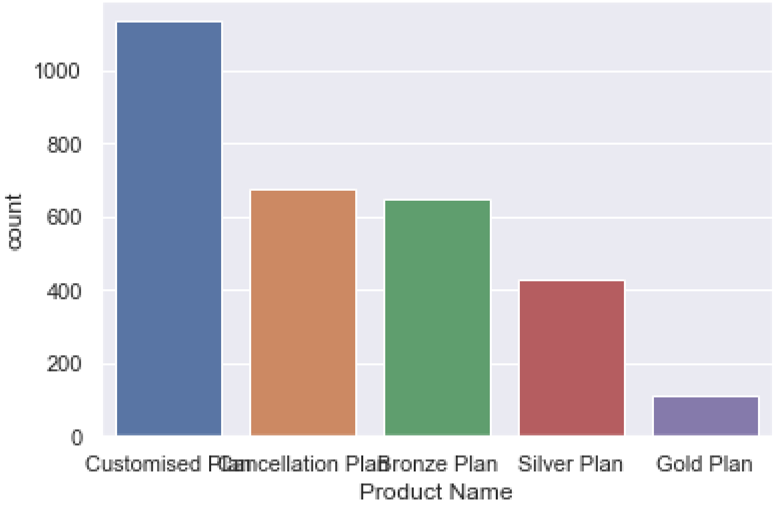
- The majority of customers have used online medium, very less with offline medium



(Fig: 31 Channel box plot)

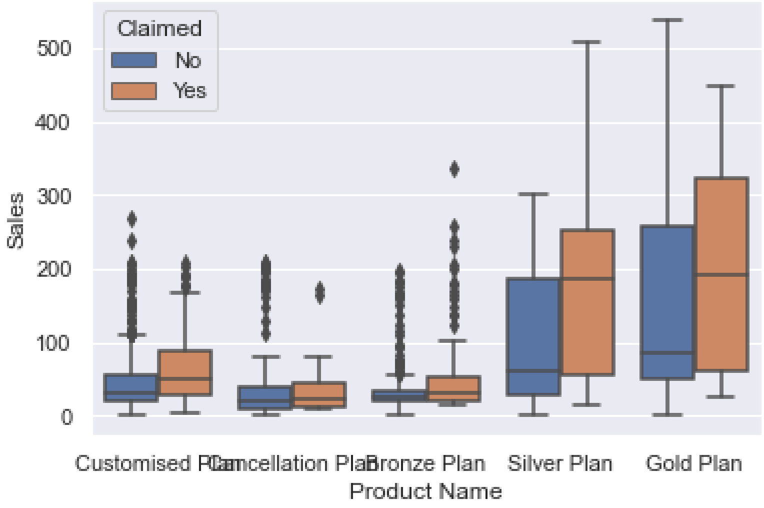
- The box plot shows the split of sales with different channel and also hue having claimed column.

PRODUCT NAME



(Fig: 32 Product name count plot)

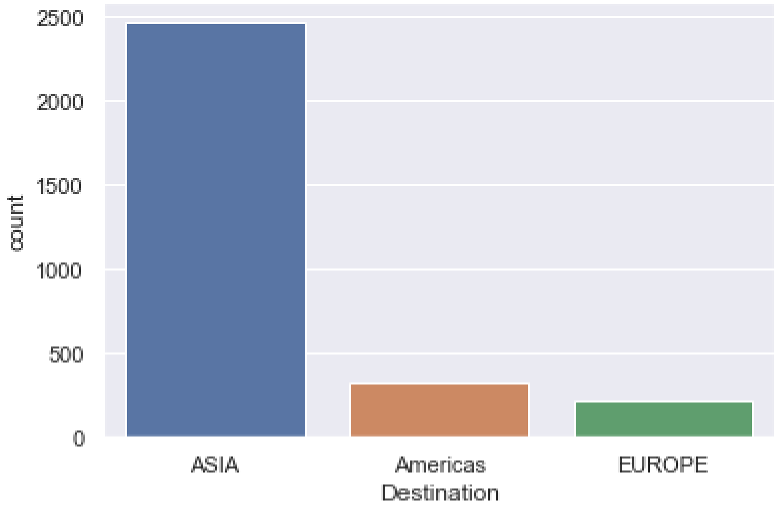
- Customized plan seems to be most liked plan by customers when compared to all other plans.



(Fig: 33 Product name box plot)

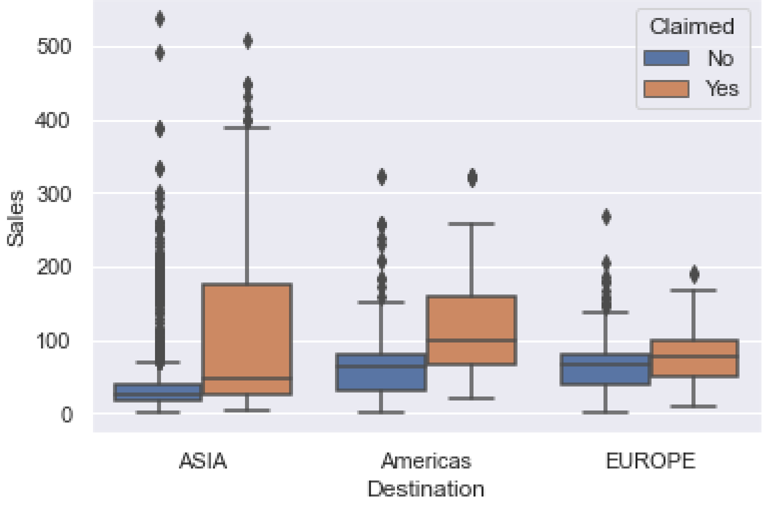
- The box plot shows the split of sales with different product name and also hue having claimed column.

DESTINATION



(Fig: 34 Destination count plot)

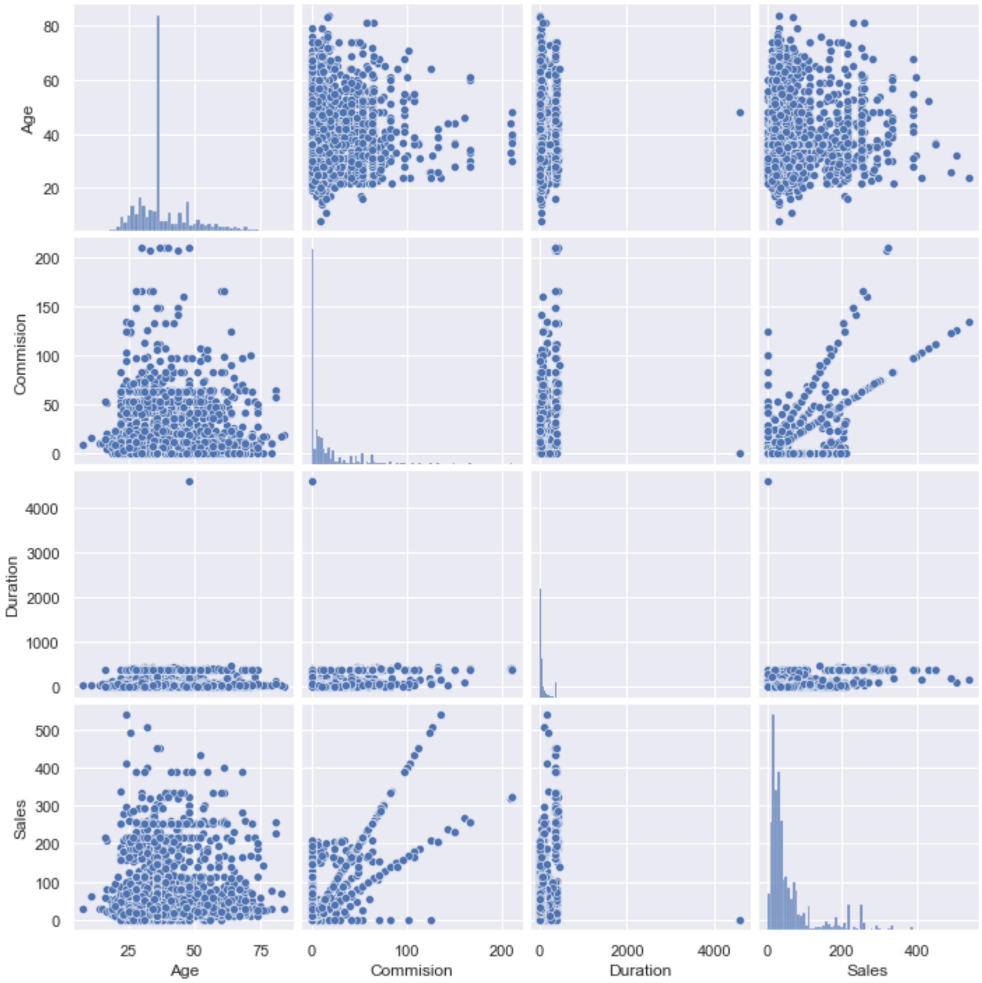
- Asia is where customers choose when compared with other destination places.



(Fig: 35 Destination box plot)

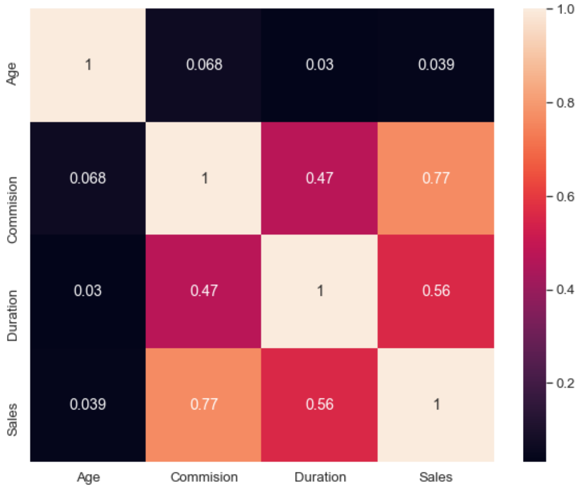
- The box plot shows the split of sales with different destination and also hue having claimed column.

CHECKING PAIRWISE DISTRIBUTION OF THE CONTINUOUS VARIABLES



(Fig: 36 Continuse variable plot)

CHECKING FOR CORRILATION



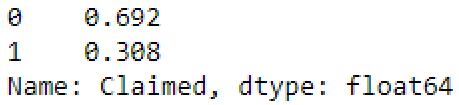
(Fig: 37 Continuse variable plot)

- Not much of multi collinearity observed no negative correlation only positive correlation.

Converting all objects to categorical codes

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


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


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


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


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


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

Proportion of 1s and 0s

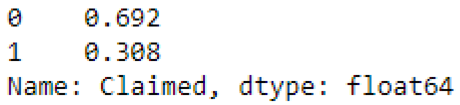


```
0    0.692
1    0.308
Name: Claimed, dtype: float64
```

- Checking the proportion of 1s and 2s in the dataset. That is our target column.

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

- To train the model, we split the entire data as a set of records and their labels in either of the Train Data or Test Data.
- Before splitting the data, we check the proportion of the target column, i.e., Claimed in our case.

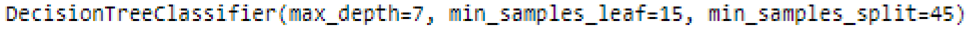


```
0    0.692
1    0.308
Name: Claimed, dtype: float64
```

- The target column is captured into separate vector for training set and testing set.
- From sklearn.model_selection package, we import the train_test_split module to divide the data as training set and testing set.
- We can build the CART on DecisionTreeClassifier module, Ranform Forest on RandomForestClassifier, Artificial Neural Networks on MLPClassifier.
- To get the best parameters for the model, we can use the GridSearchCV from sklearn.model_selection package.
- It uses cross-validation for the number of times to loop through the predefined hyperparameters and fit our model(CART,RF,ANN) on the training set.
- This can be used to evaluate the test set and compare its performance with the previous training set.

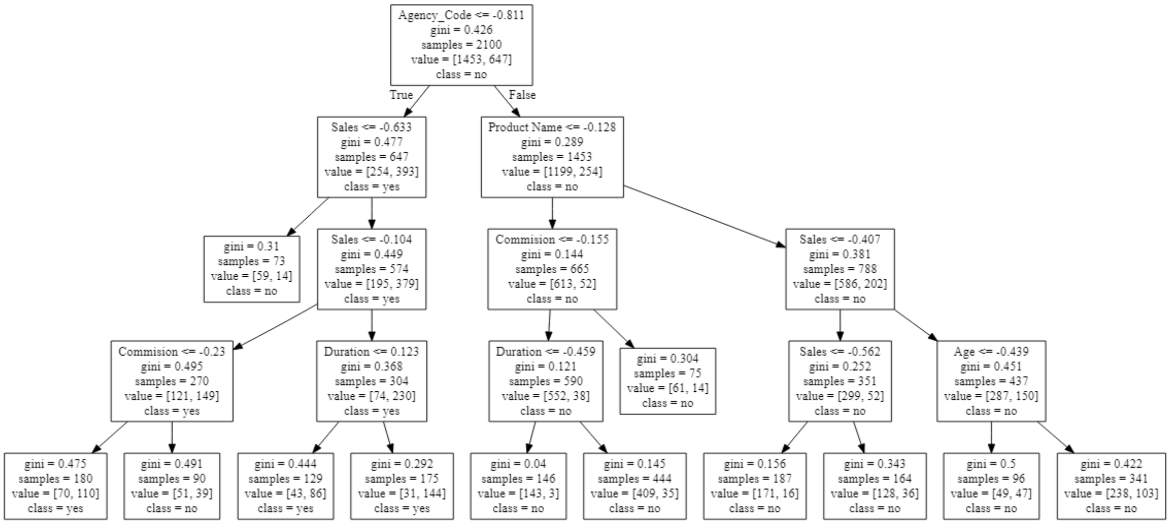
DECISION TREE CLASSIFIER (CART):

- Using the DecisionTreeClassifier module from sklearn.tree package, we create the decision tree models.
- To avoid the tree to be overgrown and model to be overfitted, we prune them using the parameters in the model.
- The various parameters used in the model are criterion, max_depth, min_samples_leaf,min_samples_split.



```
DecisionTreeClassifier(max_depth=7, min_samples_leaf=15, min_samples_split=45)
```

- Fitting the train and test split data into the decision tree model and execute them.
- To view the tree graphically, we use export_graphviz module from sklearn.tree package.

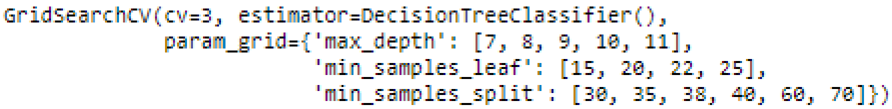


(Fig: 38 Decision Tree)

http://webgraphviz.com/

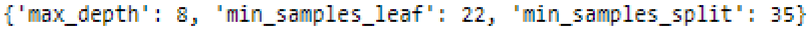
GRIDSEARCHCV WITH DECISION TREE:

- Giving the hyperparameters with our model as decision tree and fitting the training data.



```
GridSearchCV(cv=3, estimator=DecisionTreeClassifier(),
             param_grid={'max_depth': [7, 8, 9, 10, 11],
                         'min_samples_leaf': [15, 20, 22, 25],
                         'min_samples_split': [30, 35, 38, 40, 60, 70]})
```

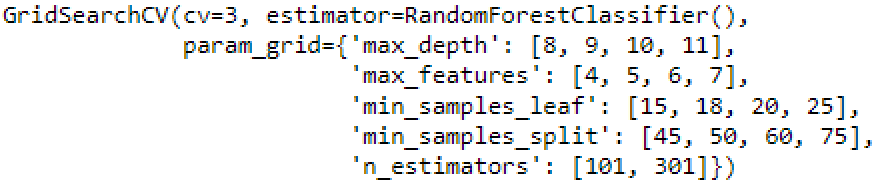
- We get the best parameters from our gridsearchcv using best_params_ attribute.



```
{'max_depth': 8, 'min_samples_leaf': 22, 'min_samples_split': 35}
```

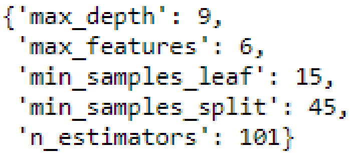
- Using the estimator that gave highest score which searching, to predict on both the train data and test data.

- The various parameters are max_depth, max_features, min_samples_leaf, min_samples_split, n_estimators that are given in param_grid.
- The random forest model is fitted to the GridSearchCV with the param_grid and cross validation.



```
GridSearchCV(cv=3, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [8, 9, 10, 11],
                         'max_features': [4, 5, 6, 7],
                         'min_samples_leaf': [15, 18, 20, 25],
                         'min_samples_split': [45, 50, 60, 75],
                         'n_estimators': [101, 301]})
```

- To know the best selected parameters by the model, using best_params_ attribute.

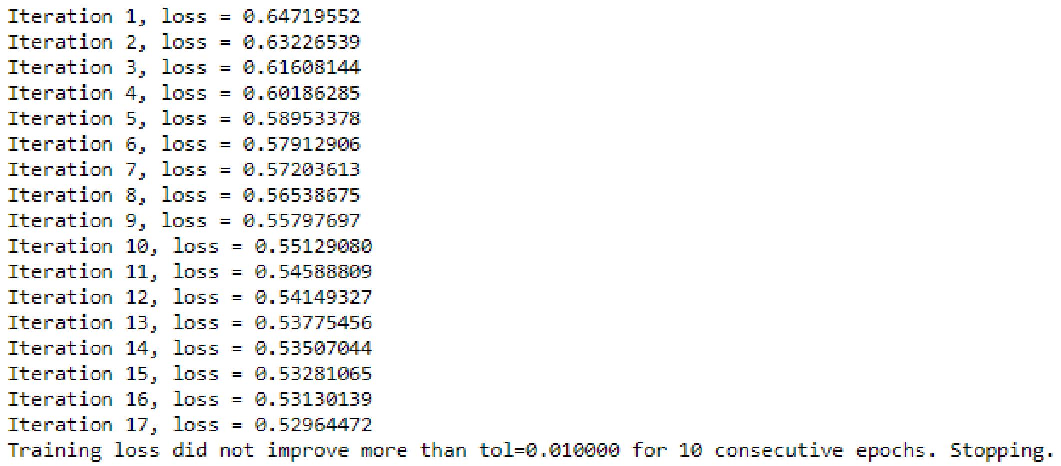


```
{'max_depth': 9,
 'max_features': 6,
 'min_samples_leaf': 15,
 'min_samples_split': 45,
 'n_estimators': 101}
```

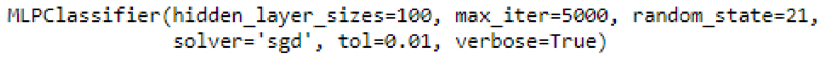
- Assigning the best estimator attribute to the model and using them for predicting the train and test data

ARTIFICIAL NEURAL NETWORKS:

- The model is created using MLPClassifier module from the sklearn.neural_network package.
- The Artificial Neural Network model needs the data to be scaled before applying data on the model, because of the non-linearity in the activation function and numerical rounding errors.
- Also scaling can accelerate learning and improve performance of the model.
- Here, we use StandardScaler module from the sklearn.preprocessing package, which uses Z-Scaling mechanism.
- The training data is fitted and transformed, while the test data is fitted alone to the model to use the same Mean and Standard Deviation for both the data.
- The various parameters used are Number of Hidden Layers, Number of Neurons in hidden layers, Maximum Iteration, Type of Solver, Type of Activation, and Tolerance Rate.



```
Iteration 1, loss = 0.64719552
Iteration 2, loss = 0.63226539
Iteration 3, loss = 0.61608144
Iteration 4, loss = 0.60186285
Iteration 5, loss = 0.58953378
Iteration 6, loss = 0.57912906
Iteration 7, loss = 0.57203613
Iteration 8, loss = 0.56538675
Iteration 9, loss = 0.55797697
Iteration 10, loss = 0.55129080
Iteration 11, loss = 0.54588809
Iteration 12, loss = 0.54149327
Iteration 13, loss = 0.53775456
Iteration 14, loss = 0.53507044
Iteration 15, loss = 0.53281065
Iteration 16, loss = 0.53130139
Iteration 17, loss = 0.52964472
Training loss did not improve more than tol=0.010000 for 10 consecutive epochs. Stopping.
```

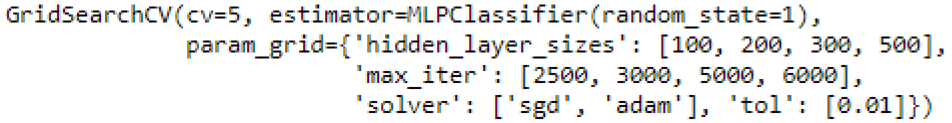


```
MLPClassifier(hidden_layer_sizes=100, max_iter=5000, random_state=21,
              solver='sgd', tol=0.01, verbose=True)
```

- Using the obtained the model we can predict on the train data and test data.

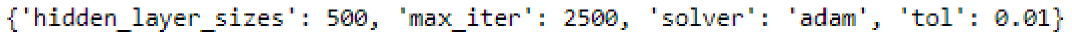
GRIDSEARCHCV FOR THE NEURAL NETWORKS:

- Various set of parameters called hyper parameters are given to the GridSearchCV, and the best set of parameters are returned by the model.
- Here, the MLPClassifier model is fitted to the GridSearchCV withthe hyper parameters and tuning them.



```
GridSearchCV(cv=5, estimator=MLPClassifier(random_state=1),
             param_grid={'hidden_layer_sizes': [100, 200, 300, 500],
                         'max_iter': [2500, 3000, 5000, 6000],
                         'solver': ['sgd', 'adam'], 'tol': [0.01]})
```

- The best parameters are selected as below by the model.



```
{'hidden_layer_sizes': 500, 'max_iter': 2500, 'solver': 'adam', 'tol': 0.01}
```

- This is used to predict the train data and test data.

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

PREDICTION USING ACCURACY:

- The Accuracy is calculated using the accuracy_score function from sklearn.metrics package.
- Here, we predict and find accuracy usin the best parameters from GridSearchCV for the models.

The Details are as follows:

Accuracy of cart on Train Data :

```
0.7852380952380953
```

Accuracy of Cart on Test Data:

```
0.7711111111111111
```

Accuracy of RF on Train Data:

```
0.8042857142857143
```

Accuracy of Rf on Test Data:

```
0.7844444444444445
```

Accuracy of ANN on Train Data:

```
0.7761904761904762
```

Accuracy of ANN on Test Data:

```
0.7688888888888888
```

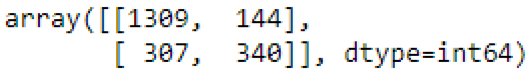
INFERENCE ON ACCURACY ACROSS CART,RF,ANN:

- Across the CART, RF, ANN Models, the Accuracy score is high for the Random Forest Model with 0.80 for the Train data and 0.79 for the Test data.

CONFUSION MATRIX:

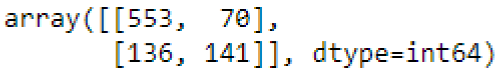
- The Confusion matrix is created using the confusion_matrix module from sklearn.metrics package.

CONFUSION MATRIX OF CART ON TRAIN DATA:



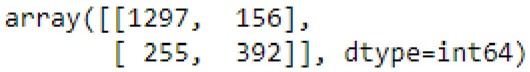
```
array([[1309,  144],
       [ 307,  340]], dtype=int64)
```

CONFUSION MATRIX OF CART ON TEST DATA:



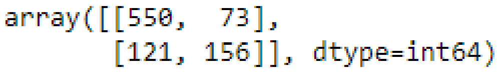
```
array([[553,  70],
       [136, 141]], dtype=int64)
```

CONFUSION MATRIX OF RF ON TRAIN DATA:



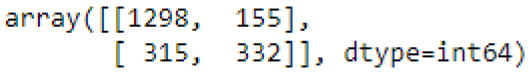
```
array([[1297,  156],
       [ 255,  392]], dtype=int64)
```

CONFUSION MATRIX OF RF ON TEST DATA:



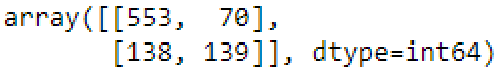
```
array([[550,  73],
       [121, 156]], dtype=int64)
```

CONFUSION MATRIX OF ANN ON TRAIN DATA:



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

CONFUSION MATRIX OF ANN ON TEST DATA:



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

INFERENCE FROM CONFUSION MATRIX ON CART, RF, ANN MODELS:

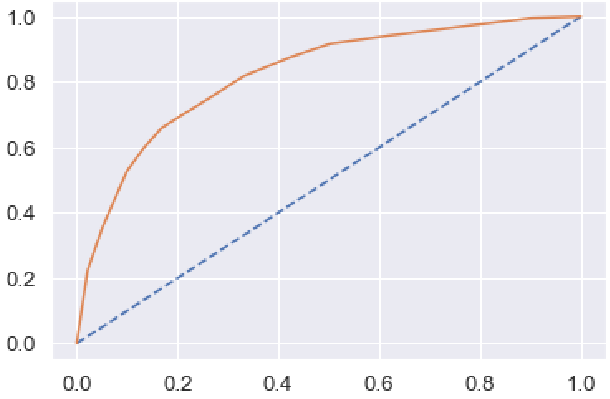
- Comparing the Confusion matrix on the Train data across CART, RF and ANN models, the Random Forest has lower False Positive and False Negative.

- For the Confusion matrix on the Test Data, the Random Forest model has lower False Negative and moderate False Positive to ANN Test data.

ROC CURVE:

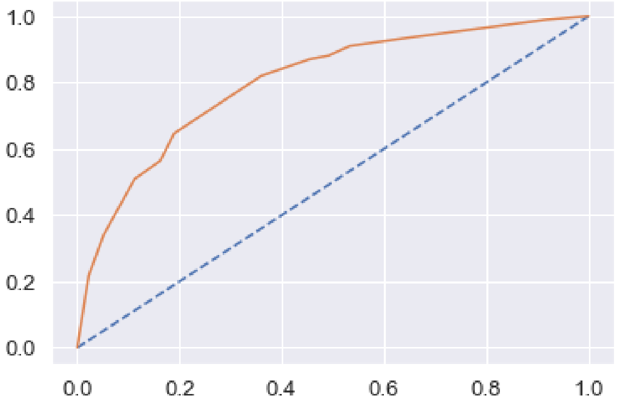
- The ROC Curve is plotted using roc_curve module from sklearn.metrics package.

ROC CURVE OF CART ON TRAIN DATA:



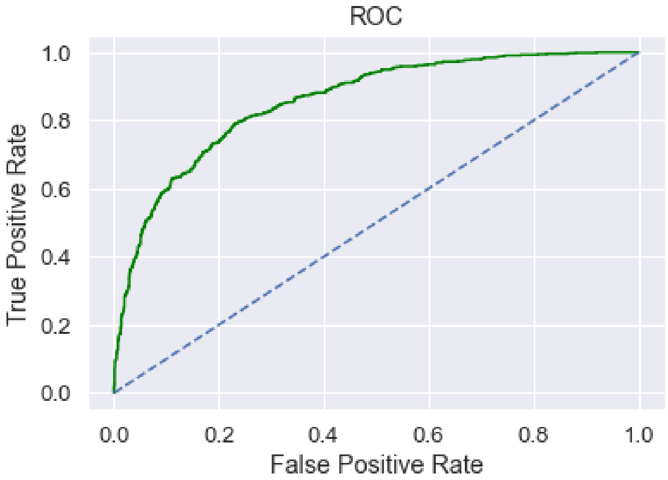
(Fig: 39 Cart Train ROC)

ROC CURVE OF CART ON TEST DATA:



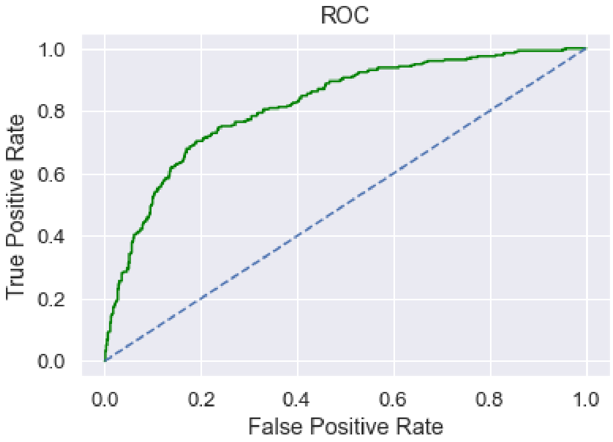
(Fig: 40 Cart test ROC)

ROC CURVE OF RF ON TRAIN DATA:



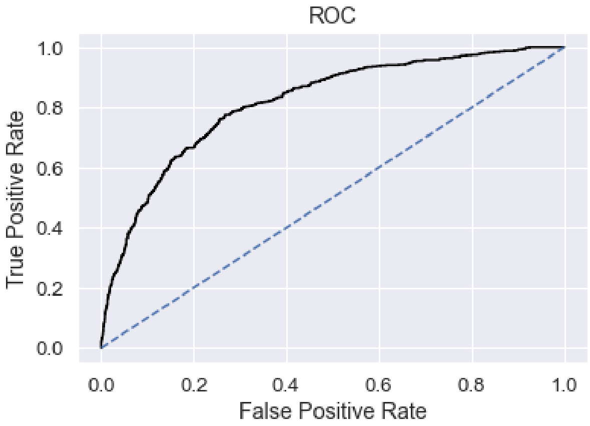
(Fig: 41 RF Train ROC)

ROC CURVE OF RF ON TEST DATA:



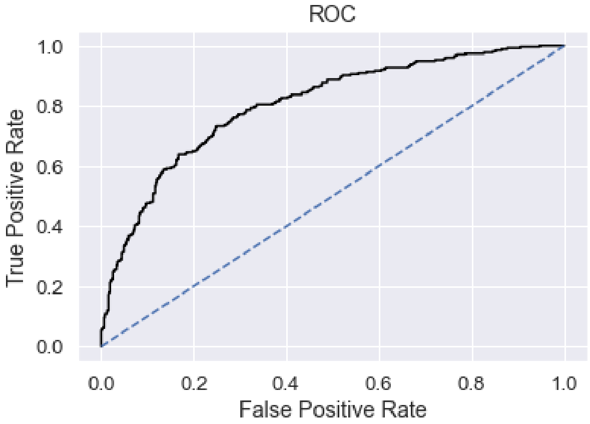
(Fig: 42 RF test ROC)

ROC CURVE OF ANN ON TRAIN DATA:



(Fig: 43 ANN train ROC)

ROC CURVE OF ANN ON TEST DATA:



(Fig: 44 ANN test ROC)

ROC-AUC SCORE:

- The ROC-AUC score is calculated using roc_auc_score module from sklearn.metrics package.

ROC-AUC SCORE OF CART ON TRAIN DATA:

```
Area under Curve is 0.823
```

ROC-AUC SCORE OF CART ON TEST DATA:

```
Area under Curve is 0.801
```

ROC-AUC SCORE OF RF ON TRAIN DATA:

```
Area under Curve is 0.8563713512840778
```

ROC-AUC SCORE OF RF ON TEST DATA:

```
Area under Curve is 0.8181994657271499
```

ROC-AUC SCORE OF ANN ON TRAIN DATA:

```
Area under Curve is 0.8166831721609928
```

ROC-AUC SCORE OF ANN ON TEST DATA:

```
Area under Curve is 0.8044225275393896
```

INFERENCE ON ROC CURVE AND ROC-AUC SCORE ACROSS THE MODELS:

- Comparing the AUC score across CART, RF and ANN models, the Random Forest model has high score for Train data with 0.86 score and Test data with 0.82 score.

OVERALL CONCLUSION:

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 good model.
Change is the most important variable for predicting diabetes

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

Neural Network Conclusion

Train Data:

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

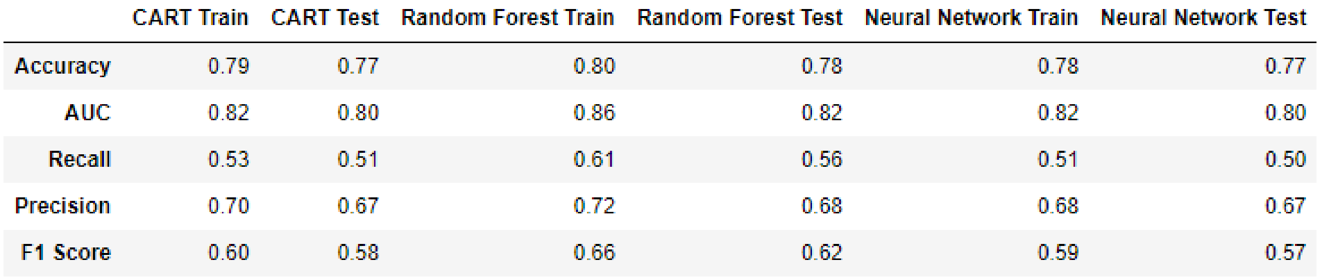
Test Data:

- 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.

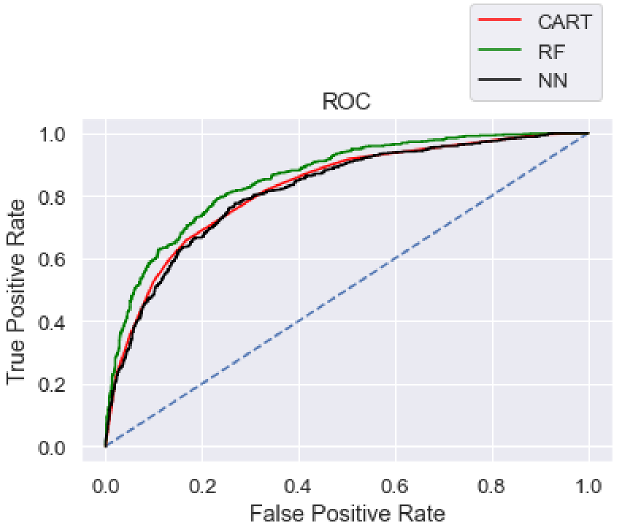
- By combining all the performance metrics from CART, RF and ANN models into a dataframe, we get the below table.



	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

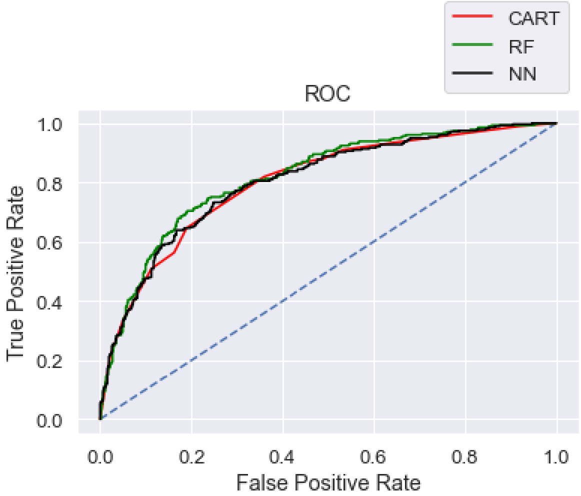
(Tab: 13 All list for best decision)

- From the above table, we can say that none of the models have been Overfitted and gives good values for their Test data.
- All the models give good results for the given problem, but we select the one that is best run or well optimized for the Business Problem.
- We could see that the Random Forest model has high values of Accuracy on the Train data (0.80) and Test data (0.79) and lower Accuracy on the Artificial Neural Networks for Train data (0.76) and CART for Test data (0.75).
- Much higher AUC score with Random Forest model for both the Train data (0.86) and Test data (0.82) and lower AUC score on ANN Train data (0.80) and CART test data (0.77).
- Recall is considerably good with Random Forest model.
- Precision is much higher with Random Forest model.
- The F1 Score is high with Random Forest model.
- Checking the ROC Curve for all the 3 models for their respective Train data.



(Fig: 45 Final train Roc)

- The Random Forest well broader curve comparatively.
- Checking ROC curve for all the 3 models for their respective Test data.



(Fig: 46 Final test Roc)

- The Artificial Neural Networks and Random Forest models have nearly same curve.
- We can conclude that the Random Forest model is the best optimized model for the given Business Problem.

2.5 Inference: Based on the whole Analysis, what are the business insights and rec ommendations.

I strongly recommended we collect more real time unstructured data and past data if possible.

This is understood by looking at the insurance data by drawing relations between different variables such as day of the incident, time, age group, and associating it with other external information such as location, behavior patterns, weather information, airline/vehicle types, etc.

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

Key performance indicators (KPI) The KPI's of insurance claims are:

- Reduce claims cycle time
- Increase customer satisfaction
- Combat fraud
- Optimize claims recovery
- Reduce claim handling costs Insights gained from data and AI-powered analytics could expand the boundaries of insurability, extend existing products, and give rise to new risk transfer solutions in areas like a non-damage business interruption and reputational damage.

```
                   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
```

- Here we can see that the column Agency_Code is given as high priority for the choosing the firm with 32%.
- The Product Name is selected next as it has various tour insurances of about 20%.
- The people select the Sales and the amount for the tour insurance policy with 19%.
- The least preferred attributes are Channel and Type with significant value.

FINAL CONCLUSION

✓ I am selecting the **RF** model, as it has better accuracy, precision, recall, f1 score better than other two ~~CART & ANN.~~

THANK YOU…!