DATA MINING PROJECT

Data Analysis Report

Prepared By

JAI GOUTHAM

Submitted on
06-12-2020

CONTENTS

1. Project Objective …………………………………………………………………………………… 3

2. Problem 1: Clustering ………………………………………………………………………………….4 – 17
   a) Data Insights & EDA……………………………………………………………………….4
   b) Solution 1.1 ……………………………………………………………………………….4
   c) Solution 1.2 ……………………………………………………………………………….11
   d) Solution 1.3 ……………………………………………………………………………….12
   e) Solution 1.4 ……………………………………………………………………………….14
   f) Solution 1.5 ……………………………………………………………………………….16

3. Problem 2: CART, RF, ANN …………………………………………………………………….18 – 34
   a) Assumptions …………………………………………………………………………….18
   b) Importing Packages ………………………………………………………………….18
   c) Solution 2.1 ……………………………………………………………………………….19
   d) Solution 2.2 ……………………………………………………………………………….24
   e) Solution 2.3 ……………………………………………………………………………….29
   f) Solution 2.4 ……………………………………………………………………………….33
   g) Solution 2.5 ……………………………………………………………………………….34

PROJECT OBJECTIVE

Problem 1: Clustering

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

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

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

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

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

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

Problem 2: CART-RF-ANN

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

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it.
2.2 Data Split: Split the data into test and train, build classification model CART, Random Forest, Artificial Neural Network
2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model
2.4 Final Model: Compare all the model and write an inference which model is best/optimized.
2.5 Inference: Based on the whole Analysis, what are the business insights and recommendations

PROBLEM 1: CLUSTERING

Data Insights and EDA:

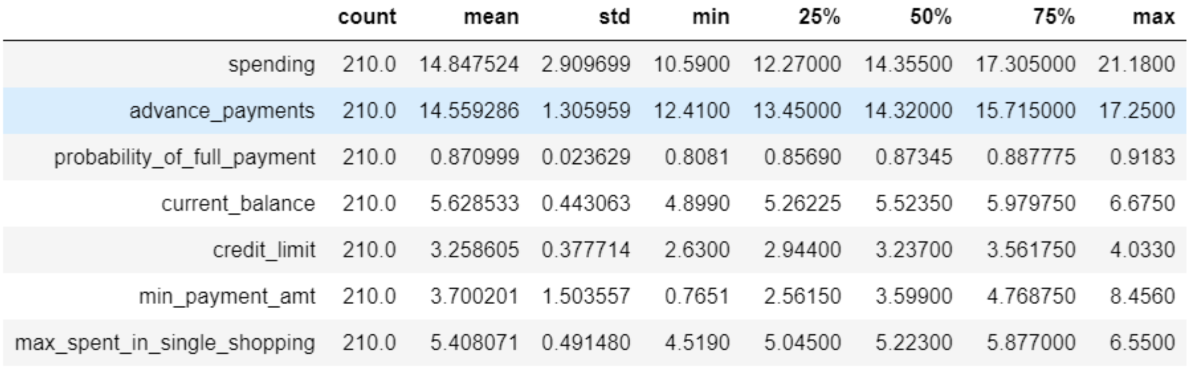
1.1 Read the data and do exploratory data analysis. Describe the data briefly.

The dataset: “bank_marketing_part1_Data.csv” which contains data of 210 customers and 7 variables namely as follows:

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)

Description of dataset:

The below table which shows the count, average, maximum, minimum values for the variables like spending, advance payment, probability of full payment, current balance, credit limit, minimum payment amount, maximum spent in single shopping.



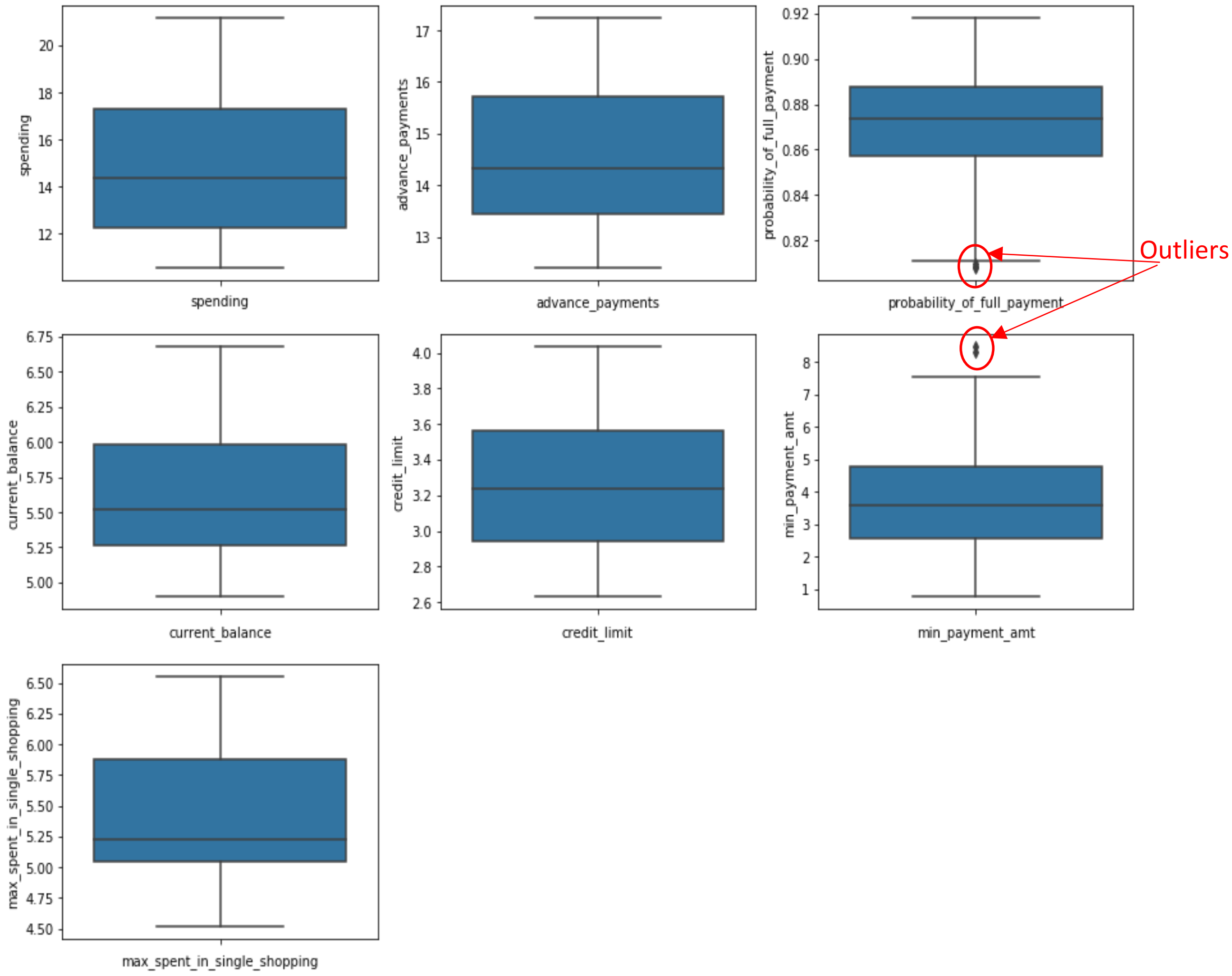
	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.847524	2.909699	10.5900	12.27000	14.35500	17.305000	21.1800
advance_payments	210.0	14.559286	1.305959	12.4100	13.45000	14.32000	15.715000	17.2500
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.85690	0.87345	0.887775	0.9183
current_balance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

Checking for Missing Values:

The dataset does not contain any missing/Null values. The total no of missing /Null values is 0.

Univariate Analysis:

Using the Boxplot in a dataset we can able to find the outliers, spread of values, median, range, etc., (outliers are the extreme values present in the dataset)



Inference from boxplot:

From the above Boxplots for all the variables, we can conclude that a very few outliers are present in the variable like **probability_of_full_payment**, **min_payment_amt**.

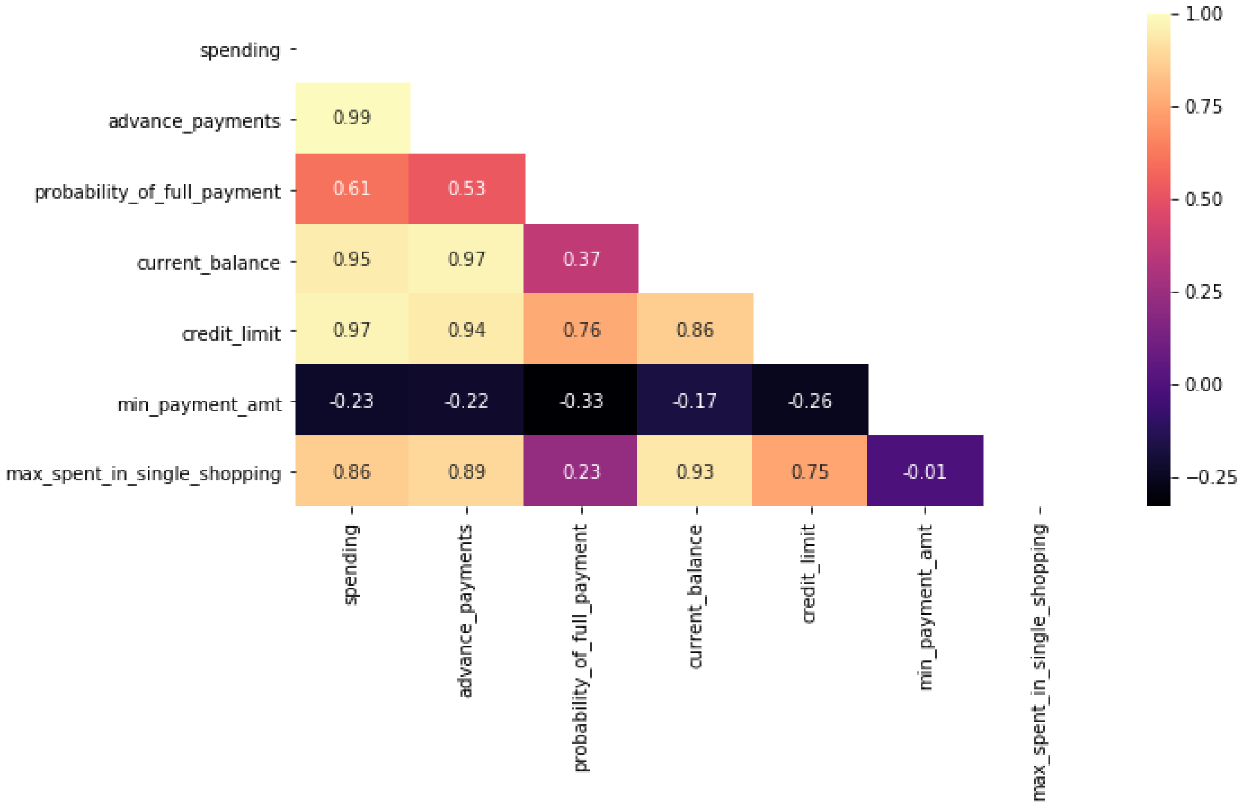
There are only a very few outliers are present in the dataset, so significantly it may not affect any impact on our dataset, so is no need to do outlier treatment.

- The customers in the dataset have higher spending capacity
- Most of the customers are making advance payments
- The probability of the customers for making full payment is more (86%-89%)
- Majority of the customers are maintaining the current balance in higher side.
- The maximum spending of customers in single shopping is high.

Multivariate Analysis

Heat Map

The Heat Map shows the relationship between different variables in our dataset. This graph can help us to check for any correlations between different variables.

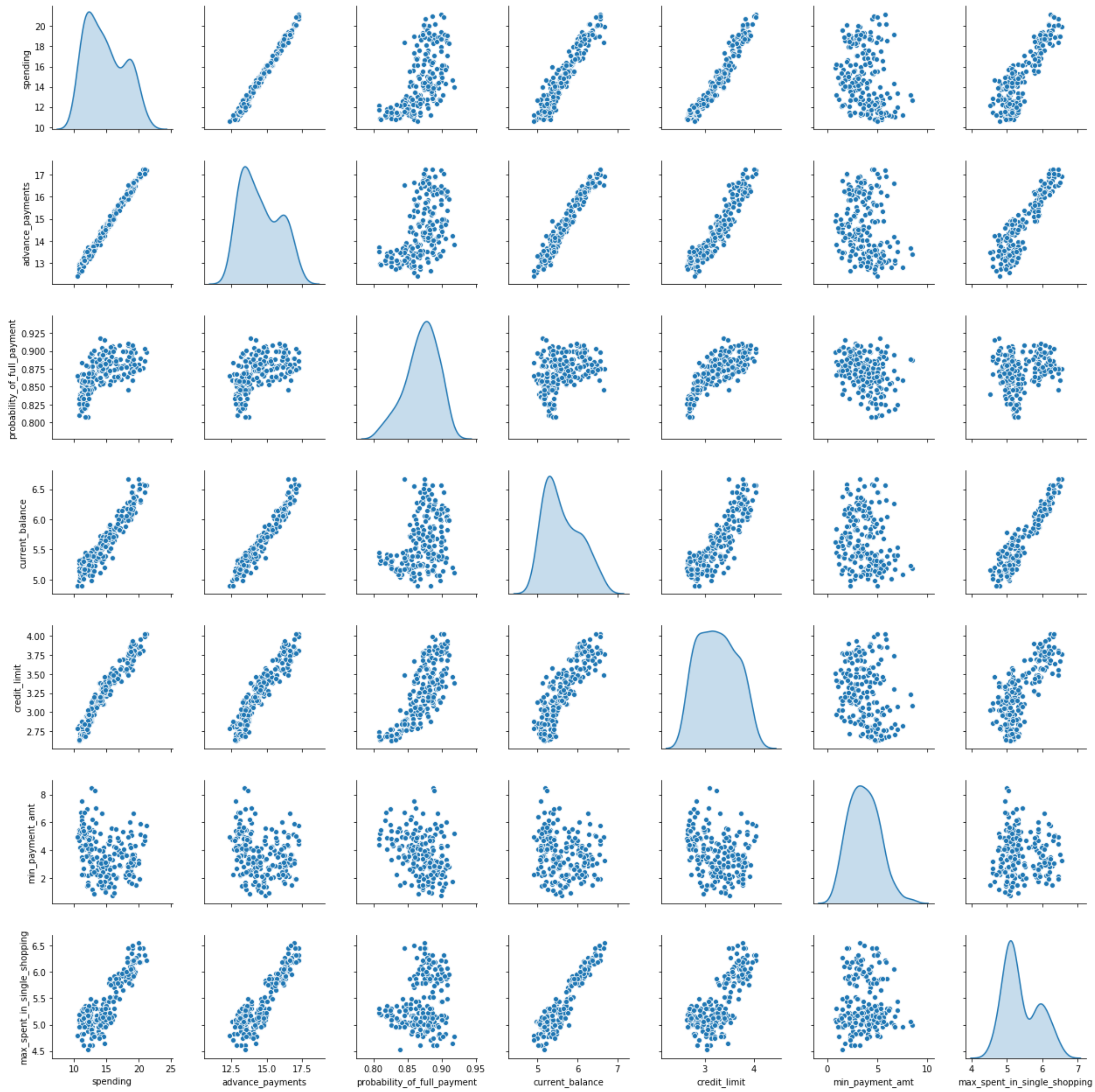


Inference from heatmap:

We can see that the there is a high positive correlation between the variables.

- The variables like advance payments, current balance, credit limit and maximum spending in single shopping are highly positive correlated with customer Spending
- The current balance, credit limit and maximum spending in single shopping are highly positive correlated with advance payments
- The credit limit and maximum spending in single shopping are highly positive correlated with current balance
- The variables like credit limit are highly positive correlated with probability of making full payment.

Pair Plot:



From the above pair plot we can able to understand the Univariate and Multivariate analysis and inferences for all the variables present in the dataset.

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

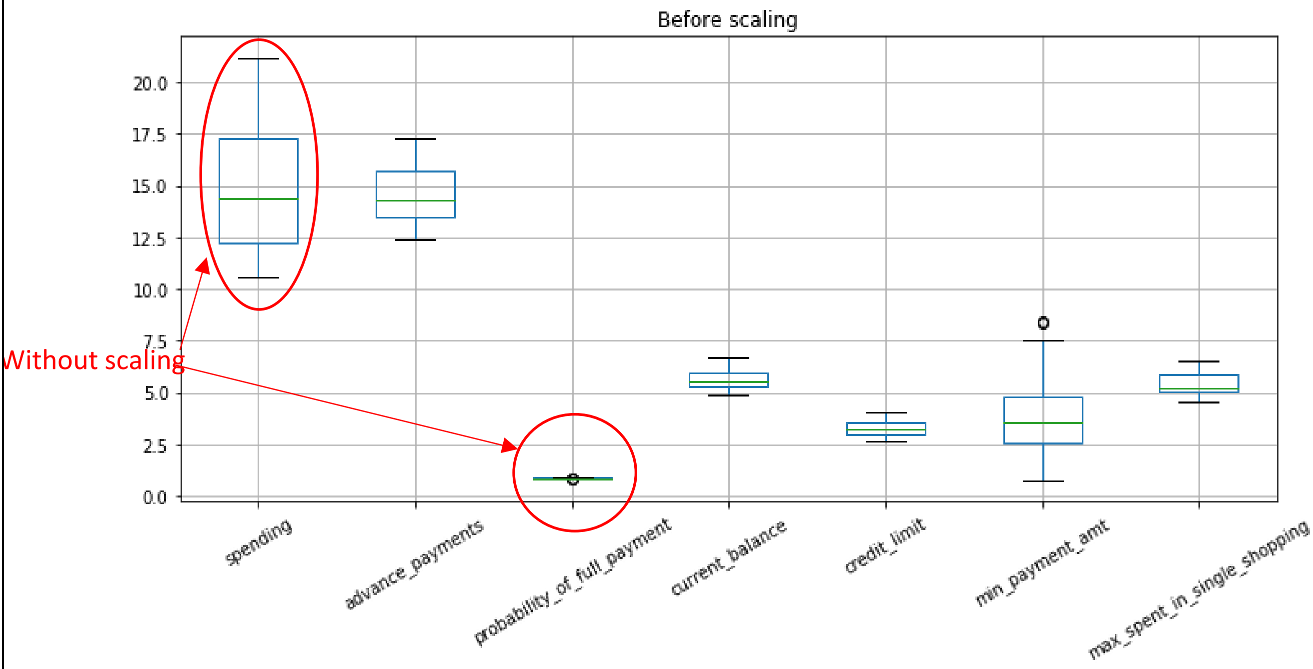
Feature scaling or Standardization is a technique used for Machine Learning algorithms which helps in data pre-processing. Especially for clustering this scaling feature is need to applied to all independent variables present in our dataset, so that it helps to standardize all the data into a similar range.

If feature scaling is not done, then a machine learning algorithm like clustering tends to give more weightage/bias for value which has high magnitude and it gives low weightage/bias for value which has low magnitude, regardless to the unit of the values.

For this dataset given to us, scaling is required for all the variables, since the variables are expressed in different units. Example: The customer spending in thousands(1000's), the advance payments in hundreds(100's) and the credit limit is in Ten thousand(10000's), whereas probability of full payment are expressed as fraction or decimal values.

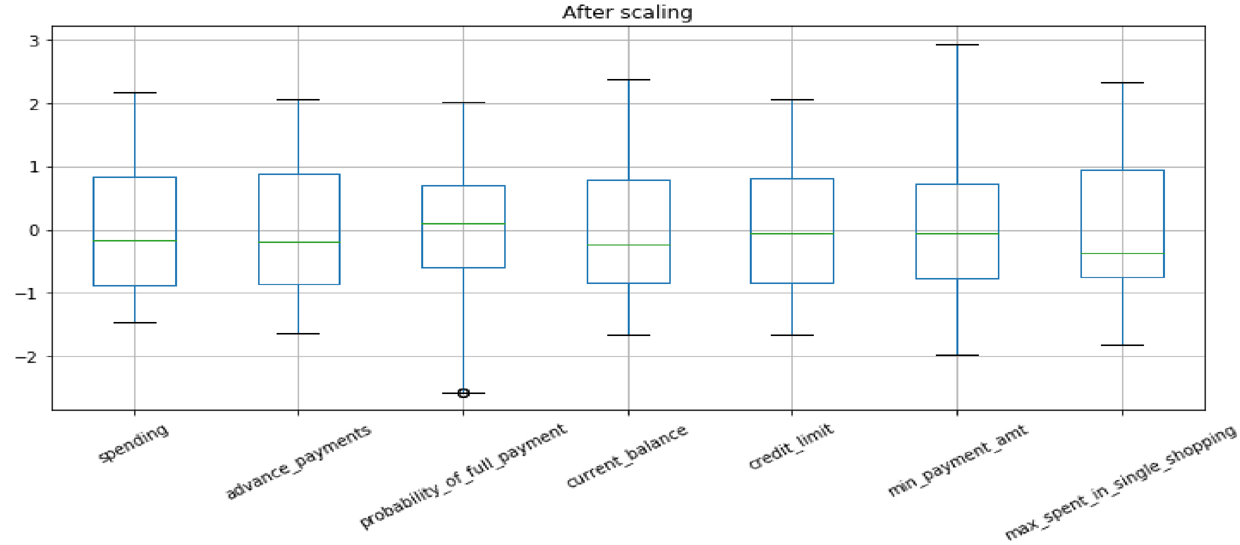
Since the values expressed in higher/lower units and the magnitude is varying between the variables, hence it is important to Scale the data using Standard Scaler/Min max scaler and therefore the standardization gives the output with the values having mean equal to 0 and standard deviation is 1.

Box plot before Scaling



Visually from above box plot, before scaling the spread of the data is varying in different magnitude and direction, the variable spending is ranging in higher value and the variable probability of full payment is very low.

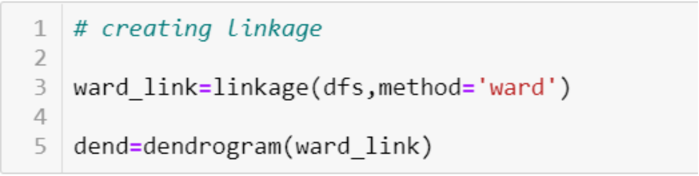
Box plot After Scaling



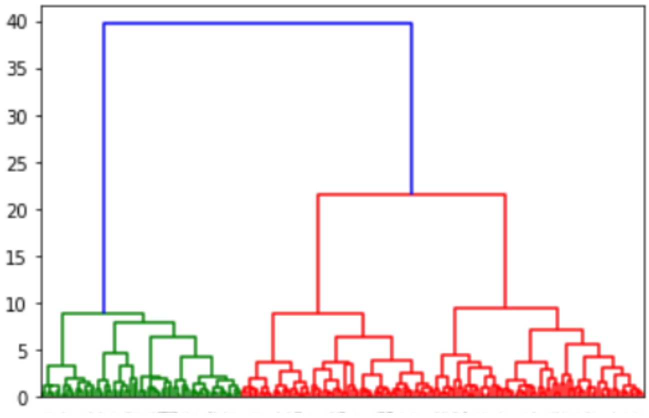
Visually from the above box plot after scaling the spread of the data is uniform in magnitude and direction and the variable having the mean is equal to zero and standard deviation is one.

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

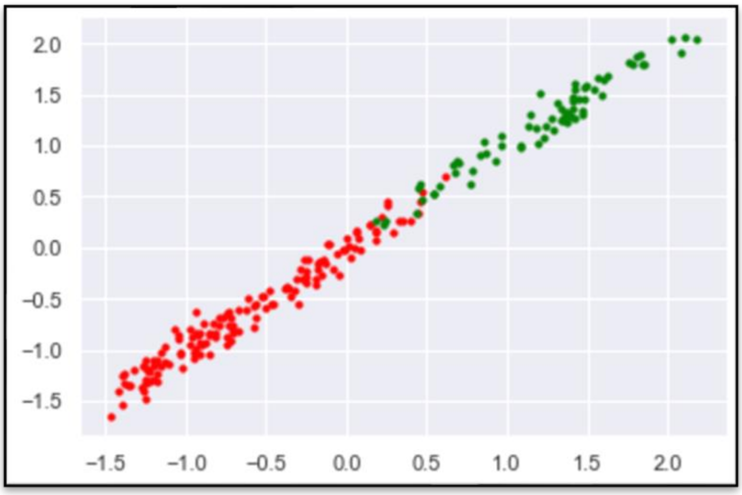
Clustering is a widely used in Unsupervised Learning technique in Machine Learning Clustering can be divided into three categories namely Agglomerative, Divisive and Partitioning. An algorithm that identifies the similar objects of data in dataset into groups called clusters. For the given dataset we will be using Hierarchical Clustering method to create optimum clusters and categorising the dataset on the basis of these clusters.



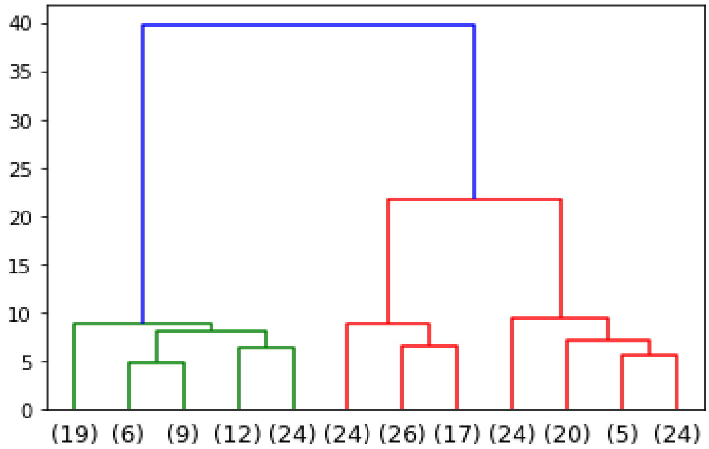
```
# creating Linkage

ward_link=linkage(dfs,method='ward')

dend=dendrogram(ward_link)
```



Plotting the cluster:

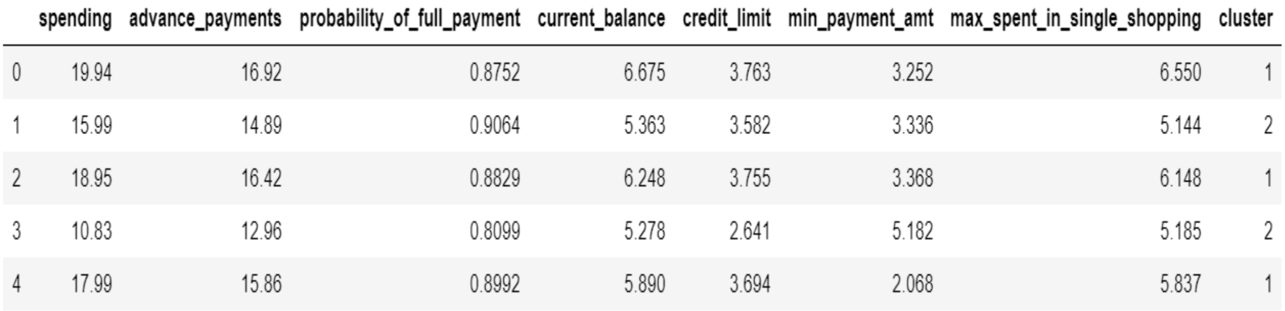


Dendrogram



From the above dendrogram we can able to see that there are two cluster are formed, the model which gives the optimal number of clusters. The colour green and red are the two clusters.

The below table which shows the clusters in the data frame.



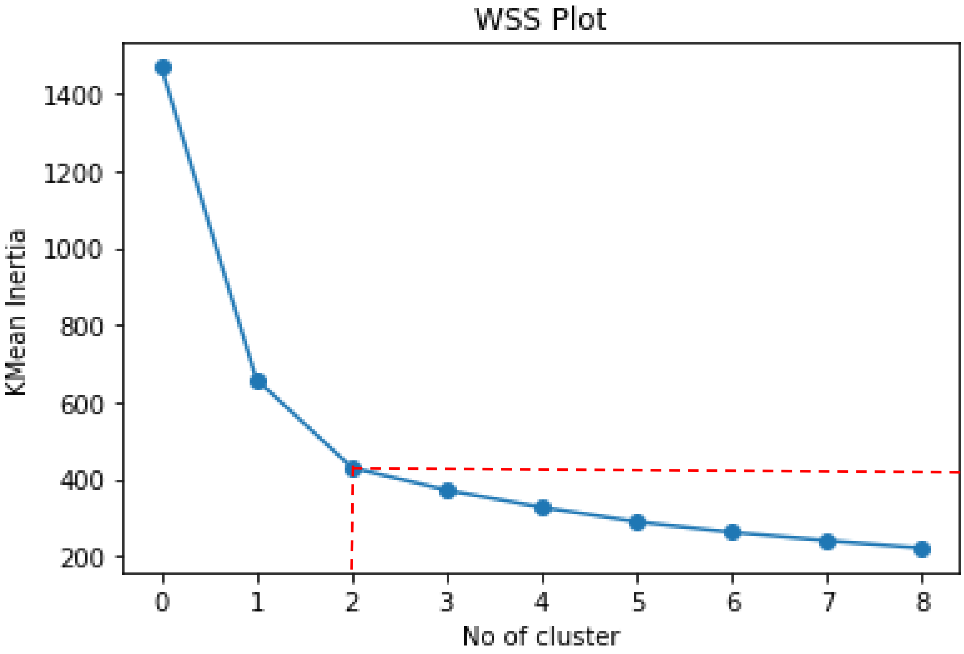
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	cluster
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	1
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	2
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1
3	10.83	12.96	0.8099	5.278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

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

K-means clustering is one of the unsupervised machine learning algorithms. K-means algorithm first identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

For the given dataset we have applied KMeans algorithm on the scaled dataset and we got the clusters as output. To check the optimal number of clusters the following methods are used 1.WSS plot 2.Silhouette analysis.

WSS Plot for finding optimum cluster:



Inference from WSS plot:

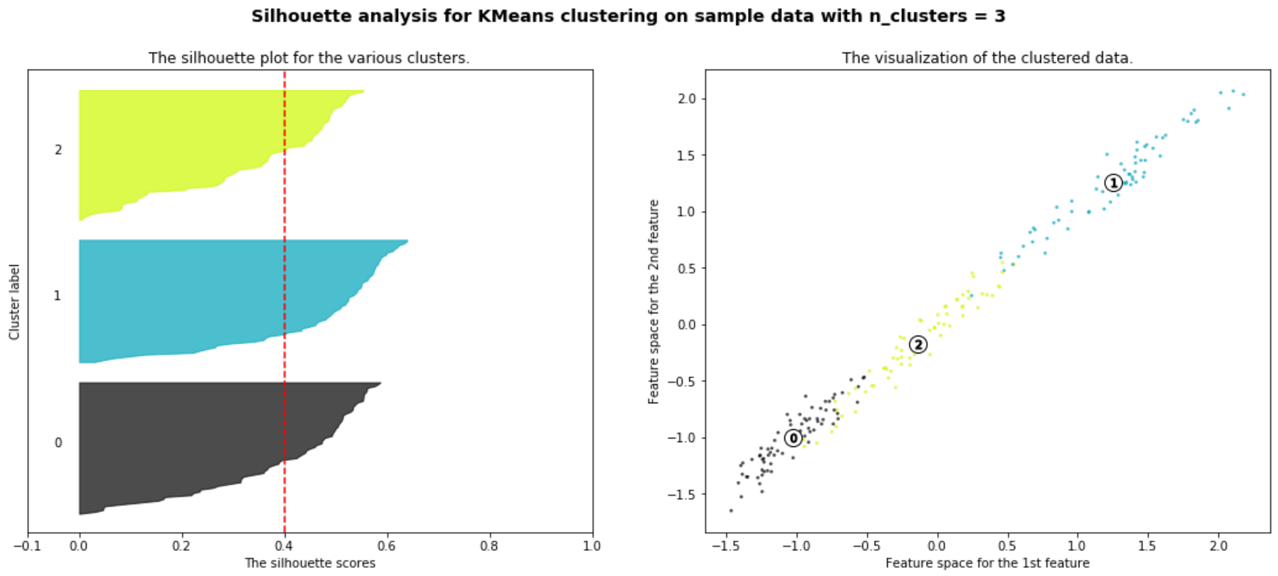
- We can conclude that the optimal number of clusters to be taken for k-means clustering is 3 since as per the elbow curve the curve bends at this point.
- From this curve the values are getting steep down from cluster 0 to 2, after 2nd cluster the curve is getting flat with respect to its number.

Silhouette Analysis

Silhouette analysis can be used to study the separation distance between the resulting clusters. The silhouette plot displays a measure of how close each point in one cluster is to points in the neighbouring clusters and thus provides a way to assess parameters like number of clusters visually. This measure has a range of [-1, 1].

Silhouette scores +1 indicate that the sample is far away from the neighbouring clusters, 0 indicates that the sample is on or very close to the decision boundary between two neighbouring clusters and -1 indicate that those samples might have been assigned to the wrong cluster. For our given dataset the following silhouette scores are formed.

For n clusters = 2 The average silhouette score is: 0.4657724.
For n clusters = 3 The average silhouette score is: 0.4007270.
For n clusters = 4 The average silhouette score is: 0.3291966.
For n clusters = 5 The average silhouette score is: 0.2859715.

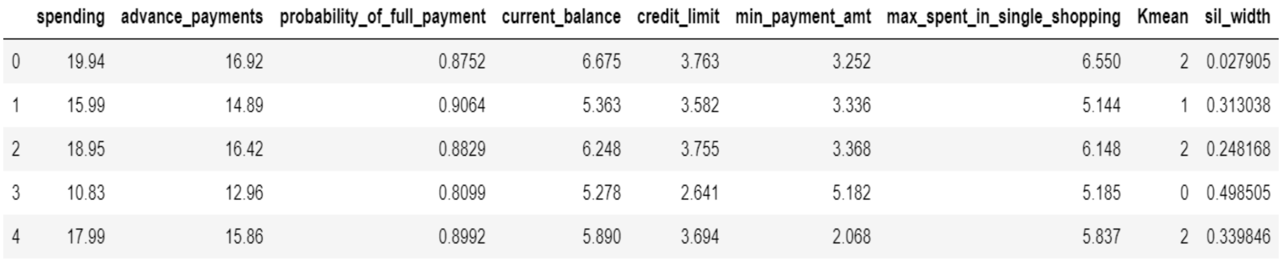


(REFER PYTHON CODE FILE FOR OTHER PLOTS)

Inference from Silhouette plot:

We can conclude that the optimal number of clusters to be taken for k-means clustering is 3 since it has got better results when compared to other n_clusters like 2,4,5,6. From the Below conditions we have been selected the n_cluster=3.

- From the silhouette plot the n_cluster =3, has high silhouette score 0.40027
- All the cluster labels are greater than the mean value (0.4)
- None of the cluster labels is below the mean value
- All the three cluster labels are looks similar size (colour-lime, cyan and black)



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Kmean	sil_width
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	2	0.027905
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	1	0.313038
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	2	0.248168
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	0	0.498505
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837	2	0.339846

After concluding the clusters, then the silhouette width for all the datapoints are formed to check how good is the separation. Out of 210 datapoints below gives the separation results, i.e. 96% the separation is greater than +1 silhouette value.

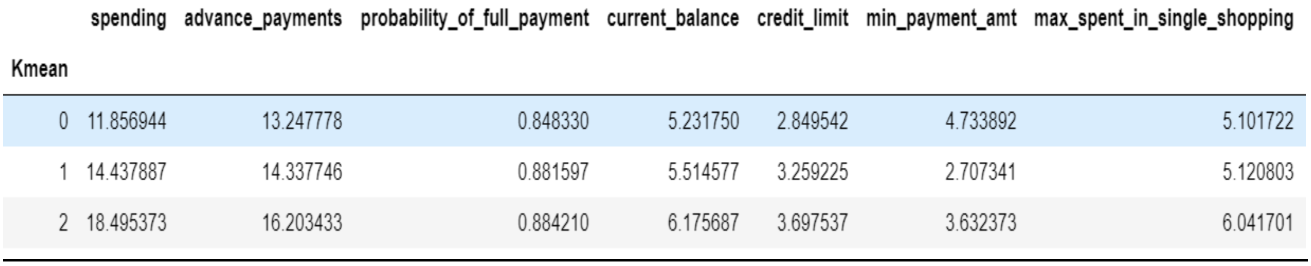
The number of datapoints less than zero silhouette width is: 14
The number of datapoints greater than zero silhouette width is: 196

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

Now, the final step is to identify the clusters that we have created using Hierarchical clustering and K-means clustering for our market segment analysis and devise promotional strategies for the different clusters.

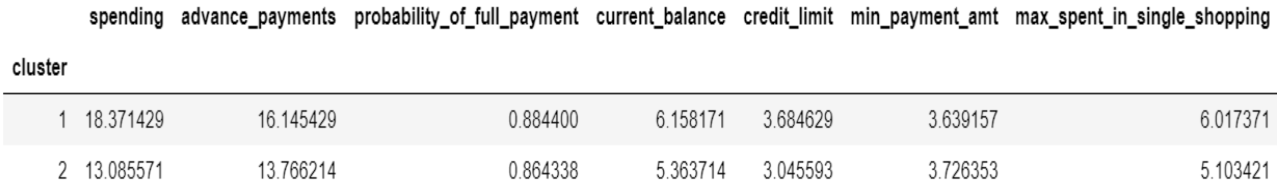
Since from the above analysis we have identified 2 clusters from hierarchical clustering and 3 optimal clusters from k-means clustering. We will now further analyse and determine the best clustering approach that can be helpful for the market segmentation problem in hand. We will first plot and map the clusters from both the methods.

K-Mean cluster: Average values



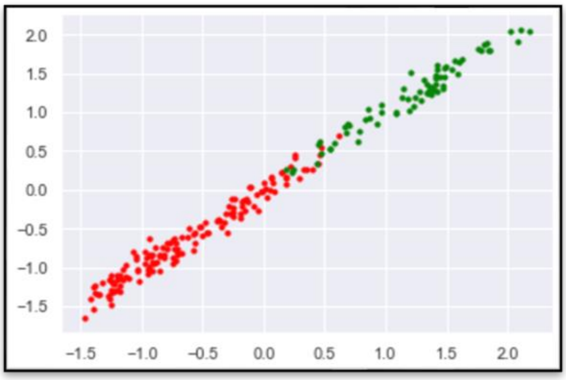
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
Kmean							
0	11.856944	13.247778	0.848330	5.231750	2.849542	4.733892	5.101722
1	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803
2	18.495373	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701

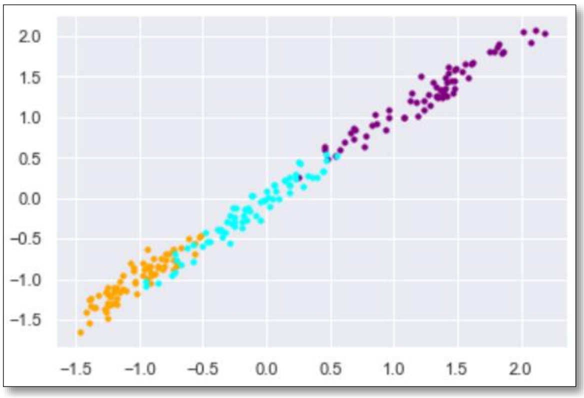
Hierarchical cluster: Average values



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
cluster							
1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371
2	13.085571	13.766214	0.864338	5.363714	3.045593	3.726353	5.103421

Plotting the two different clusters:





HIEARCHIAL CLUSTRING

K-MEANS CLUSTERING

Now, in the below table we have tabulated the averages for all the variables of the five clusters created from the above clustering using hierarchical and K-means methods. As per the values we can segment the clusters into two for Hierarchical and three segments for K-means clusters.

KMEAN CLUSTER	CLASS	CARD TYPE	AVGERAGE SPENDING	AVG.CREDIT LIMIT	PROMOTIONAL STATERGIES
Cluster 0	MIDDLE CLASS	GOLD CARD	11.85	28500	1.FREE COUPONS 2.EXTRA REWARD POINTS 3.ZERO ANNUAL CHARGES
Cluster 1	UPPER MIDDLE CLASS	DIAMOND CARD	14.43	32500	1.LOW INTEREST RATE 2.EXTRA CASHBACK 3.INCREASE LOYALITY POINTS
Cluster 2	HIGH CLASS	PLATINUM CARD	18.49	37000	1.INCREASE CREDIT LIMMIT 2.SPECIAL GIFTS CARDS 3.LOW INTEREST LOANS

- The cluster 0 is 'MIDDLE CLASS' people has low credit limit of 28500 rupees and they are classified as 'GOLD card holders', these customers has average spending amount 11.85. These customers are paying the minimum payment and they are maintaining low current balance, high minimum payment amount and their credit amount in advance payments is less.
- So, for Cluster 0: We can give Promotional strategies like 'FREE COUPONS','EXTRA REWARD POINTS',’ZERO ANNUAL CHARGES’ **to increase their spending amount.**

- The cluster 1 is 'UPPER MIDDLE CLASS' people has average credit limit of 32500 rupees : they are classified as 'DIAMOND card holders', these customers has average spending amount 14.43.These customers are giving income to the bank by collecting the interest from them and they are maintaining moderate current balance less amount on minimum payment and their credit amount in advance payments is moderate.
- So for Cluster 1 : We can give Promotional strategies like 'LOW INTEREST RATE','EXTRA CASHBACK',’INCREASE LOYALITY POINTS’ to **increase their minimum payment amount.**

- The cluster 2 is 'HIGH CLASS' people has more credit limit of 37000 rupees: they are classified as 'PLATINUM card holders', these customers has average spending amount 18.49. These customers are called max payers because they are doing advance payments, maintaining high current balance and their spending amount is also more.
- So for Cluster 2 : We can give Promotional strategies like 'INCREASE CREDIT LIMIT','SPECIAL CREDIT CARDS','LOW INTEREST LOANS' **to reduce their advance/full payment**.

PROBLEM 2: CART-RF-ANN

Data Insights and EDA:

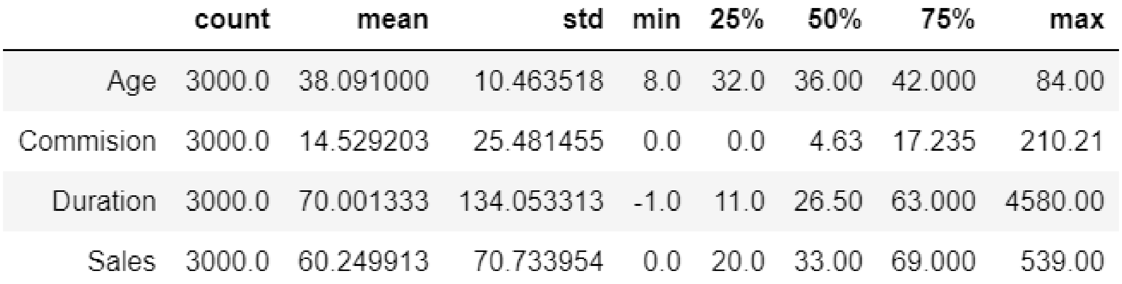
The dataset provided to us is stored as “insurance_part2_data.csv” which contains data of 3000 customers and 10 variables namely:

Age	Age of insured
Agency_Code	Code of tour firm
Type	Type of tour insurance firms
Claimed	Target: Claim Status
Commission	The commission received for tour insurance firm
Channel	Distribution channel of tour insurance agencies
Duration	Duration of the tour
Sales	Amount of sales of tour insurance policies
Product Name	Name of the tour insurance products
Destination	Destination of the tour

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it.

Description of dataset:

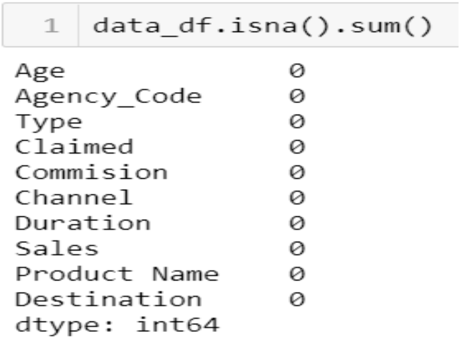
The below table which shows the count, average, maximum, minimum values for the variables like spending, advance payment, probability of full payment, current balance, credit limit, minimum payment amount, maximum spent in single shopping.



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

Checking for Missing Values:

The dataset does not contain any Null values. The total no of missing /Null values is 0.

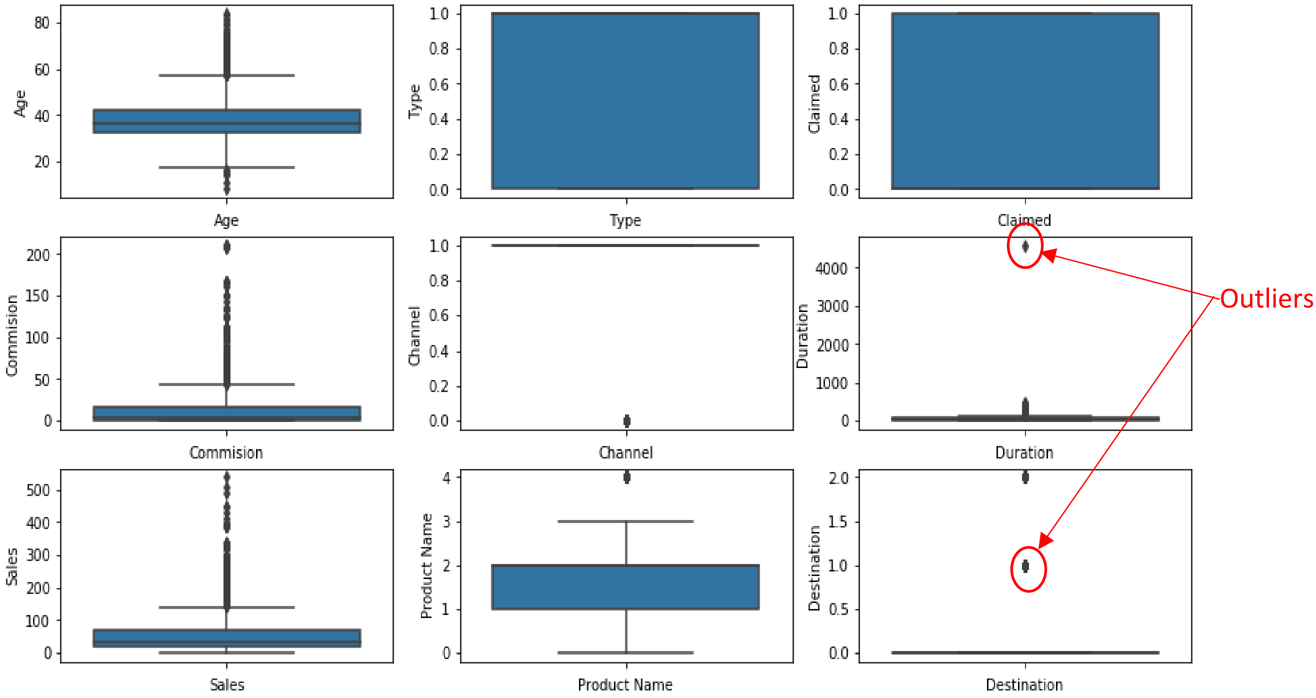


```
data_df.isna().sum()
```

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

Univariate Analysis:

Using the Boxplot in a dataset we can able to find the outliers, spread of values, median, range, etc., (outliers are the extreme values present in the dataset)

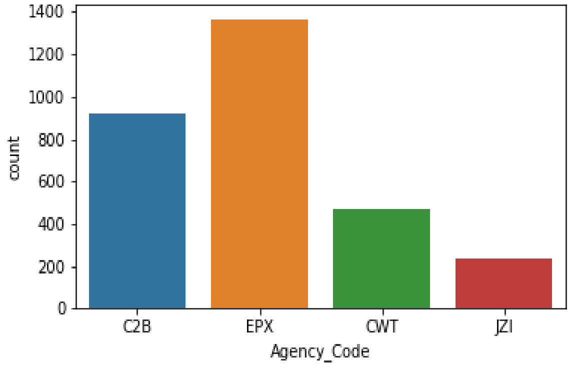


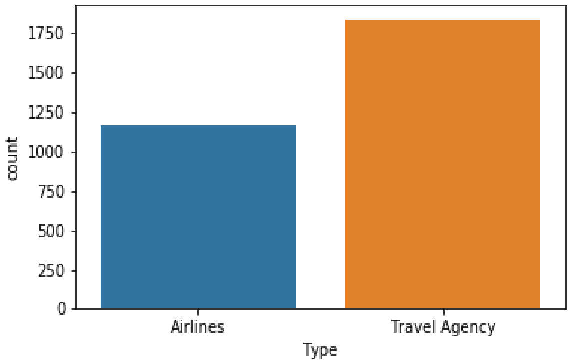
Inference from boxplot:

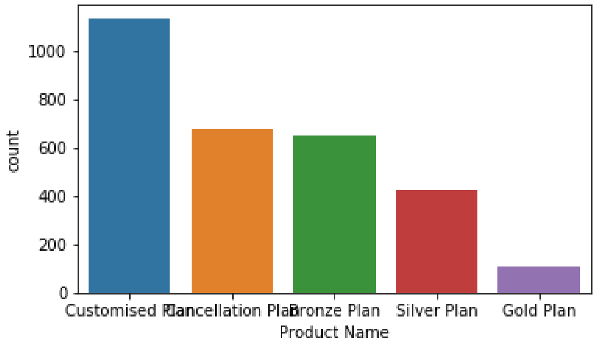
From the above Boxplots for all the variables, we can conclude that outliers are present in the variables like Age, Commission, Sales, Product Name, Destination, Duration

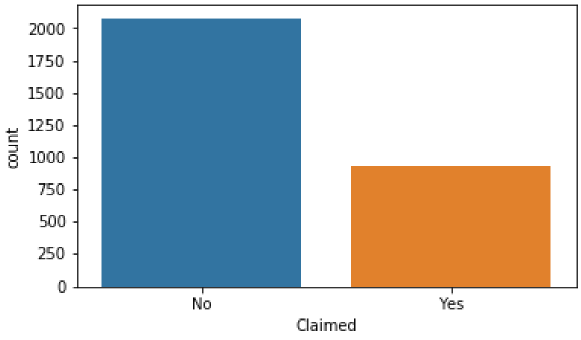
There are only a very few outliers are present in the dataset, so significantly it may not affect any impact on our dataset, so is no need to do outlier treatment.

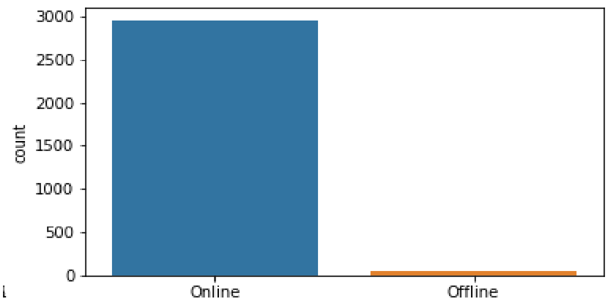
Count Plot:

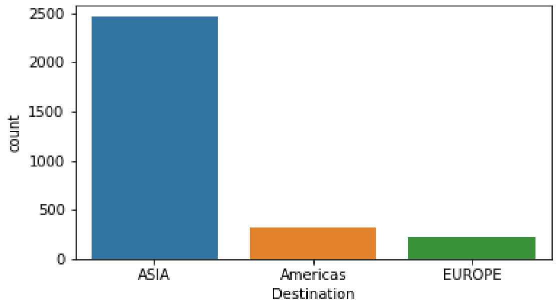












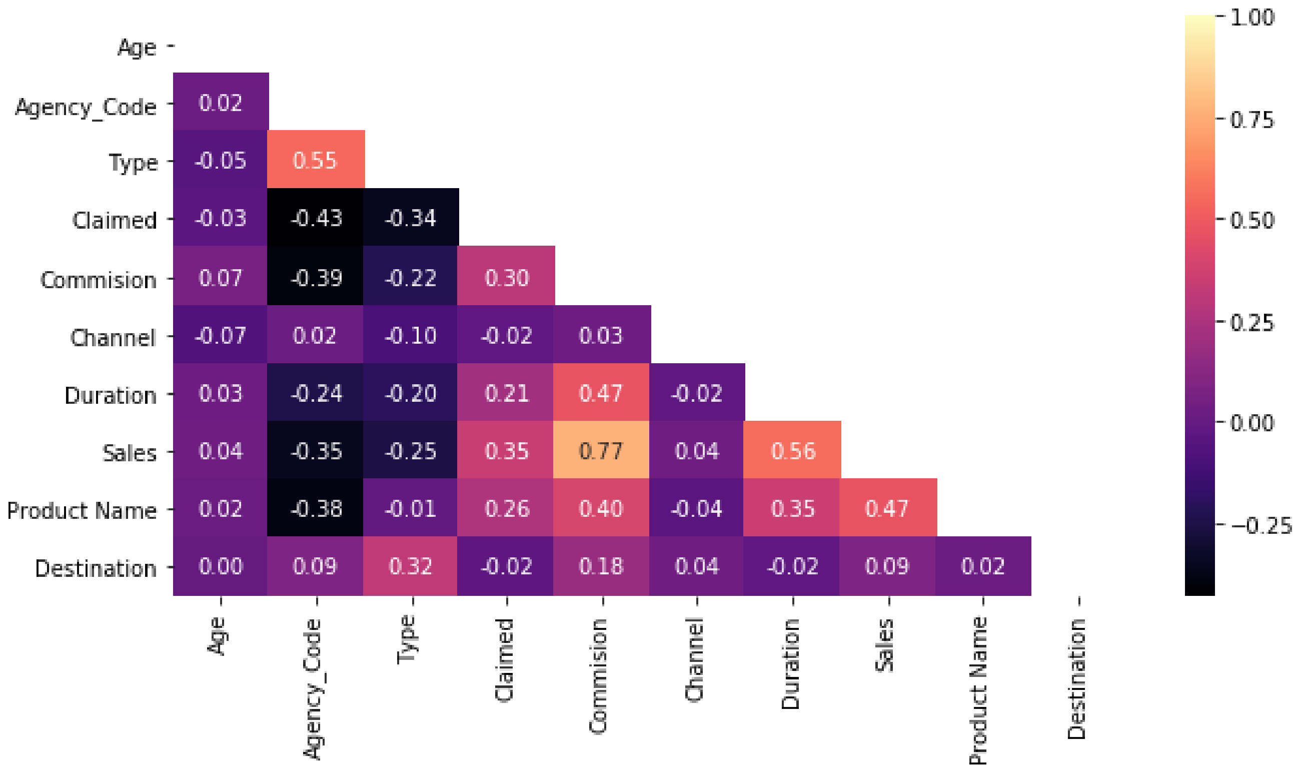
Inference from Countplot:

- The travel Agency has most count when compared to airlines
- The Insurance claimed has nearly 50% of the numbers when compared to those who are unclaimed.
- The Destination Asia has made more number of insurance when compared to others.
- For transaction Online mode is more preferable when compared to offline mode

Multivariate Analysis

Heat Map

The Heat Map shows the relationship between different variables in our dataset. This graph can help us to check for any correlations between different variables.

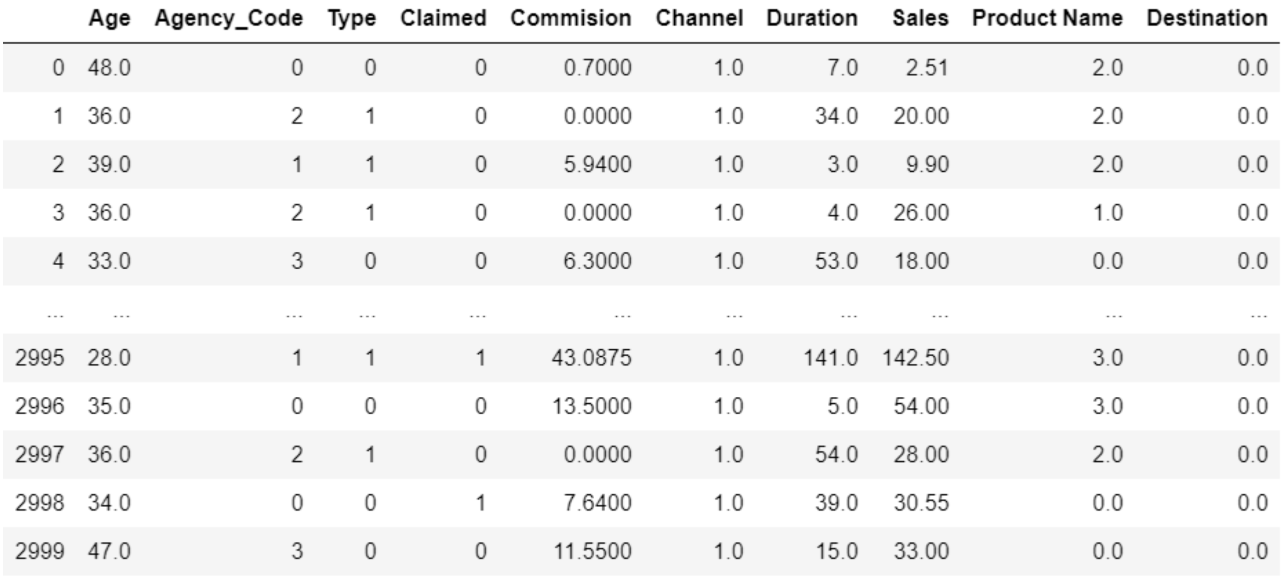


Inference from heatmap:

We can see that except Commission and Sales there is **no** highly positive correlation between the variables.

- The variables like sales and commission are highly positive correlated.
- The Agency code and Type are positively correlated with eachother.
- The variables like duration and commission, sales and duration, sales and product name are weakly positive correlated.
- The variables like claimed, commission, channel, duration, sales are having weak negative correlation with type.

Data Transformation



	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48.0	0	0	0	0.7000	1.0	7.0	2.51	2.0	0.0
1	36.0	2	1	0	0.0000	1.0	34.0	20.00	2.0	0.0
2	39.0	1	1	0	5.9400	1.0	3.0	9.90	2.0	0.0
3	36.0	2	1	0	0.0000	1.0	4.0	26.00	1.0	0.0
4	33.0	3	0	0	6.3000	1.0	53.0	18.00	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...
2995	28.0	1	1	1	43.0875	1.0	141.0	142.50	3.0	0.0
2996	35.0	0	0	0	13.5000	1.0	5.0	54.00	3.0	0.0
2997	36.0	2	1	0	0.0000	1.0	54.0	28.00	2.0	0.0
2998	34.0	0	0	1	7.6400	1.0	39.0	30.55	0.0	0.0
2999	47.0	3	0	0	11.5500	1.0	15.0	33.00	0.0	0.0

In this dataset, “Agency Code” Is the column which cannot be useful for our analysis. Hence, we will be dropping this column and transforming the categorical variable to numerical labels.

2.2. To split the data into test and train, build classification model CART, Random Forest and Artificial Neural Network.

Splitting Dataset in Train and Test Data (70:30)

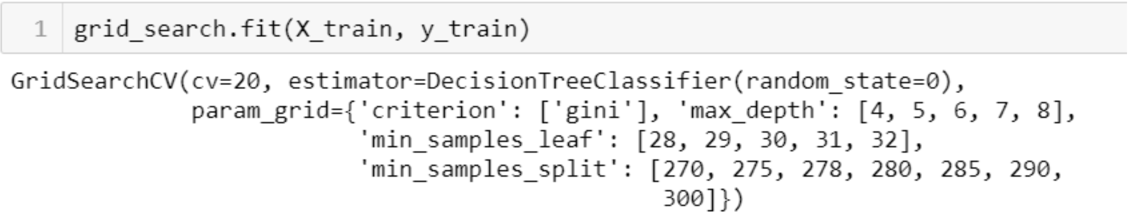
For building the models we will now have to split the dataset into Training and Testing data with the ratio of 70:30.

CART Model

Classification and Regression Trees (CART) are a type of Decision trees used in Data mining. It is a type of Supervised Learning Technique where the predicted outcome is either a discrete or class (classification) of the dataset or the outcome is of continuous or numerical in nature(regression).

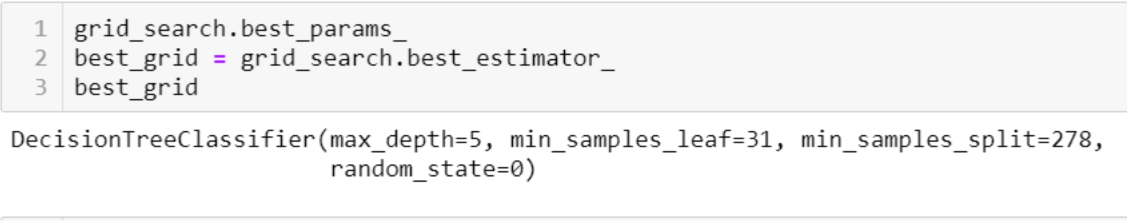
Using the Train Dataset (X Train) we will be creating a CART model and then further testing the model on Test Dataset (X Test)

For creating the CART Model two packages were imported namely, “DecisionTreeClassifier” and “tree” from sklearn.



```
grid_search.fit(X_train, y_train)
```

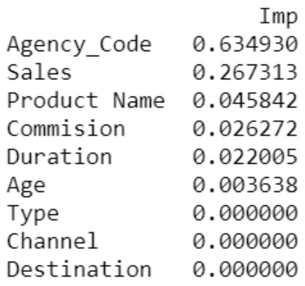
```
GridSearchCV(cv=20, estimator=DecisionTreeClassifier(random_state=0),
             param_grid={'criterion': ['gini'], 'max_depth': [4, 5, 6, 7, 8],
                         'min_samples_leaf': [28, 29, 30, 31, 32],
                         'min_samples_split': [270, 275, 278, 280, 285, 290,
                                               300]})
```



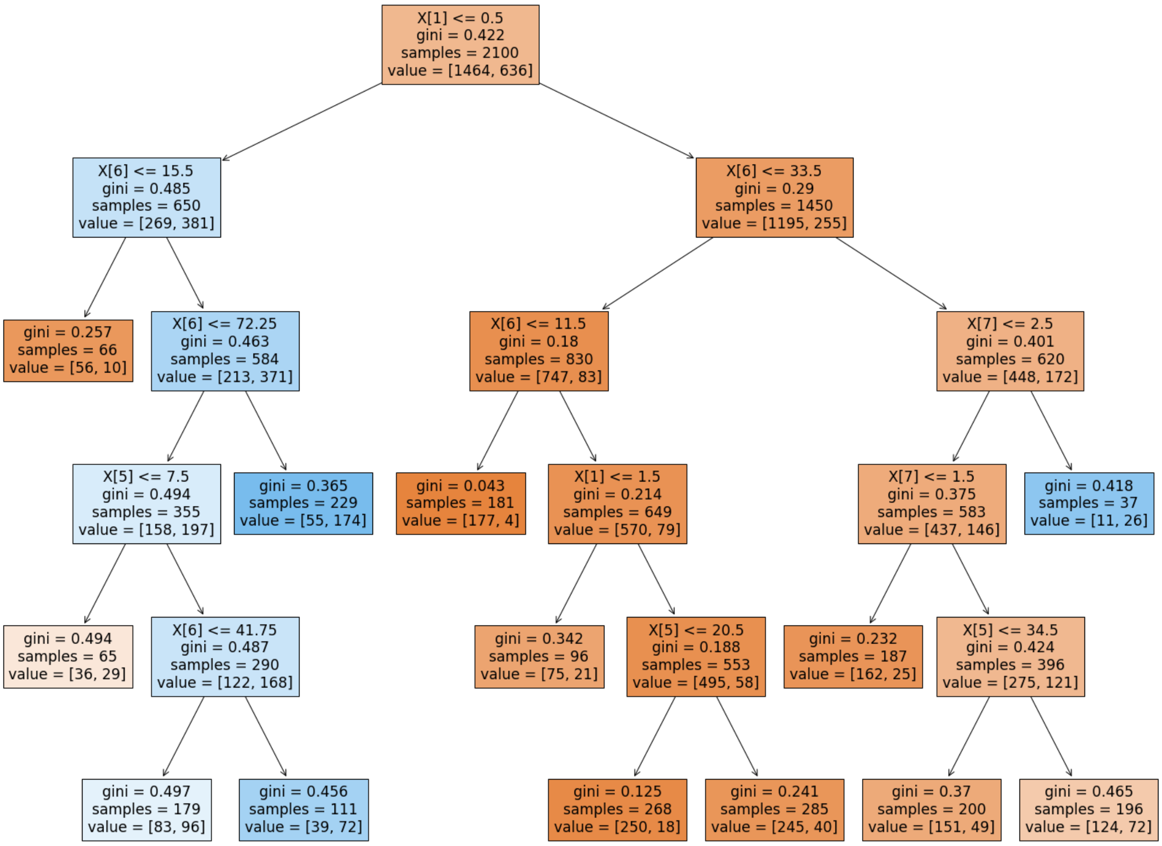
```
grid_search.best_params_
best_grid = grid_search.best_estimator_
best_grid
```

```
DecisionTreeClassifier(max_depth=5, min_samples_leaf=31, min_samples_split=278,
                       random_state=0)
```

With the help of DecisonTreeClassifier we will create a decision tree model namely, dt_model and using the “gini” criteria we will fit the train data into this model. Below are the variable importance values or the feature importance to build the tree.



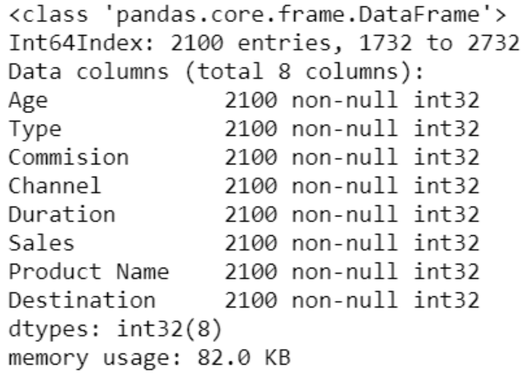
```
                   Imp
Agency_Code   0.634930
Sales         0.267313
Product Name  0.045842
Commision     0.026272
Duration      0.022005
Age           0.003638
Type          0.000000
Channel       0.000000
Destination   0.000000
```



Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees.

For this random forest algorithim the variable datatype has to be converted to integer, it does not works with object datatype.



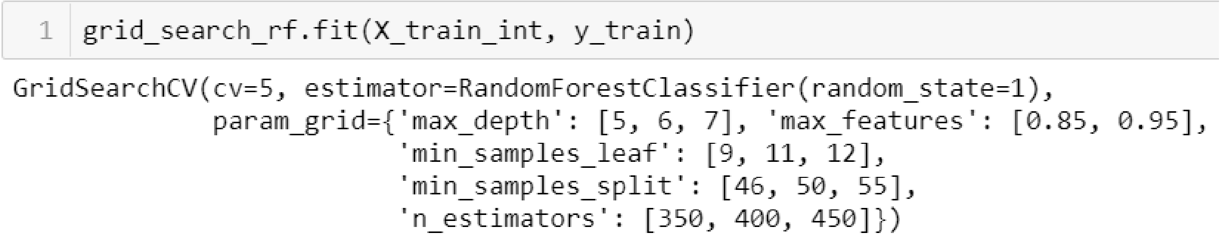
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2100 entries, 1732 to 2732
Data columns (total 8 columns):
Age             2100 non-null int32
Type            2100 non-null int32
Commision       2100 non-null int32
Channel         2100 non-null int32
Duration        2100 non-null int32
Sales           2100 non-null int32
Product Name    2100 non-null int32
Destination     2100 non-null int32
dtypes: int32(8)
memory usage: 82.0 KB
```

Using the Train Dataset(X_train) we will be creating a Random Forest model and then further testing the model on Test Dataset(X_test)

For creating the Random Forest, the package "RandomForestClassifier" is imported from sklearn library.

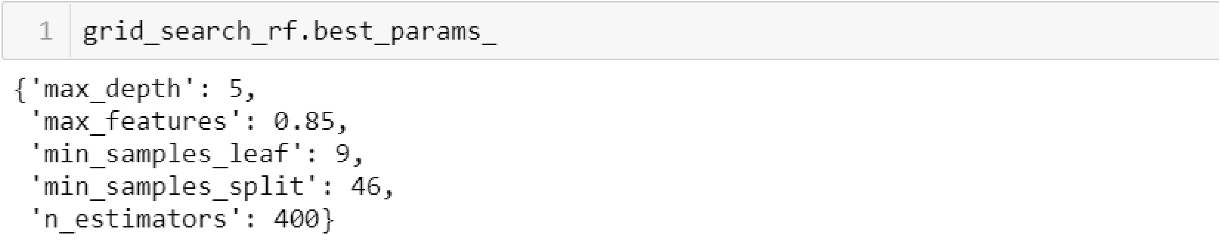
Using the GridSearchCV package from sklearn.model_selection we will identify the best parameters to build a Random Forest namely, rfcl. Hence, doing a few iterations with the values we got the best parameters to build the RF Model which are as follows

BEST PARAMETER:



```
grid_search_rf.fit(X_train_int, y_train)
```

```
GridSearchCV(cv=5, estimator=RandomForestClassifier(random_state=1),
             param_grid={'max_depth': [5, 6, 7], 'max_features': [0.85, 0.95],
                         'min_samples_leaf': [9, 11, 12],
                         'min_samples_split': [46, 50, 55],
                         'n_estimators': [350, 400, 450]})
```



```
grid_search_rf.best_params_
```

```
{'max_depth': 5,
 'max_features': 0.85,
 'min_samples_leaf': 9,
 'min_samples_split': 46,
 'n_estimators': 400}
```

Artificial Neural Network (ANN)

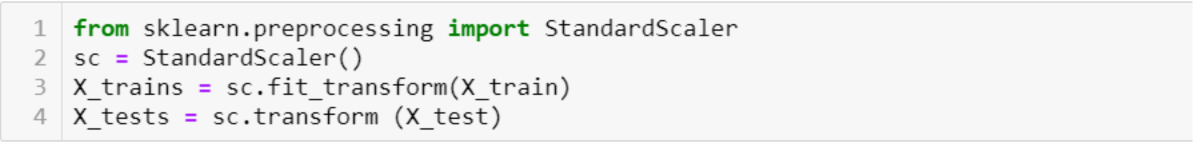
ANNs are composed of artificial neurons which are conceptually derived from biological neurons. Each artificial neuron has inputs and produce a single output which can be sent to multiple other neurons.

The inputs can be the feature values of a sample of external data, such as images or documents, or they can be the outputs of other neurons. The outputs of the final output neurons of the neural net accomplish the task, such as recognizing an object in an image.

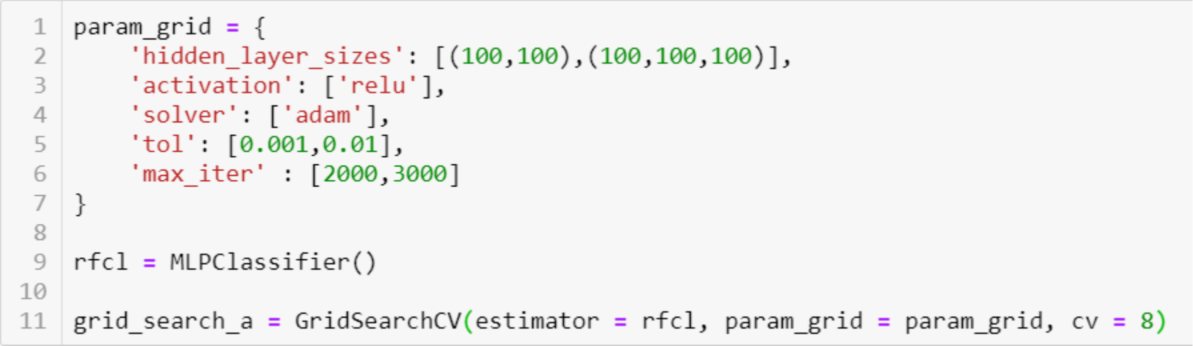
To find the output of the neuron, first we take the weighted sum of all the inputs, weighted by the weights of the connections from the inputs to the neuron. We add a bias term to this sum. This weighted sum is sometimes called the activation.

This weighted sum is then passed through a (usually nonlinear) activation function to produce the output. The initial inputs are external data, such as images and documents. The ultimate outputs accomplish the task, such as recognizing an object in an image

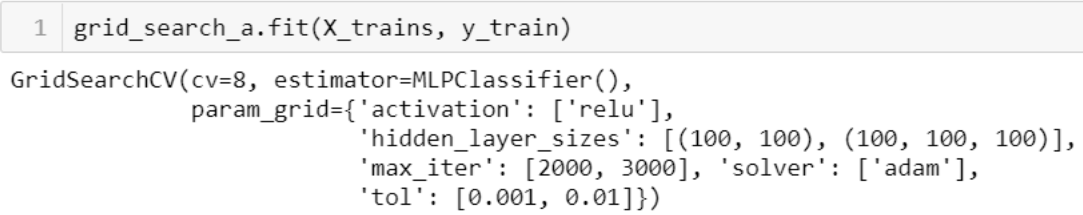
Using the train dataset(X_train) and test dataset(X_test) we will be creating a Neural Network using MLPClassifier from sklearn.metrics.



```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_trains = sc.fit_transform(X_train)
X_tests = sc.transform (X_test)
```

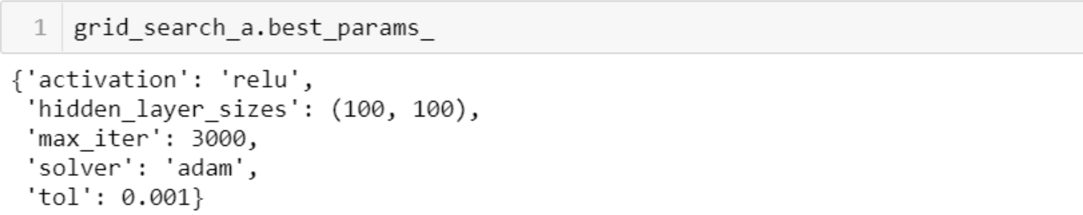


```
param_grid = {
    'hidden_layer_sizes': [(100,100),(100,100,100)],
    'activation': ['relu'],
    'solver': ['adam'],
    'tol': [0.001,0.01],
    'max_iter' : [2000,3000]
}

rfcl = MLPClassifier()

grid_search_a = GridSearchCV(estimator = rfcl, param_grid = param_grid, cv = 8)
```



```
grid_search_a.fit(X_trains, y_train)
```

```
GridSearchCV(cv=8, estimator=MLPClassifier(),
             param_grid={'activation': ['relu'],
                         'hidden_layer_sizes': [(100, 100), (100, 100, 100)],
                         'max_iter': [2000, 3000], 'solver': ['adam'],
                         'tol': [0.001, 0.01]})
```



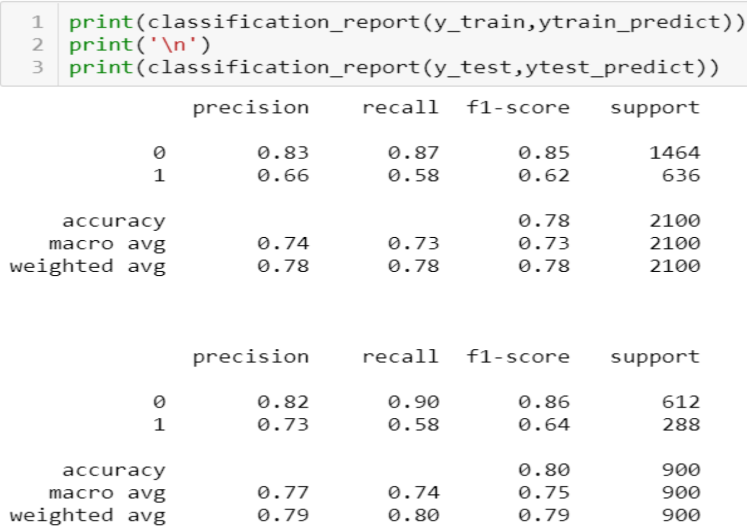
```
grid_search_a.best_params_
```

```
{'activation': 'relu',
 'hidden_layer_sizes': (100, 100),
 'max_iter': 3000,
 'solver': 'adam',
 'tol': 0.001}
```

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

CART Model

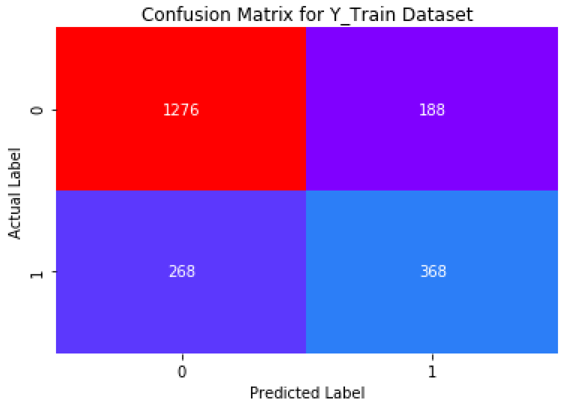
Classification Report for training and testing dataset:

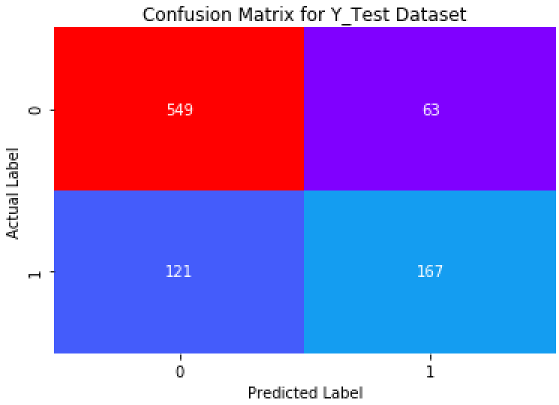


```
print(classification_report(y_train,ytrain_predict))
print('\n')
print(classification_report(y_test,ytest_predict))
```

```
              precision    recall  f1-score   support

           0       0.83      0.87      0.85      1464
           1       0.66      0.58      0.62       636

    accuracy                           0.78      2100
   macro avg       0.74      0.73      0.73      2100
weighted avg       0.78      0.78      0.78      2100


              precision    recall  f1-score   support

           0       0.82      0.90      0.86       612
           1       0.73      0.58      0.64       288

    accuracy                           0.80       900
   macro avg       0.77      0.74      0.75       900
weighted avg       0.79      0.80      0.79       900
```

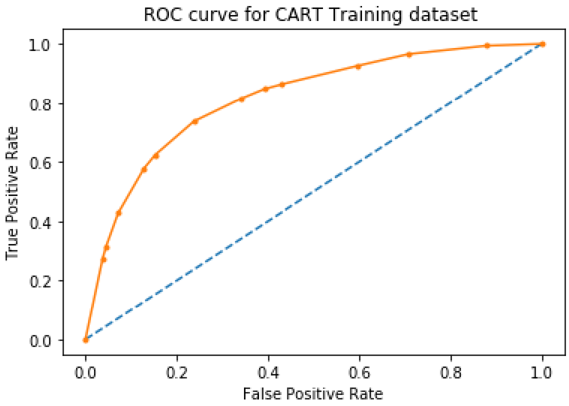
Confusion Matrix

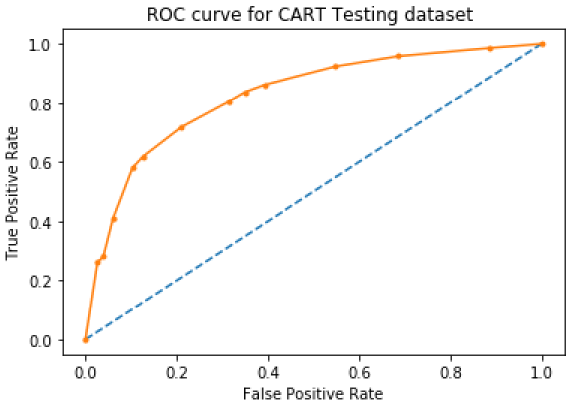




-

ROC_AUC Score and ROC Curve





-

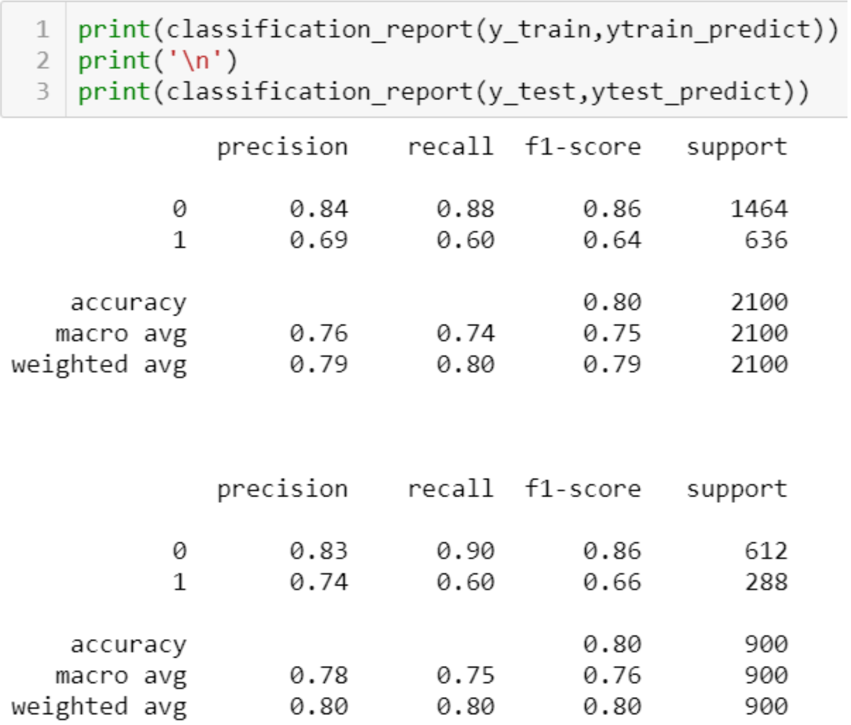
Model Score

- The Decision tree CART model Training dataset AUC score is 0.816
- The Decision tree CART model Testing dataset AUC score is 0.826

Random Forest Model

The classification report, confusion matrix, AUC-ROC score for random forest model is described below

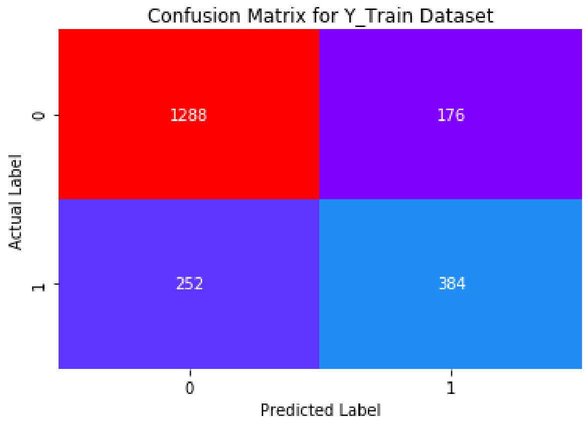
Classification Report

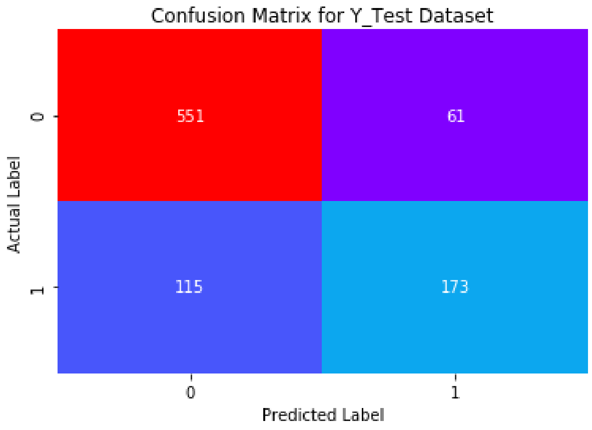


```
print(classification_report(y_train,ytrain_predict))
print('\n')
print(classification_report(y_test,ytest_predict))
```

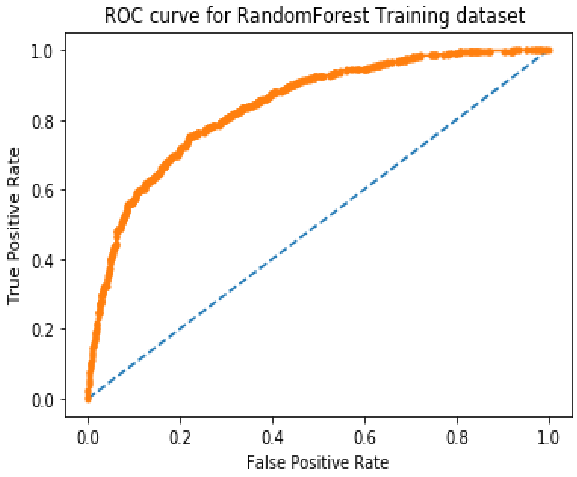
```
              precision    recall  f1-score   support

           0       0.84      0.88      0.86      1464
           1       0.69      0.60      0.64       636

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



              precision    recall  f1-score   support

           0       0.83      0.90      0.86       612
           1       0.74      0.60      0.66       288

    accuracy                           0.80       900
   macro avg       0.78      0.75      0.76       900
weighted avg       0.80      0.80      0.80       900
```

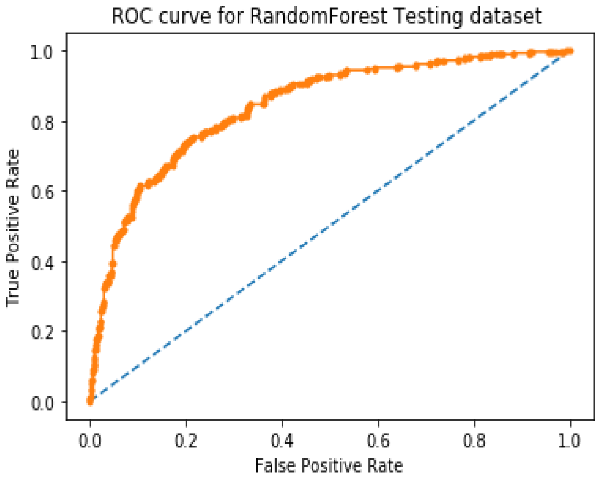
Confusion Matrix





AUC_ROC Score and ROC Curve



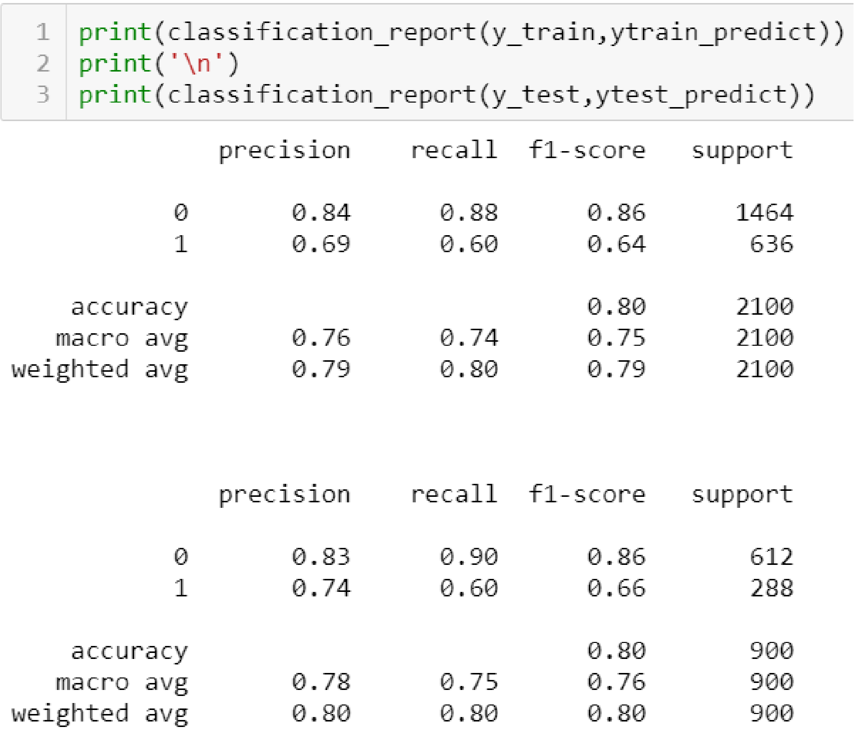


Model Score

- The Random Forest model Training dataset AUC score is 0.841
- The Random Forest model Testing dataset AUC score is 0.842

Artificial Neural Network Model

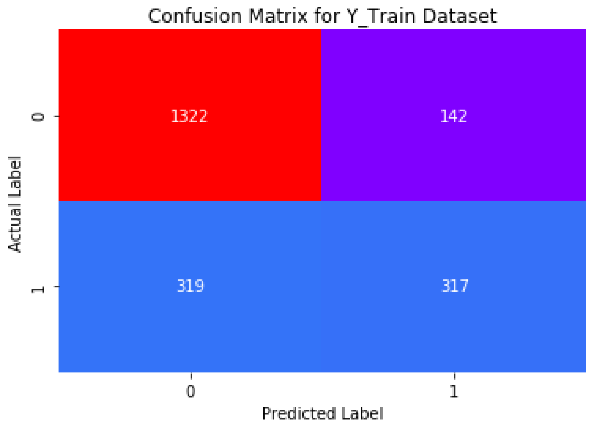
Classification Report

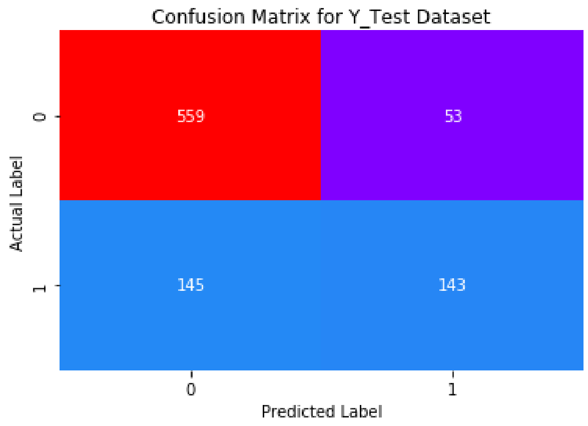


```
print(classification_report(y_train,ytrain_predict))
print('\n')
print(classification_report(y_test,ytest_predict))
```

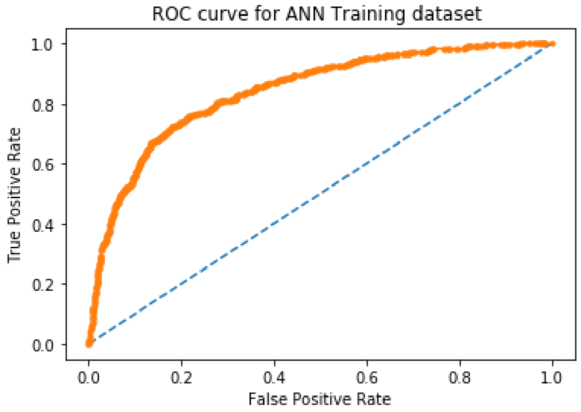
```
              precision    recall  f1-score   support

           0       0.84      0.88      0.86      1464
           1       0.69      0.60      0.64       636

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


              precision    recall  f1-score   support

           0       0.83      0.90      0.86       612
           1       0.74      0.60      0.66       288

    accuracy                           0.80       900
   macro avg       0.78      0.75      0.76       900
weighted avg       0.80      0.80      0.80       900
```

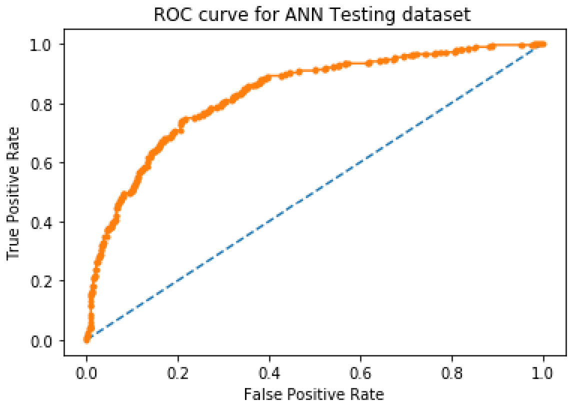
Confusion Matrix





AUC_ROC Score and ROC Curve

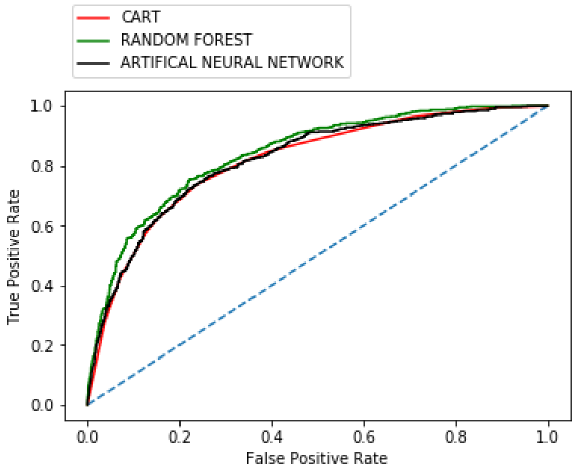


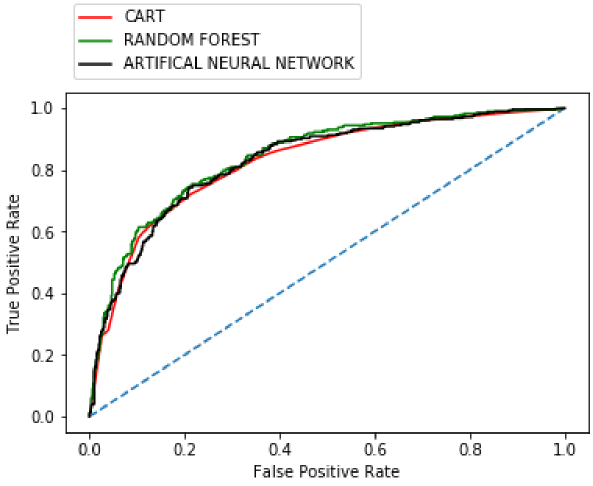


Model Score

- The Artificial neural network model Training dataset AUC score is 0.820
- The Artificial neural network model Testing dataset AUC score is 0.831

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





Since we are building a model to predict claim status as claimed or not claimed, for our purposes, we will be more interested in correctly classifying 1 (insurance is claimed) than 0 (insurance is not claimed).

Actually, if the insurance is claimed and our model is incorrectly predicted as insurance is not claimed, in this situation**, the cost and other impacts** will cost severe effects for the organization than when we incorrectly predicted as someone who is actually not claimed, as claimed the insurance.

From the above all three model, we looking at the Accuracy, Precision, Recall, F1score and AUC score for the training and testing data and we are looking especially in predicting **Class 1.**Comparison of all the performance evaluators for the three models are given in the following table. We are using Accuracy, Precision, Recall, F1 Score and AUC Score for our evaluation.

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**AUC Score**
		CART Model			
Train Data	0.78	0.66	0.58	0.62	**0.816**
Test Data	0.80	0.76	0.58	0.64	**0.826**
		Random Forest			
Train Data	0.80	0.69	0.60	0.64	**0.841**
Test Data	0.80	0.74	0.60	0.66	**0.842**
		Neural Network			
Train Data	0.78	0.69	0.50	0.58	0.820
Test Data	0.78	0.73	0.50	0.59	0.831

From the above table, comparing the model performance evaluators for the three models it is quite clear that the **Random Forest Model** is performing well as compared to the other two as it has high Recall,Precision,F1 score for both training and testing data and although the AUC Score is the same for all the three models for training and testing data Random Forest Model has great score outcome.

Choosing Random Forest Model is the best option in this case as it will exhibit very less variance as compared to a single decision tree or a multi – layered Neural Network.

2.5 Inference: Based on the whole Analysis, what are the business insights and recommendations

For the business problem of an Insurance firm providing Tour Insurance, we have attempted to make a few Data Models for predictions of probabilities. The models that are attempted are namely, CART or Classification and Regression Trees, Random Forest and Artificial Neural Network (MLP). The three models are then evaluated on training and testing datasets and their model performance scores are calculated.

Insights and Recommendation:

- From the data, we can find that almost 90% of insurance is done by online channel. The online experiences are well encouraged by the benefitted customers, leading to an increase in sales, which subsequently raise the profits.
- Try to make Online/Offline Ad campaign other than Asian countries, who are not well aware of our insurance policies.
- we 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 booking airline tickets or plans, cross sell the insurance based on the claim data pattern.
- More sales happened 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.
- I strongly recommend to collect more real time unstructured data and past data if possible. This is to understood by looking at the insurance data by drawing relations between different variables such as day of the incident, time, age group, and associating it with other external information such as location, behaviour patterns, weather information, airline/vehicle types, etc