DATA MINING ASSIGNMENT PROJECT

1. CLUSTERING:

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

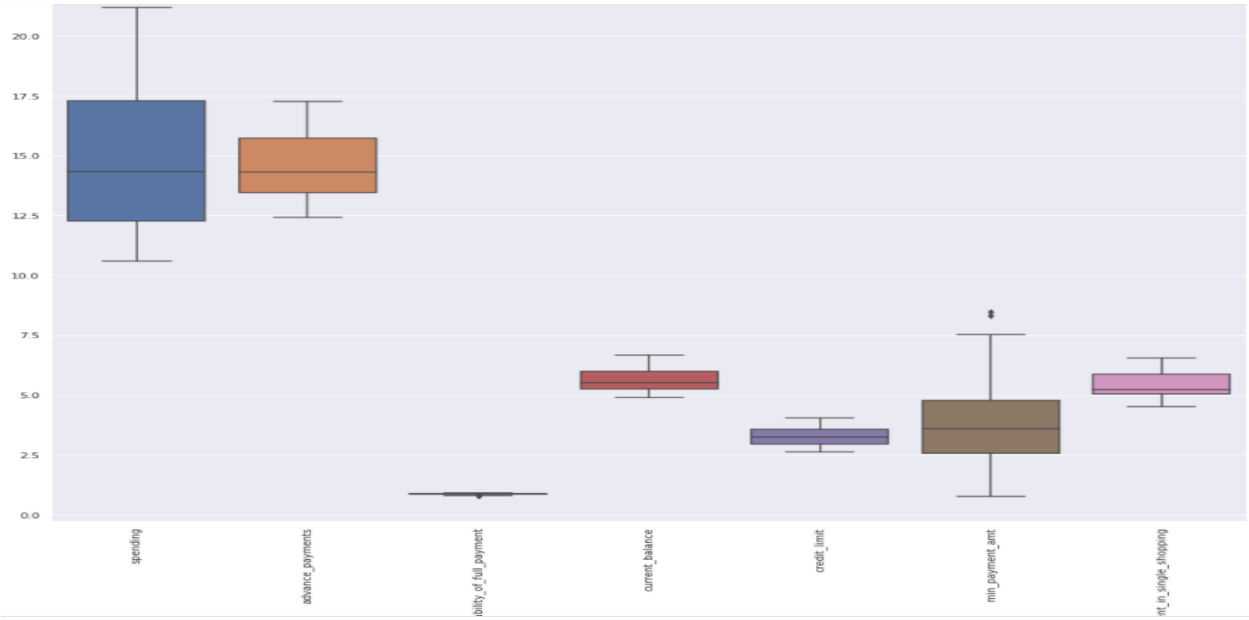
1.1 Read the data and do exploratory data analysis. Describe the data briefly.

- The first step is to import the necessary packages and libraries. After importing the data, we will perform exploratory data analysis using pandas profiling.

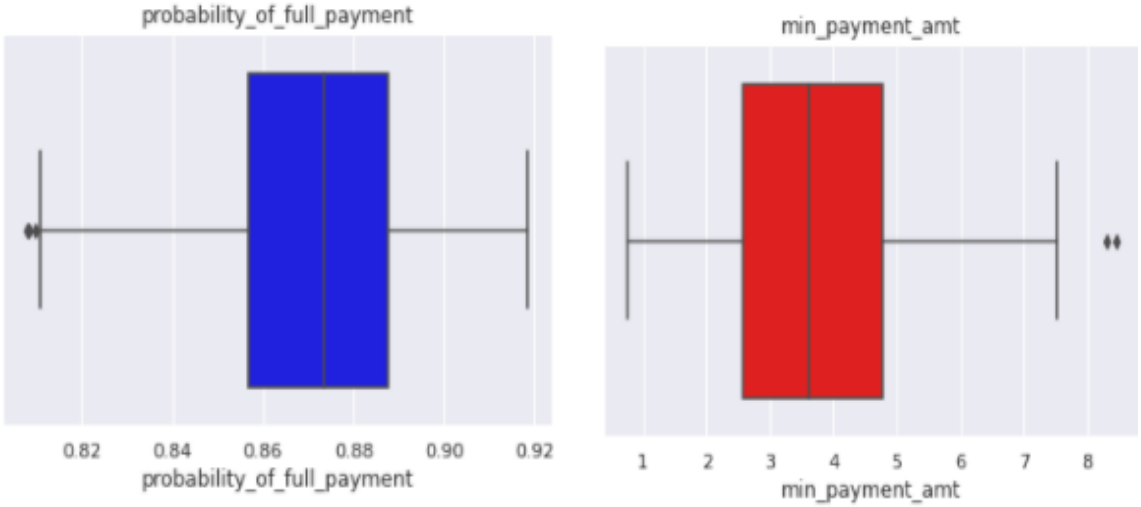
EDA of bank marketing.html

Also please find the html file attached with the document file.

- Next step is checking for outliers.

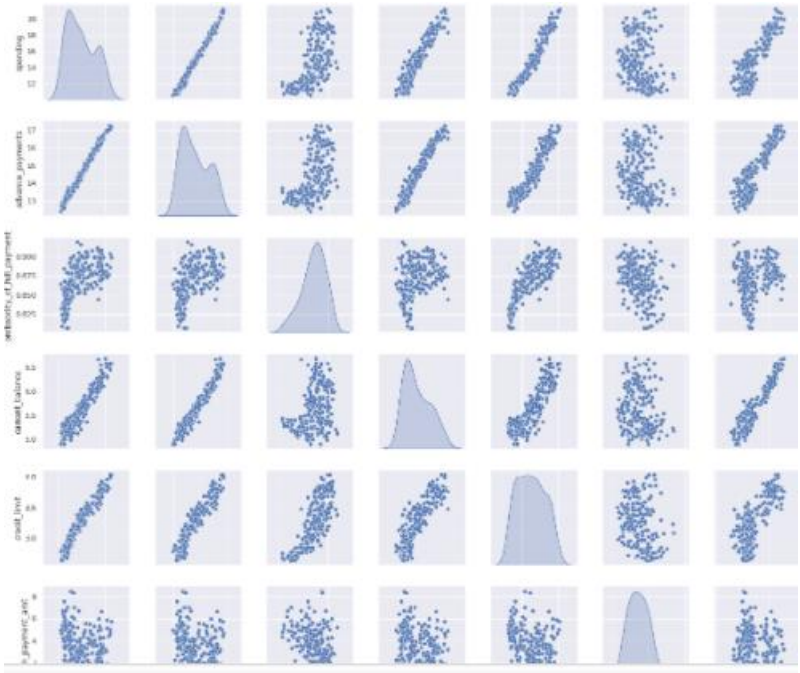


- The shape of the data is (210,7)
- The information of the data tells that all the variable values are float in nature.
- No null values in the data.
- No missing values in the data.
- No duplicates in the dataset.
- All the variables are of numeric type.
- Mean and median seems to be almost equal.
- The standard deviation for the variable spending is high while comparing with other variables.
- The distribution looks even.
- The boxplot of spending, advance payment, current payment, credit limit and max spent have no outliers
- The boxplot of probability of full payment, min payment amount have few outliers
- Positively skewed- spending, advance payment, current payment, credit limit, min payment amount and max spent.
- Negatively skewed - probability of full payment
- The distribution of spending variable seen to be from 10-22
- The distribution of advance payment variable seen to be from 12-17
- The distribution of probability of full payment is from 0.80-0.92
- The distribution of current payment is from 5.0-6.5
- The distribution of credit limit is from 2.5-4.0
- The distribution of min payment amount is from 2-8
- The distribution of max spent is from 4.5-6.5



  - We can see the outliers in the data and it is fine to process as it has no extreme values or the least values.

- Multivariate analysis

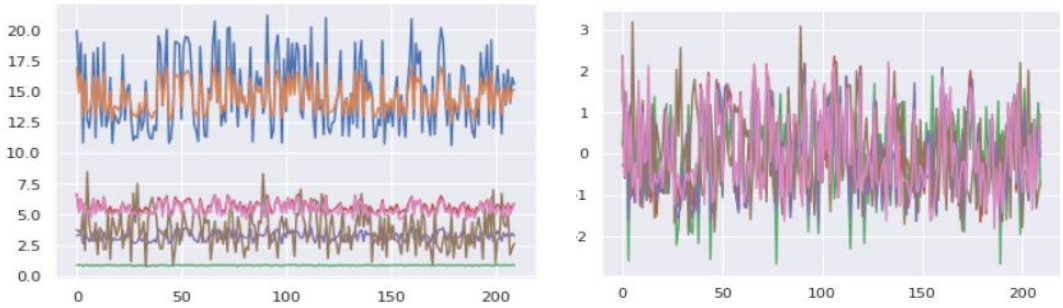


Strong positive correlations between

- spending and advance payment
- 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

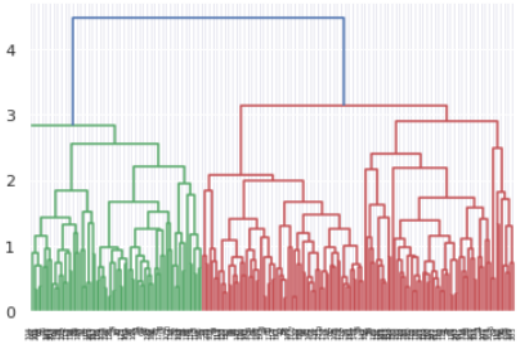
Yes, Scaling is necessary because the values of the variables are different and the model works based on the distance measure. Here, spending, advance payment are in different values which may carry more weightage to the data. After scaling, the data of all the variables will have same range. We use z-score to scale the 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

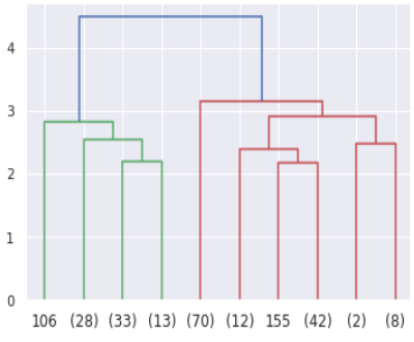
1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendogram and briefly describe them

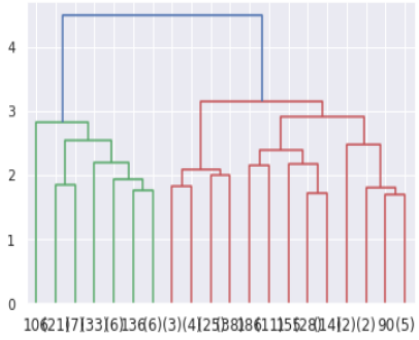
Hierarchical clustering – linkage method

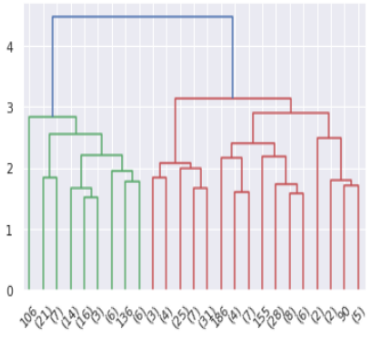


The Dendogram shows that all the data points have clustered to different clusters by linkage method. In order to find the optimal number of clusters, we use lastp.

We use p=10 p=20 p=25







When p=10, 20 and 25, all the data points have 3 clusters. We use fclusters to map these clusters and set the criterion as max.

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

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

These are the 3 cluster frequencies.

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

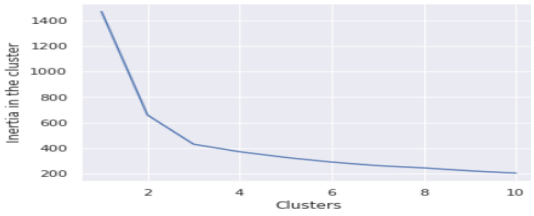
Cluster grouping is based on the dendrogram, 3 or 4 looks good. For further analysis, dataset had gone for 3 group cluster solution based on the hierarchical clustering. Also in real time, there colud have been more variables value captured - tenure, BALANCE_FREQUENCY, balance, purchase, installment of purchase, others. And three group cluster solution gives a pattern based on high/medium/low spending with max_spent_in_single_shopping (high value item) and probability_of_full_payment(payment made).

1.4 Apply K-Means clustering on scaled data and determine optimum clusters (2 pts). Apply elbow curve and silhouette score (2 pts). Interpret the inferences from the model (1 pts).

Calculating wss for values when k=1 to 11. Using loop to find the optimal number of clusters.

```
[1470.0,
 659.1717544870407,
 430.65897315130053,
 371.65314399951626,
 327.05106145316563,
 290.59003059682186,
 262.9396666832541,
 245.01985320749773,
 222.18558154711812,
 204.43340697039045]
```

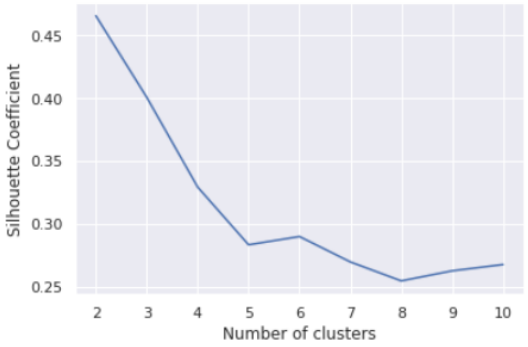
We look at the plot and say 3 is the optimal number of clusters to process using elbow method.



	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

```
                                [0.46577247686580914,
                                 0.4007270552751299,
                                 0.3291966792017613,
                                 0.28316654897654814,
                                 0.2897583830272518,
                                 0.2694844355168535,
                                 0.25437316027505635,
                                 0.2623959398663564,
0.32732359239831144              0.2673980772529917]
```

Plotting the silhouette scores. From the plot, the number of optimal clusters will be 3 or 4.




```
array([0, 2, 0, 1, 0, 1, 1, 2, 0, 1, 0, 2, 1, 0, 2, 1, 2, 1, 1, 1, 1, 1,
       0, 1, 2, 0, 2, 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 0, 0, 2, 0, 0,
       1, 1, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 2, 1, 1, 2, 2, 0,
       0, 2, 0, 1, 2, 1, 0, 0, 1, 0, 2, 1, 0, 2, 2, 2, 2, 0, 1, 2, 0, 2,
       0, 1, 2, 0, 2, 1, 1, 0, 0, 0, 1, 0, 2, 0, 2, 0, 2, 0, 0, 1, 1, 0,
       2, 2, 0, 1, 1, 0, 2, 2, 1, 0, 2, 1, 1, 1, 2, 2, 0, 1, 2, 2, 1, 2,
       2, 0, 1, 0, 0, 1, 0, 2, 2, 2, 1, 1, 2, 1, 0, 1, 2, 1, 2, 1, 2, 2,
       1, 2, 2, 1, 2, 0, 0, 1, 0, 0, 0, 1, 2, 2, 2, 1, 2, 1, 2, 0, 0, 0,
       2, 1, 2, 1, 2, 2, 2, 2, 0, 0, 1, 2, 2, 1, 1, 2, 1, 0, 2, 0, 0, 1,
       0, 1, 2, 0, 2, 1, 0, 2, 0, 2, 2, 2], dtype=int32)
```

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

	Cluster_Size	Cluster_Percentage
1	67	31.90
2	72	34.29
3	71	33.81

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

Cluster 1, Cluster 2, Cluster 3 are High, medium and low spending.

There are 67 data points which lies under cluster 1, 72 under cluster 2 and 71 under cluster 3.

The values for each variable under each cluster is also shown. The frequency of clusters occuring and the type of cluster for each row is also shown in the above figures.

By using k–means clustering, we find the optimal cluster to be used is 3 because using elbow method we find that after 3 there is no huge drop in the values. The silhouette score seems to be less which indicates that all the data points are properly clustered. Clustering grouping based on 3 and 4 looks fine.

1.5 Describe cluster profiles for the clusters defined (2.5 pts). Recommend different promotional strategies for different clusters in context to the business problem in-hand (2.5 pts).

Cluster 1 – High Spending:

- In order to increase the purchase, we can give some reward points.
- We can also increase the credit card limit and the spending habits of the customer.
- We can provide discount percentages for maximum spending products.
- We can provide offers for full payment products.
- If the customer is a good repayer, we can also provide loans on credit cards.
- To increase the spending, we can tie up with most known brands and provide offers.

Cluster 2 – Medium Spending:

- For medium spending customers, we can provide offers on travel site, ecommerce, airlines, hotels and other sites which will encourage for more spending.
- In order to increase transactions, promote on premium cards.
- Providing offers on paying electricity bills, purchases, mobile recharges will maintain a good score on the card.
- Decreasing the interest rate for good repayment customers will increase in the spending.

Cluster 3 – Low Spending

- Offers can be provided for early repayment.
- The remainders for using the cards and repayment should be given frequently.
- Having a tie up with grocery products, electronic gadgets, payment bills and others may increase spending habits.

A

2. CART-RF-ANN

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

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check (3 pts), Interpret the inferences from the descriptive statistics in a detailed manner (2 pts).

- The first step is to import the necessary packages and libraries. After importing the data, we will perform exploratory data analysis using pandas profiling.

Insurance.html

- This HTML file has all kinds of exploratory data analysis. Please find the html file attached with the document file.

```
<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
```

- It shows that the datatypes are int, float and object. We have to change the object type to integer.
- No missing values in the data.

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

- Shape of the data - `(3000, 10)`

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

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

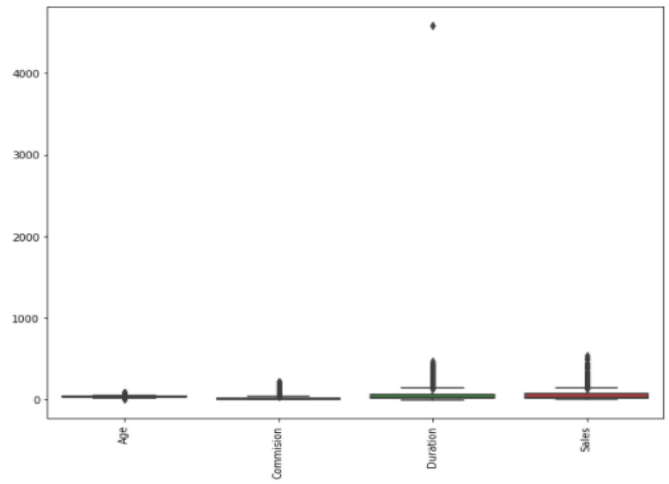
```
TYPE :  2
Airlines         1163
Travel Agency    1837
Name: Type, dtype: int64
```

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

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

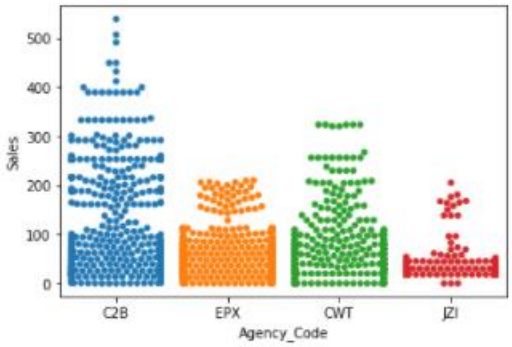
```
DESTINATION :  3
EUROPE       215
Americas     320
ASIA        2465
Name: Destination, dtype: int64
```

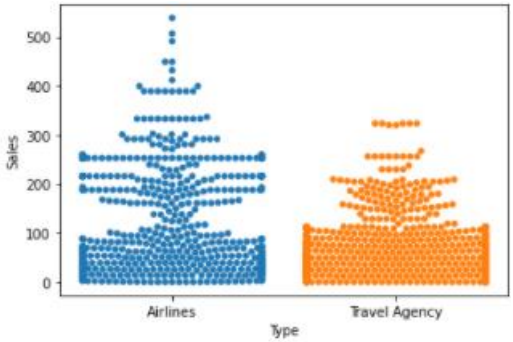
- There are 6 categorical values and 4 numeric values.
- Numerical – Age, comission, duration and sales
- Target variable - Claimed
- There most preferred type is Travel agency
- The most preferred Agency code is EPX
- The most preferred Channel is Online
- Customized plan is the most bought.
- Most selected destination is ASIA
- Duration has negative values. It may be a wrong entry
- There are 139 duplicates in the data. Since there is no unique ID, dropping off those rows.

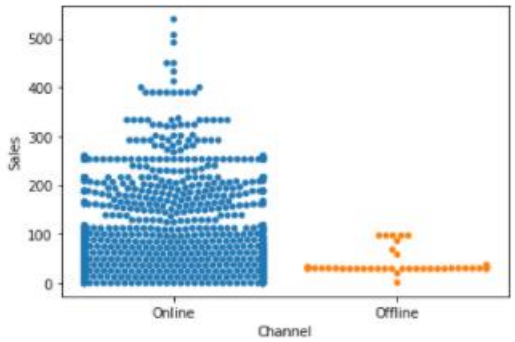


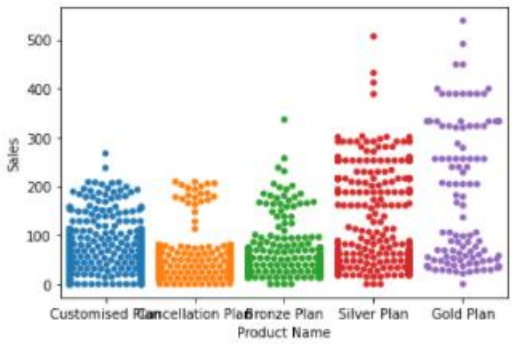
- There are outliers in all the variales. Since Random forest and CART handle outliers. It is not treated as of now. It is treated for ANN model.
- Positively skewed- Age, Commission, Duration, Sales

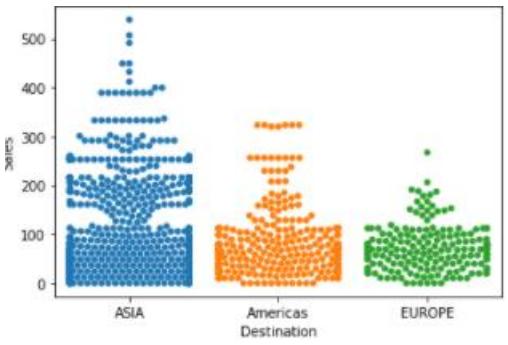
- The distribution of the age variable is from 20-80
- The distribution of Commission variable is from 0-30
- The distribution of Duration variable is from 0-100
- The distribution of Sales variable is from 0-300
- Plotting for variables.





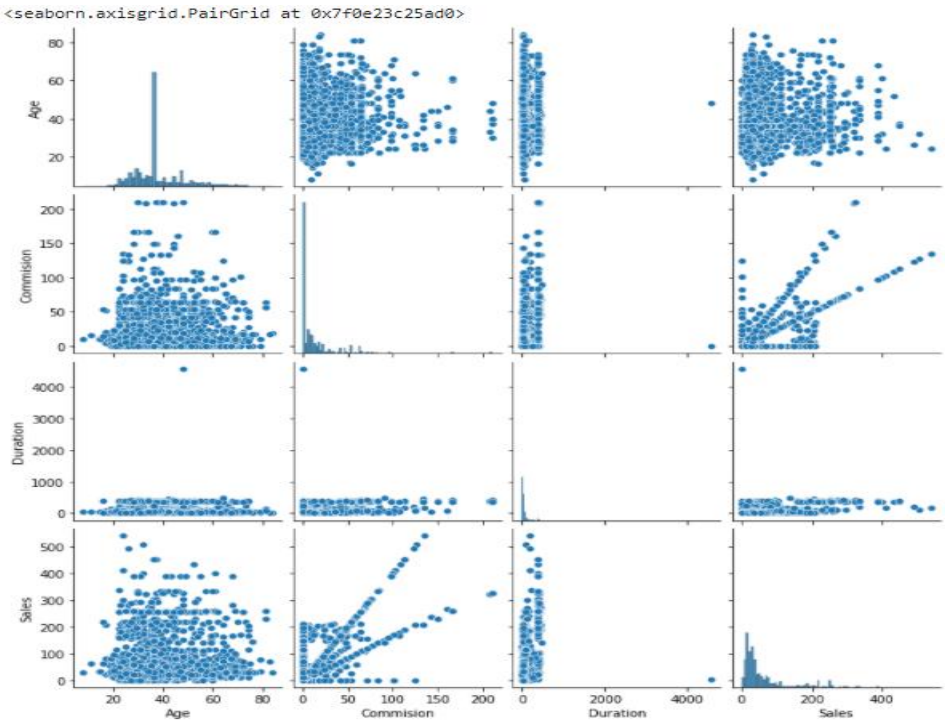






- Not much of multi collinearity is oserved.
- There is no negative correlation.
- Only positive correlation is observed.
- Pair wise distribution of continuous variables

```
<seaborn.axisgrid.PairGrid at 0x7f0e23c25ad0>
```



- Next step is to convert all the objects to categorical variables.
- In order to build our model, we are changing object type to numerical values.

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


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


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


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


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


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

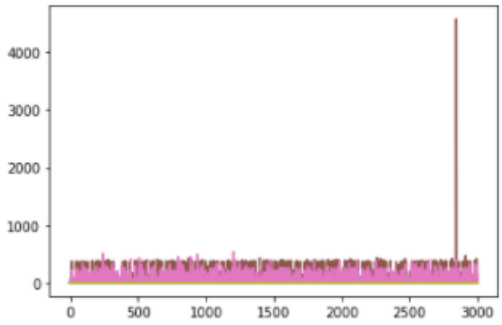
```
0    0.692
1    0.308
Name: Claimed, dtype: float64
```

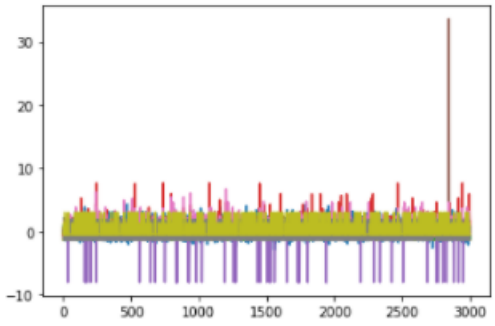
It is found that 30% is claimed and 70% is not claimed.

2.2 Data Split: Split the data into test and train (2 pts), build classification model CART (1 pts), Random Forest (1 pts), Artificial Neural Network(1 pts).

Extracting the target column into separate vector for tarining and testing set.

We are splitting the ratio to 70:30






```
x_train (2100, 9)
x_test (900, 9)
```
- checking the dimensions of the splitted data

- Building a Decision tree classifier and finding the optimal values to the training set.

```
{'criterion': 'gini', 'max_depth': 4.85, 'min_samples_leaf': 44, 'min_samples_split': 260}
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=4.85, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=44, min_samples_split=260,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=1, splitter='best')
```

- Checking the feature importance

	Importance
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

- Predicting on training dataset

	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

- Random forest classifier

```
{'max_depth': 6, 'max_features': 3, 'min_samples_leaf': 8, 'min_samples_split': 46, 'n_estimators': 350}
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=6, max_features=3,
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=8, min_samples_split=46,
                       min_weight_fraction_leaf=0.0, n_estimators=350,
                       n_jobs=None, oob_score=False, random_state=1, verbose=0,
                       warm_start=False)
```

	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

	Importance
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

These are the feature importance.

To find the optimal values for hyper parameters. Grid search is used to find optimal numbers.

- Building a Neural network classifier

```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=200, learning_rate='constant',
              learning_rate_init=0.001, max_fun=15000, max_iter=2500,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=1, shuffle=True, solver='adam',
              tol=0.01, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

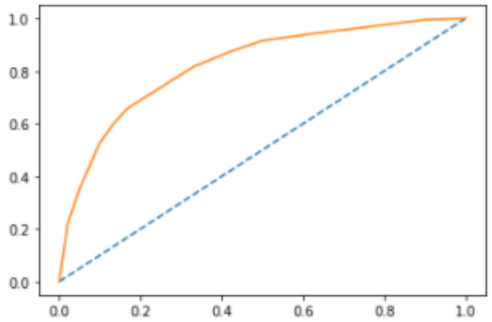
- Probabilities

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

2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy (1 pts), Confusion Matrix (1 pts), Plot ROC curve and get ROC_AUC score for each model (1 pts), Write inferences on each model (1 pts).

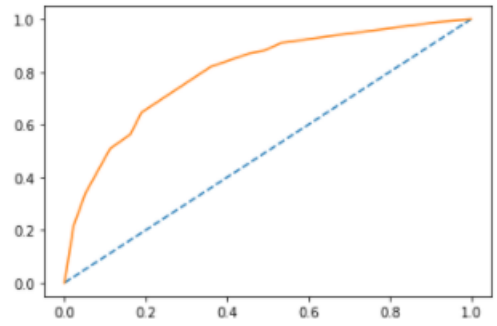
- AUC of training set of decision tree classifier

```
AUC: 0.823
[<matplotlib.lines.Line2D at 0x7f0e1abfd750>]
```



- AUC of test set of decision tree classifier

```
AUC: 0.801
[<matplotlib.lines.Line2D at 0x7f0e18746190>]
```



- Confusion matrix of decision tree classifier

```
array([[1309,  144],
       [ 307,  340]])
```

- Training data Accuracy

```
0.7852380952380953
```

- Classification report of training data

```
              precision    recall  f1-score   support

           0       0.81      0.90      0.85      1453
           1       0.70      0.53      0.60       647

    accuracy                           0.79      2100
   macro avg       0.76      0.71      0.73      2100
weighted avg       0.78      0.79      0.78      2100
```

- Precision, Recall and F1 score of training set

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

- Confusion matrix of decision tree classifier- test set

```
array([[553,  70],
       [136, 141]])
```

- Testing data Accuracy

```
0.7711111111111111
```

- Classification report of training data

```
              precision    recall  f1-score   support

           0       0.80      0.89      0.84       623
           1       0.67      0.51      0.58       277

    accuracy                           0.77       900
   macro avg       0.74      0.70      0.71       900
weighted avg       0.76      0.77      0.76       900
```

- Precision, Recall and F1 score of training set

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

- Confusion matrix of random forest classifier

```
array([[1297,  156],
       [ 255,  392]])
```

- Training data Accuracy – Random forest

```
0.8042857142857143
```

- Classification report of training data – Random forest

```
              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
```

- Precision, Recall and F1 score of training set

```
rf_train_precision  0.72
rf_train_recall  0.61
rf_train_f1  0.66
```

- Confusion matrix of random forest classifier – test set

```
array([[550,  73],
       [121, 156]])
```

- Testing data Accuracy – Random forest

```
0.7844444444444445
```

- Classification report of testing data – Random forest

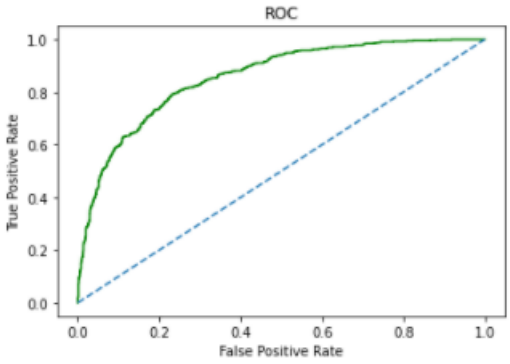
```
              precision    recall  f1-score   support

           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
```

- Precision, Recall and F1 score of testing set

```
rf_test_precision  0.68
rf_test_recall  0.56
rf_test_f1  0.62
```

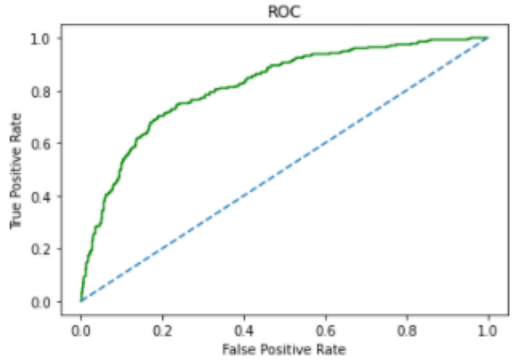
- AUC curve for random forest training data

```
Area under Curve is 0.8563713512840778
```



- AUC for testing data

```
Area under Curve is 0.8181994657271499
```



- Confusion matrix of neural network classifier

```
array([[1298,  155],
       [ 315,  332]])
```

- Training data Accuracy – Neural network

```
0.7761904761904762
```

- Classification report of training data – Random forest

	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

- Precision, Recall and F1 score of training set

```
nn_train_precision  0.68
nn_train_recall  0.51
nn_train_f1  0.59
```

- Confusion matrix of neural network classifier – test set

```
array([[553,  70],
       [138, 139]])
```

- Testing data Accuracy – neural network

```
0.7688888888888888
```

- Classification report of testing data – neural network

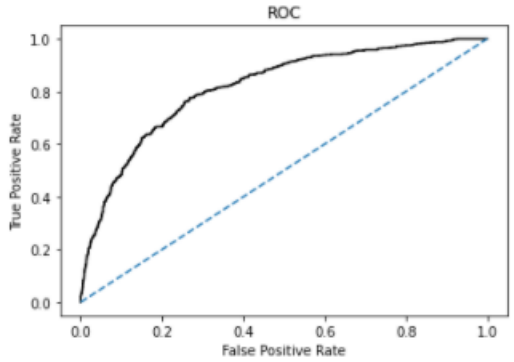
```
              precision    recall  f1-score   support

           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
eighted avg       0.76      0.77      0.76       900
```

- Precision, Recall and F1 score of testing set

```
nn_test_precision  0.67
nn_test_recall  0.5
nn_test_f1  0.57
```

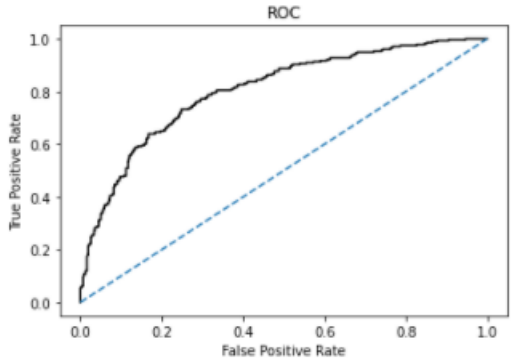
- ROC for training data

```
Area under Curve is 0.8166831721609928
```



- ROC for testing data

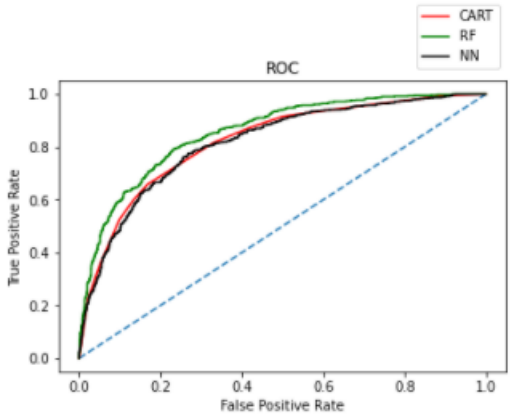
```
Area under Curve is 0.8044225275393896
```



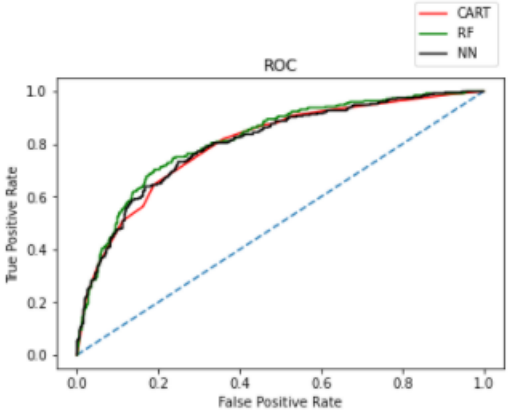
2.4 Final Model - Compare all models on the basis of the performance metrics in a structured tabular manner (3 pts). Describe on which model is best/optimized (2 pts).

	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

```
<matplotlib.legend.Legend at 0x7f0e1aa9ae90>
```




```
<matplotlib.legend.Legend at 0x7f0e160bd310>
```



Random Forest model is the best model in comparison with Decision tree CART and neural networks since it gives better accuracy, precision, recall and f1 score.

2.5 Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective.

This business problem can be improved and get better by following these approaches.

- It will be better and accurate if we have more past data and information.
- We can look into relations between the variables such as age, time with other variables like location and so on.
- Increasing the conversation with the customers by streaming online which helps in more profit.
- As per our data, 90% of the insurance is done by online.
- The JZI agency which needs to be improved since they have very less sales.
- Based on the 80% of accuracy we got, we can cross sell the insurance based on claim patterns.
- Other interesting fact is that more sales happen via Agency than Airlines and the trend shows the claim are processed more at Airline. Focus needs to be kept.
- Other facts are:
    - Reduce claims cycle time
    - Increase customer satisfaction
    - Reduce fraud
    - Optimize claims recovery
    - Expanding the boundaries of insurability
    - Extending existing products

These insights and recommendations helps us in improving our business.