DATA MINING
PROJECT -4 REPORT

Piyush Kumar Singh
PGP – DSBA Online
May-21 Batch

Date: 20/09/2021

Table of Contents

List of Figures

List of Tables

Problem 1: Clustering

Executive Summary

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.

Exploratory Data Analysis

Dataset Sample

	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

Table 1.

Dataset Info

```
<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
```

Table 2.

- The given dataset has 7 columns each one is continuous (float-type) data
- Dataset has 210 rows and 7 columns
- No need to for imputation as all values are in correct format

Checking for missing values in dataset

```
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
```

Table 3.

- No missing values in any of the columns in dataset
- No duplicate values found in dataset

Summary of Dataset

	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

Table 4.

- We can see that for column probability_of_full_payment max value is 0.9 while for rest columns it values are present in both single and double digits. So scaling needs to be performed before performing clustering technique on the dataset.

Checking for outliers

- Since outliers are present in the 2 columns in dataset we treat them.

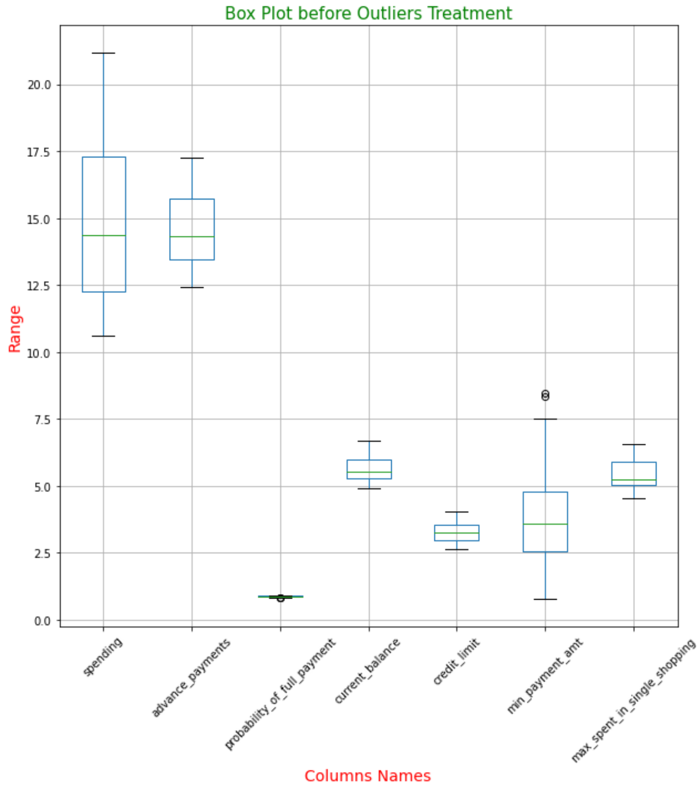


Fig. 1

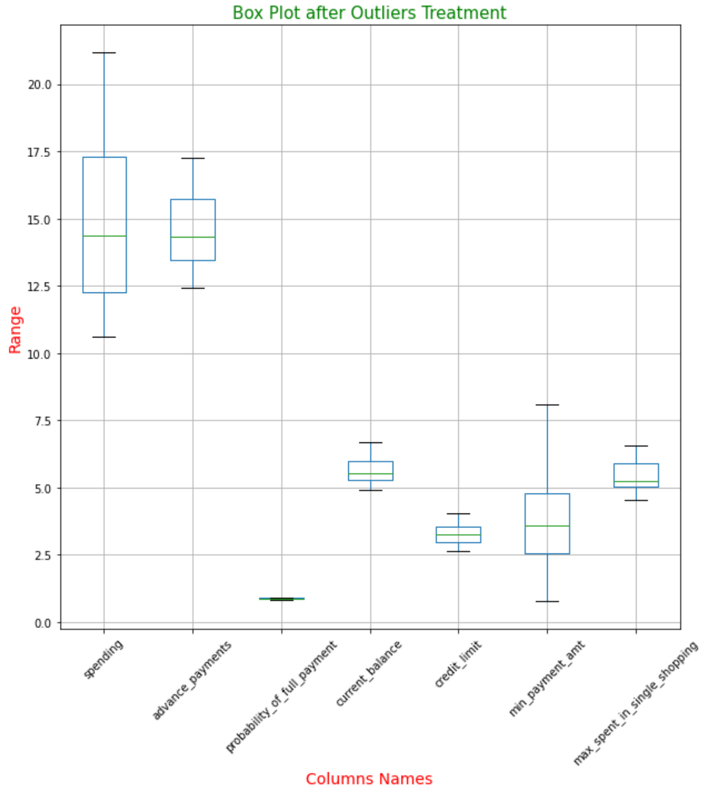


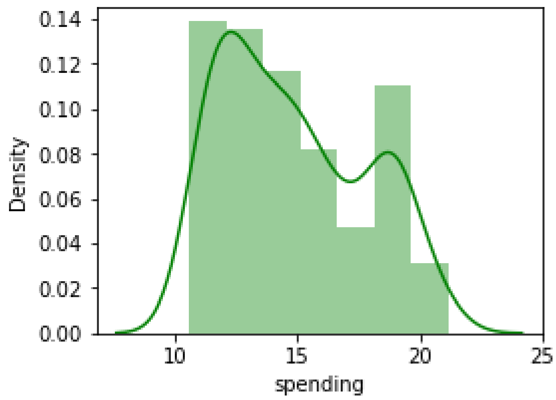
Fig. 2

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

Univariate Analysis

We perform the univariate analysis on the data set and display distplot to check distribution of each column and use boxplot to check for outliers if any.

spending



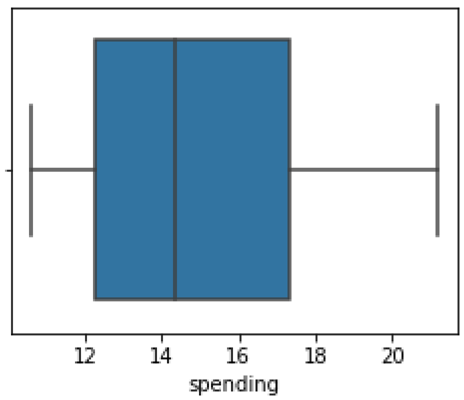
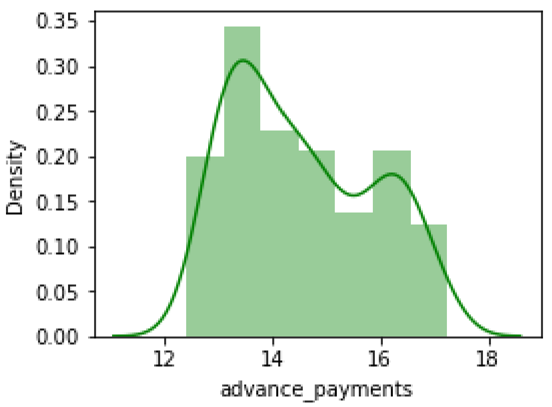


Fig.3 - Distplot and boxplot of spending

- The mean number of spending is 14.84 whereas the SD is 2.90
- The maximum value is 21.18 and min value is 10.59
- The distribution is highly skewed towards right.
- No outliers are present in spending

advance_payments



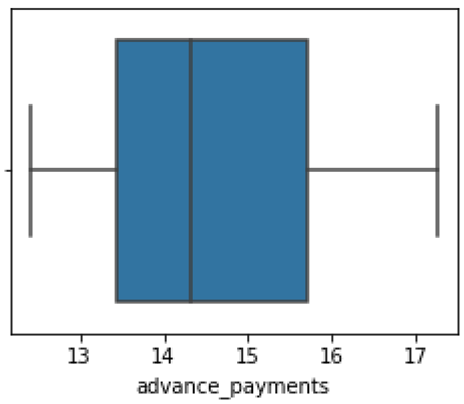


Fig.4 - Distplot and boxplot of advance_payments

- The mean number of advance_payments is 14.55 whereas the SD is 1.30
- The maximum value is 17.25 and min value is 12.41
- The distribution is highly skewed towards right.
- No outliers are present in advance_payments

probability_of_full_payments

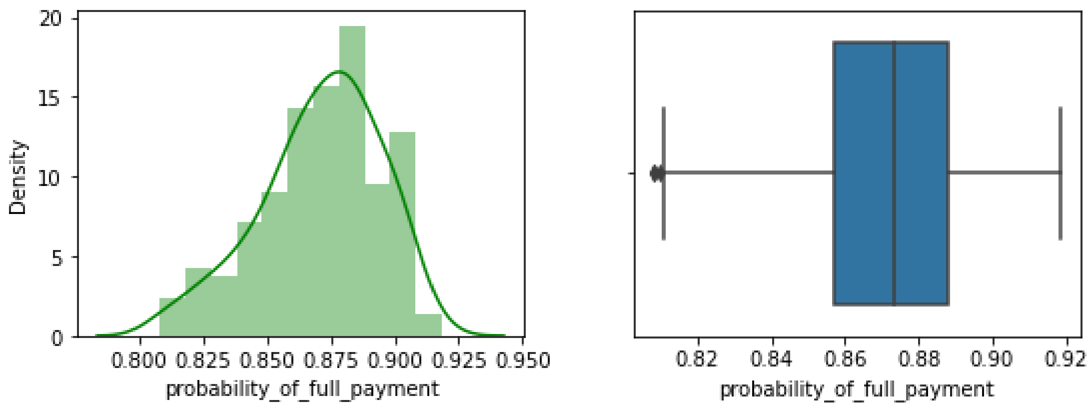


Fig.5 - Distplot and boxplot of probability_of_full_payments

- The mean number of probability_of_full_payments is 0.87 whereas the SD is 0.02
- The maximum value is 0.91 and min value is 0.80
- The distribution is highly skewed towards left.
- Outliers are present in probability_of_full_payments

current_balance

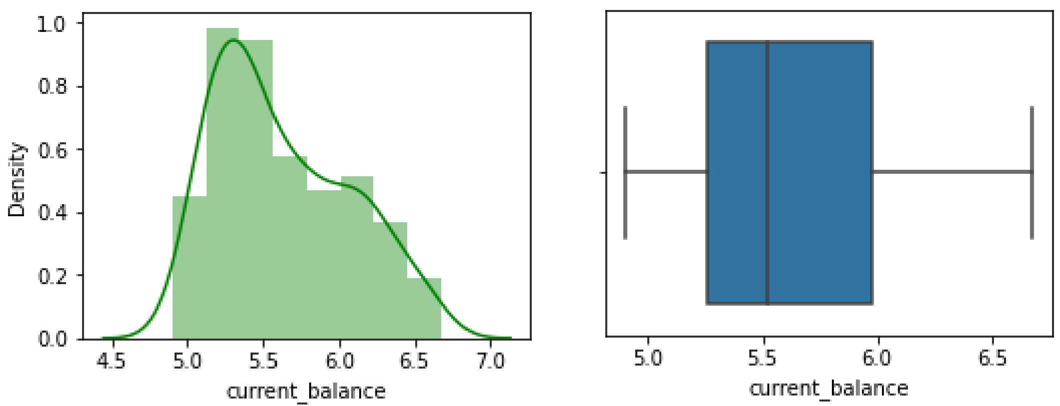


Fig.6 - Distplot and boxplot of current_balance

- The mean number of current_balance is 5.62 whereas the SD is 0.44.
- The maximum value is 6.67 and min value is 4.89
- The distribution is highly skewed towards right.
- No outliers are present in current_balance

credit_limit

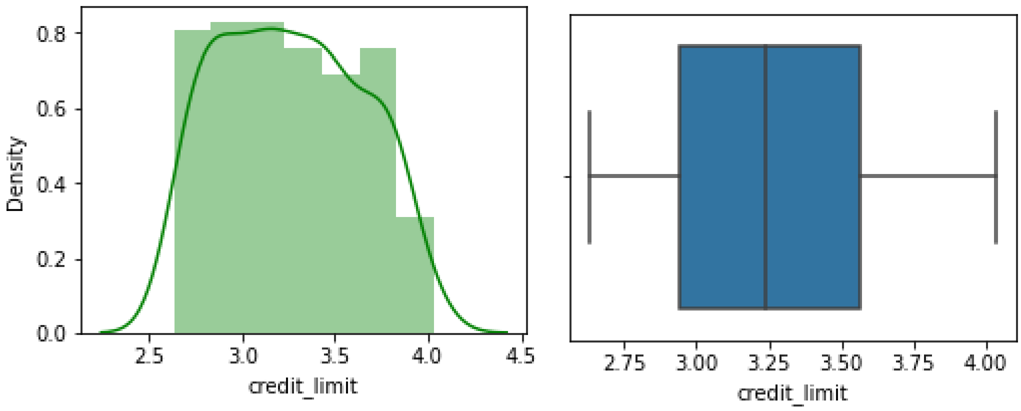


Fig.7 - Distplot and boxplot of credit_limit

- The mean number of credit_limit is 3.25 whereas the SD is 0.37
- The maximum value is 4.03 and min value is 2.63
- The distribution is somewhat normally distributed
- No outliers are present in credit_limit

min_payment_amt

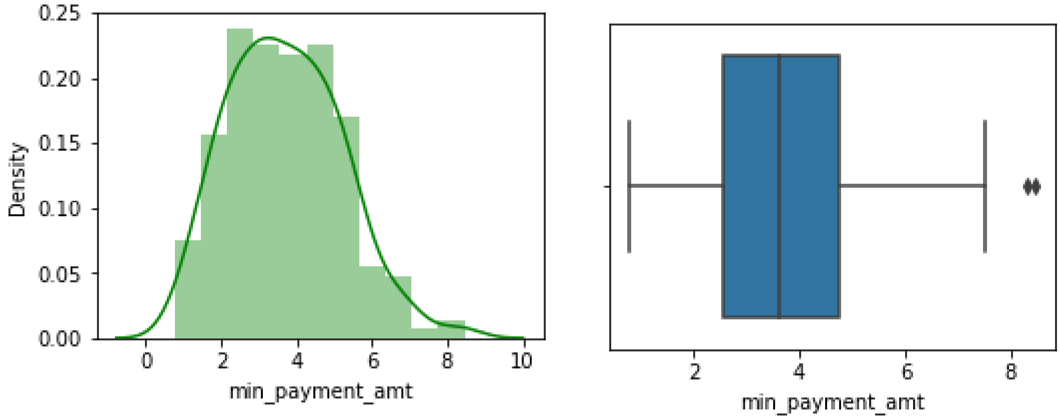


Fig.8 - Distplot and boxplot of min_payment_amt

- The mean number of min_payment_amt is 3.7 whereas the SD is 1.50
- The maximum value is 8.45 and min value is 0.76
- The distribution is normally distributed
- Outliers are present in min_payment_amt

max_spent_in_single_shopping

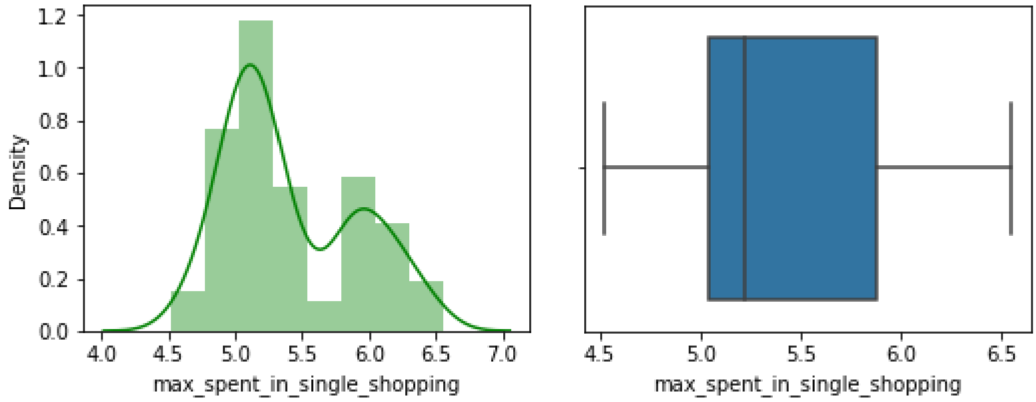


Fig.9 - Distplot and boxplot of max_spent_in_single_shopping

- The mean number of max_spent_in_single_shopping is 5.4 whereas the SD is 0.49.
- The maximum value is 6.5 and min value is 4.5
- The distribution is highly skewed towards right.
- No outliers are present in max_spent_in_single_shopping

Bi/Multivariate Analysis

For Multivariate Analysis we are using pair plot and heatmap:

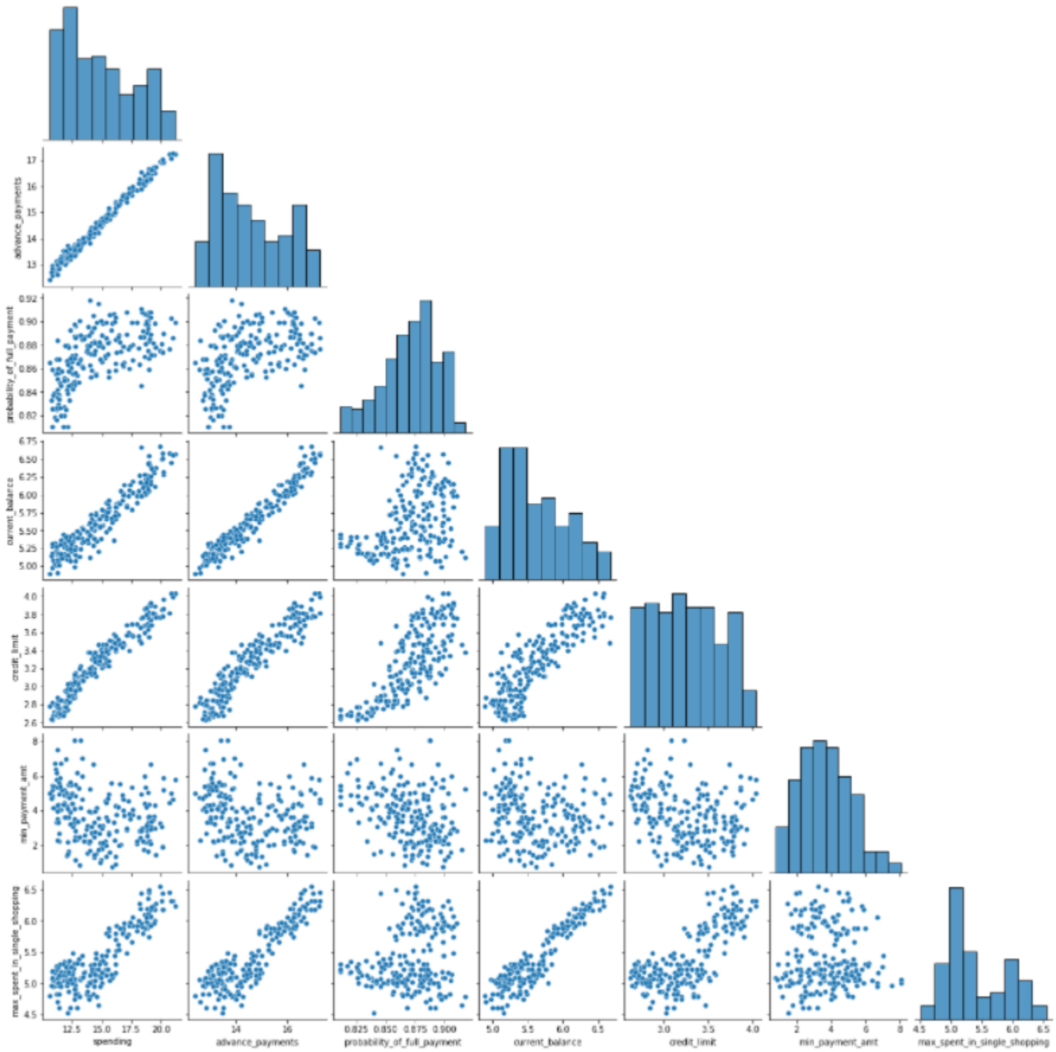


Fig.10 - Pairplot

- Since all the columns in our dataset is of numeric type, the above pair plot represents correlation between two columns in the dataset. The pair plot function in seaborne makes it very easy to generate joint scatter plots for all the columns in the data.

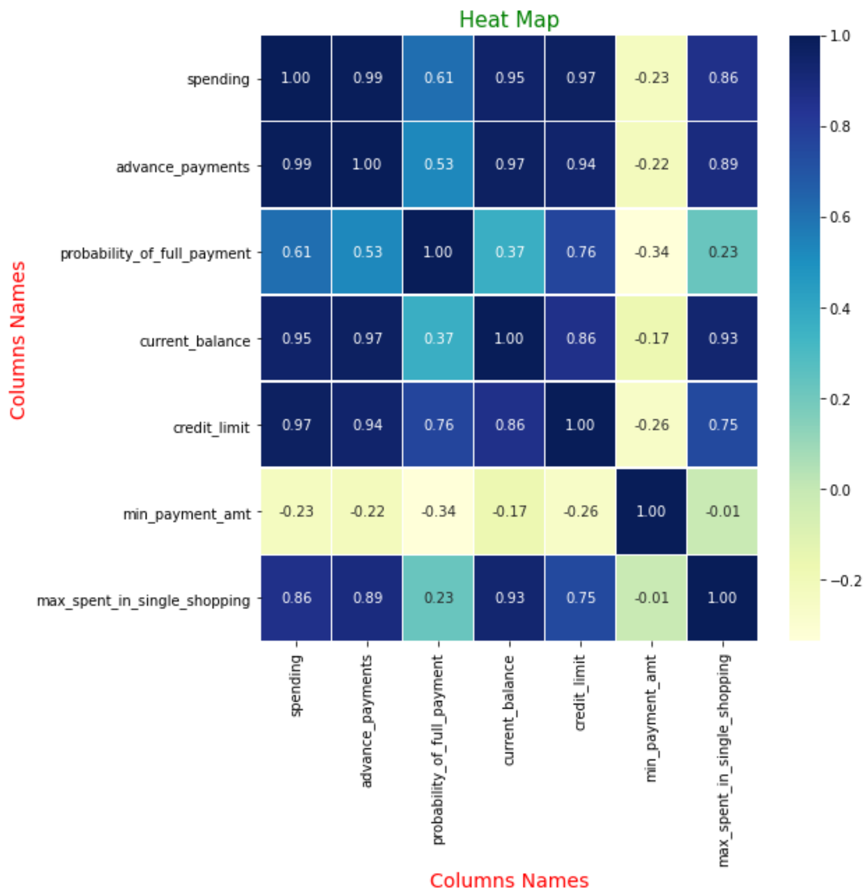


Fig.11 – Heatmap

- Degree of correlation between the columns is represented by the above heatmap, 1 being max value of correlation and 0 below no correlation between them.

<u>Pair with high correlation are as follows</u>:

- advance_payment and spending
- current_balance and spending
- credit_limit and spending
- current_balance and advance_payments
- credit_limit and advance_payments
- max_spent_in_single_shopping and current_balance

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

- The main objective of scaling is to normalize a data with a particular range. It is a step of data pre-processing which is applied to independent variables. Also, another importance of scaling is that it helps in speeding up the calculations in an algorithm.
- In our dataset, we have all numerical data type but for some columns mean value is in 0 and for others it's in 2 digit, so scaling needs to be done here.
- If we perform cluster analysis on unscaled data, differences in column value will most likely dominate each other simply because of the scale. In most practical cases, all these different variables need to be converted to one scale in order to perform meaningful analysis.
- I have used Standard Scalar 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	0.177628	2.367533	1.338579	-0.298625	2.328998
1	0.393582	0.253840	1.505071	-0.600744	0.858236	-0.242292	-0.538582
2	1.413300	1.428192	0.505234	1.401485	1.317348	-0.220832	1.509107
3	-1.384034	-1.227533	-2.571391	-0.793049	-1.639017	0.995699	-0.454961
4	1.082581	0.998364	1.198738	0.591544	1.155464	-1.092656	0.874813

Table 5. Dataset after Scaling

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

- I have performed hierarchical clustering to scaled data using linkage method – ward.

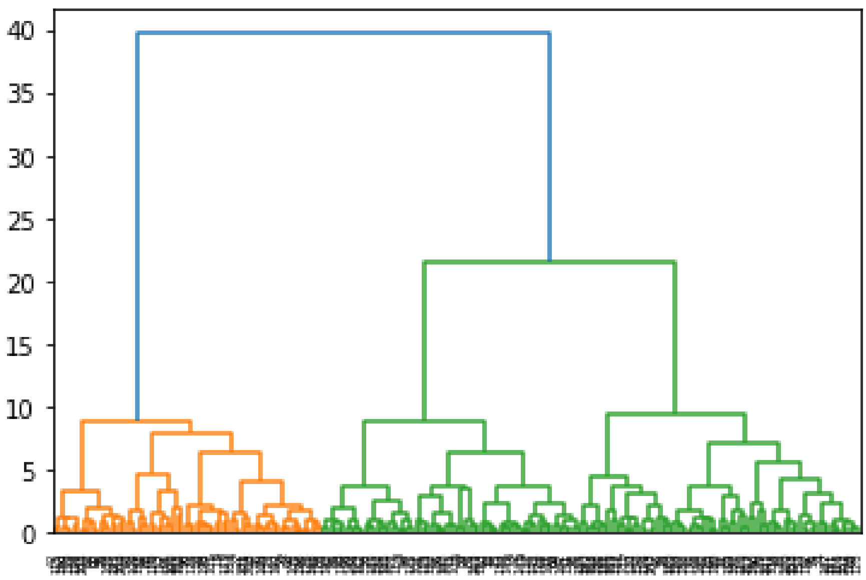


Fig.12 – Dendrogram of whole dataset

- Using truncate mode and by visualisation the dendrogram we can say optimal numbers of clusters as per dendrogram for the dataset is 3. Generally the colours shown in dendrogram represent the optimal number of cluster.

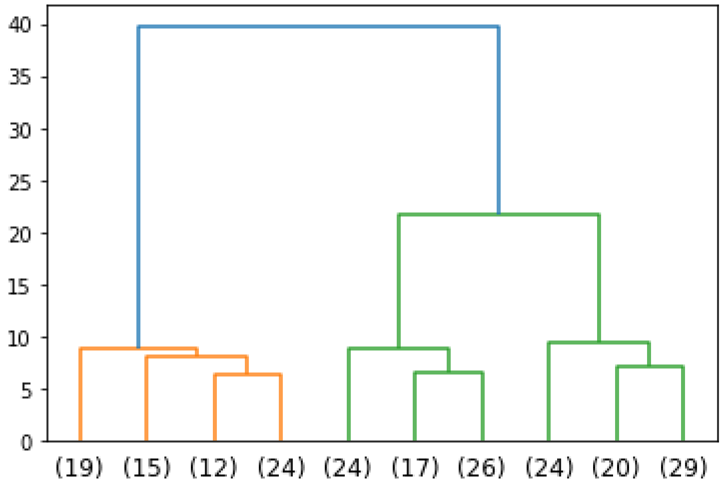


Fig.13 – Dendrogram with truncate applied

- Now using fcluster method, criterion – maxclust and using 3 as optimal number of clusters we can assign each record to its unique cluster.

```
1    70
2    67
3    73
Name: clusters_dend, dtype: int64
```

Table 6. No's of Records in each cluster

- Cluster values has also been added to dataset as clusters_dend.
- Cluster profile has been created below for the 3 clusters along with the mean values, and frequency for each attribute in dataset and presented as dataframe below.

clusters_dend	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371	70
2	11.872388	13.257015	0.848155	5.238940	2.848537	4.940302	5.122209	67
3	14.199041	14.233562	0.879190	5.478233	3.226452	2.612181	5.086178	73

Table 7. Cluster profile for Fcluster

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.

- Using K Means method on scaled data and finding WSS score for cluster values from 1 to 10. Visualizing the values in of WSS using Elbow Curve, we can say that after value 3 the drop in curve (WSS value) is not that dramatic. So optimal number of cluster as per elbow curve can be checked at 3 and 4 using Silhouette_score.

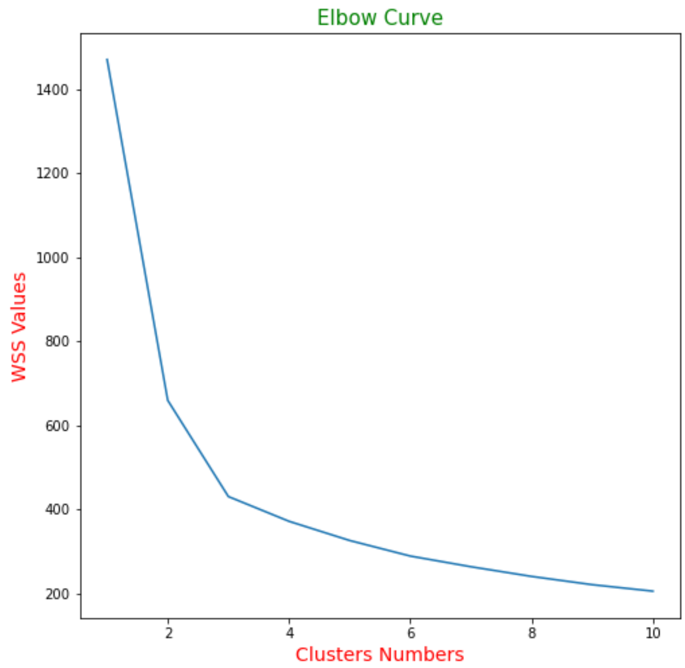


Fig.14 – Elbow Curve

- **For n_cluster value 3:**
  Silhouette_score is 0.401
  Silhouette_sample min value is 0.003

  Since the min value of Silhoutte Sample is positive we can say that all records are correctly identified to its cluster. There is no miss labelling present in our model.

- **For n_cluster value 4:**
  Silhouette_score is 0.329
  Silhouette_sample min value is -0.051

  Since the min value of Silhoutte Sample is negative some records are not correctly identified to its cluster. Model is not accurate for this cluster value.
  Also Silhouette_score is less for n_cluster value 4 than for 3.

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

- Total numbers of records in Clusters identified as per K Means are as below:

```
0    71
1    72
2    67
Name: Clust_kmeans, dtype: int64
```

Table 8. No's of record in each Cluster

- Cluster values has been added to dataset as Clust_kmeans.
- Cluster profile for 3 clusters as per K Means method along with the mean values, and frequency for each attribute in dataset is presented as data frame below.

Clust_kmeans	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
0	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803	71
1	11.856944	13.247778	0.848253	5.231750	2.849542	4.742389	5.101722	72
2	18.495373	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701	67

Table 9. Cluster profile for K Means

Business Recommendations:

The 3 clusters can be identified as low, medium and high groups based on spending.

- Low spending group represents clusters with labels as 1
- Medium spending group represents clusters with labels as 0
- High spending group represents clusters with labels as 2

- **High Spending Group**

  Average spending for this group is highest.
  Max_spent_in_single_shopping is highest for this group, so offers can provided to this group on their next purchase to promote more shopping.
  credit_limit can be increased for this group to increase spending further.

- **Medium Spending Group**

  probability_of_full payment of this group is as good as high spending group. We can target this group by offering promotional offers to push them towards high spending group.
  credit_limit can be increased for this group to increase spending.

- **Low Spending Group**

  Lowest spending is of this group.
  probability_of_full, current_balance and credit_limit payment is also lowest among all the three groups.
  Offers can be given to increase credit_limit is advance payments are increased.
  Reminder regarding advance payment can be given to this group.
  Min_payment_amt is highest for this group.
  Max_spent_in_single_shopping is comparable to that of medium spending group.

Problem 2: CART-RF-ANN

Executive Summary

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.

Exploratory Data Analysis

Dataset Sample

	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

Table .10

Dataset Info

```
<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
```

Table .11

- The given dataset has 10 columns of following types:
  float-type : 2
  int64-type : 2
  object-type : 6
  Dataset has 3000 rows and 10 columns

- No need to for imputation as all values are in correct format

Checking for missing values in dataset

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

Table .12

- No missing values in any of the columns in dataset
- 139 duplicate values found in dataset

	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

Table .13

- Since there is no unique identifier in the dataset to check whether these records below to same person or not, so I am not removing these duplicate records.

Summary of Dataset

	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

Table .14

- Dataset is mix of numeric, float and object type data.
- There is no bad data present in our dataset.
- Claimed is our target variable for analysis.
- Treating for outliers and scaling is not required for Decision tree and Random forest models as both are tolerant to it.

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

Univariate Analysis for Numeric Datatype

We perform the univariate analysis on the data set for numeric columns using boxplot to check for outliers.

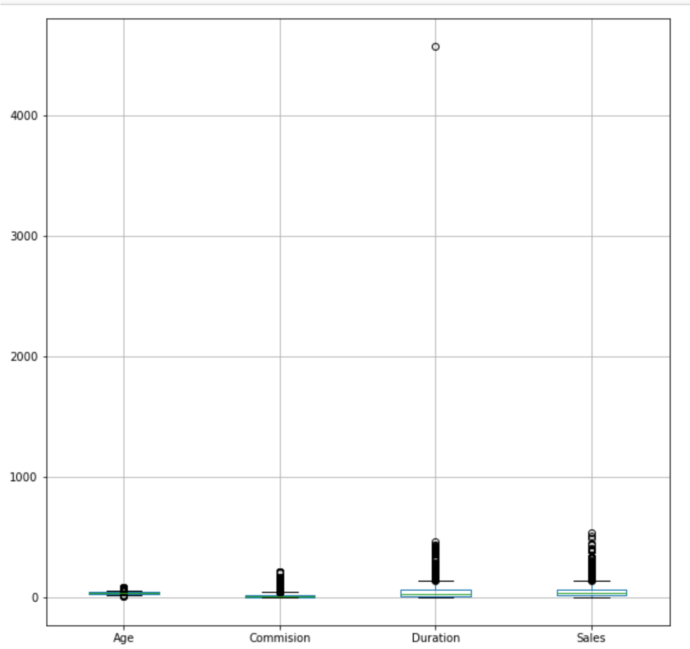


Fig.15 – Boxplot of Numeric Columns

<u>Age</u>

- The mean number of age is 38.09 whereas the SD is 10.46
- The maximum value is 84 and min value is 8
- Outliers are present in age

<u>Commision</u>

- The mean number of commision is 14.52 whereas the SD is 25.48
- The maximum value is 210 and min value is 0
- Outliers are present in commision

<u>Duration</u>

- The mean number of duration is 70 whereas the SD is 134.05
- The maximum value is 4580 and min value is -1
- Outliers are present in duration

<u>Sales</u>

- The mean number of sales is 60.24 whereas the SD is 70.73
- The maximum value is 539 and min value is 0
- Outliers are present in sales

Multi-Variate Analysis Numeric Datatype

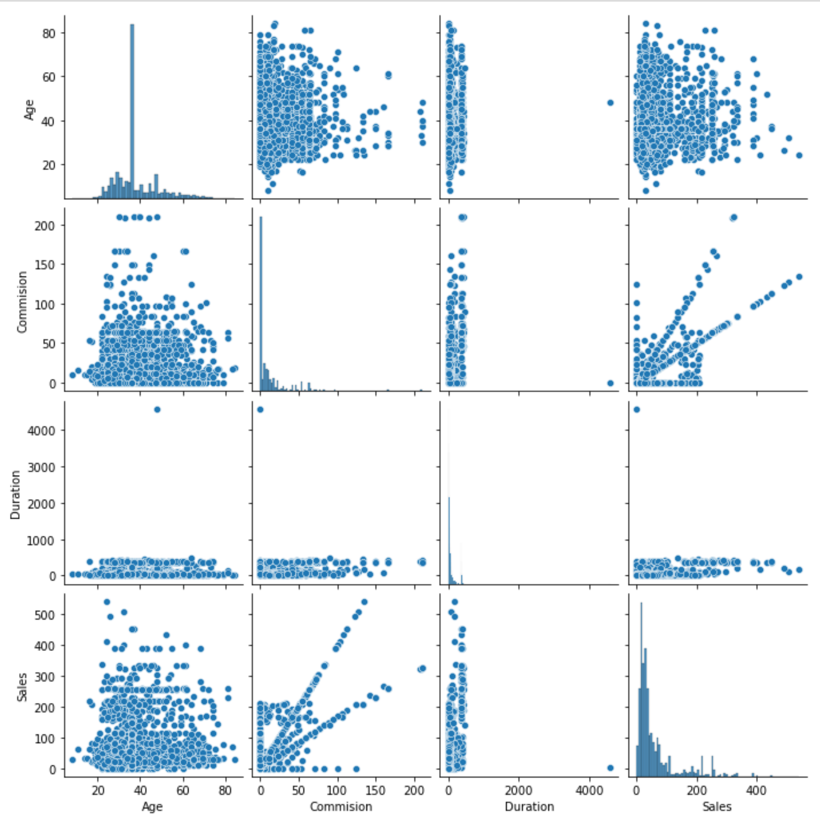


Fig.16 – Pair-plot of Numeric Columns

- The distribution is highly skewed towards right for all the numeric columns.

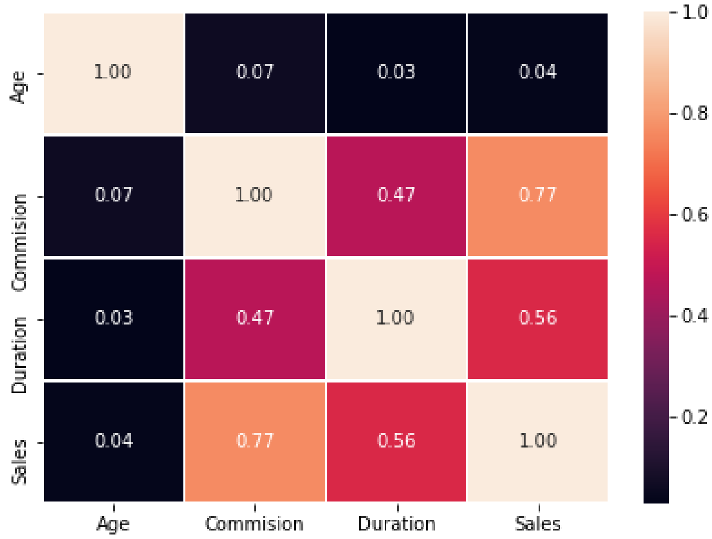


Fig.17 – Heat map of Numeric Columns

- Strong correlation is present between sales and duration
- Significant correlation is present between sales and commision

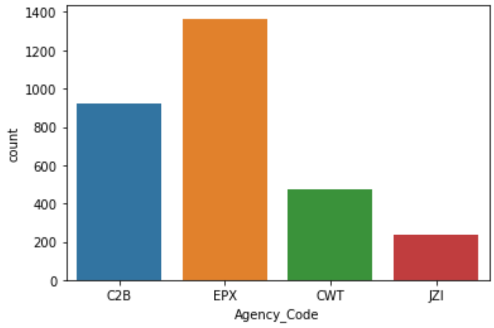
Univariate Analysis for Object Datatype

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

*********************************
Travel Agency    1837
Airlines         1163
Name: Type, dtype: int64

*********************************
No     2076
Yes     924
Name: Claimed, dtype: int64

*********************************
Online     2954
Offline      46
Name: Channel, dtype: int64

*********************************
Customised Plan      1136
Cancellation Plan     678
Bronze Plan           650
Silver Plan           427
Gold Plan             109
Name: Product Name, dtype: int64

*********************************
ASIA        2465
Americas     320
EUROPE       215
Name: Destination, dtype: int64

*********************************
```

Fig.18 – Value Counts of Object datatype

Bivariate Analysis of Object Datatype

Agency_code



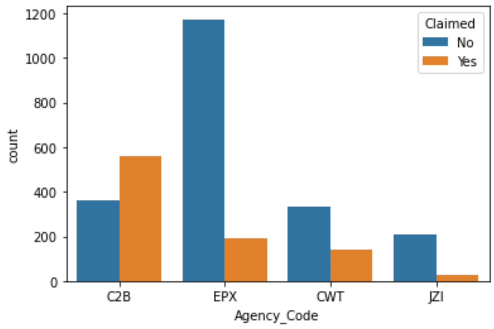
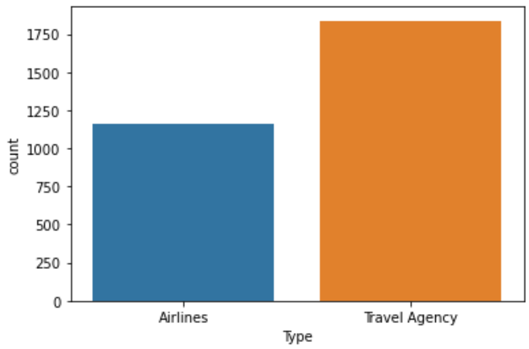


Fig.19 – Agency_code Uni-Bi Variant Analysis

Type



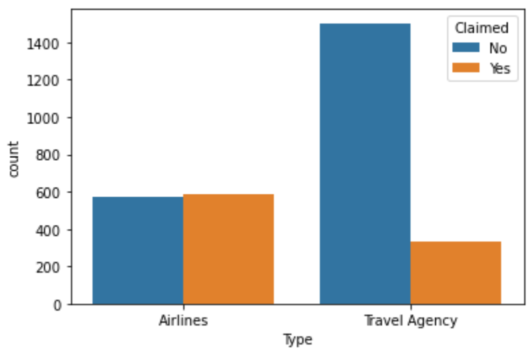


Fig.20 – Type Uni-Bi Variant Analysis

Channel

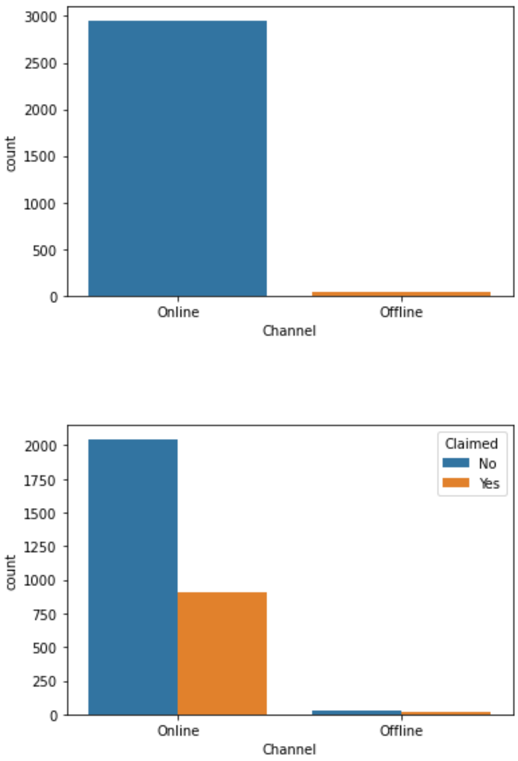


Fig.21 – Channel Uni-Bi Variant Analysis

Product Name

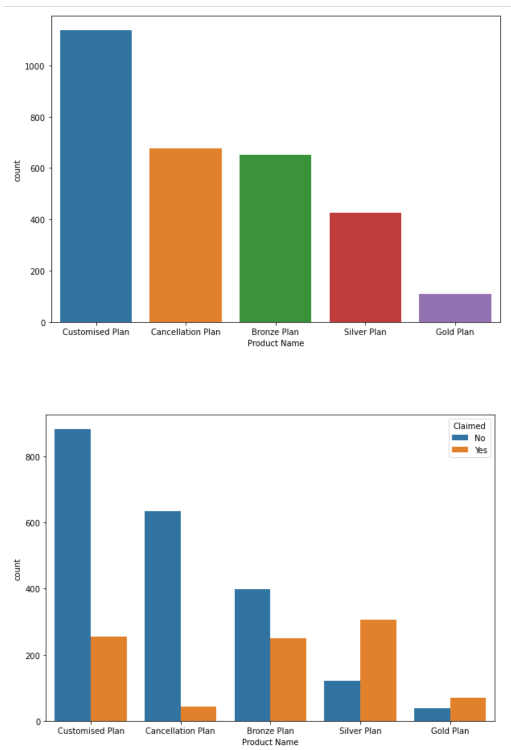
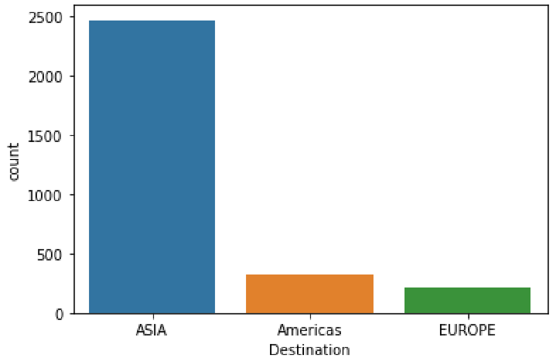


Fig.22 – Product Name Uni-Bi Variant Analysis

Destination



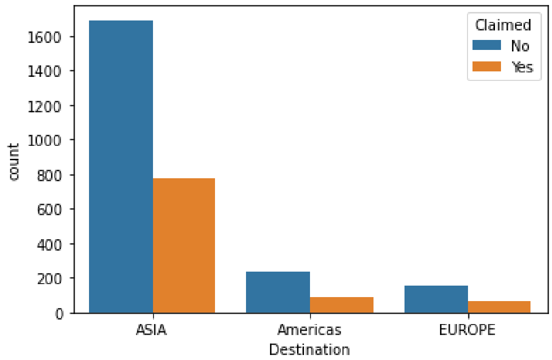
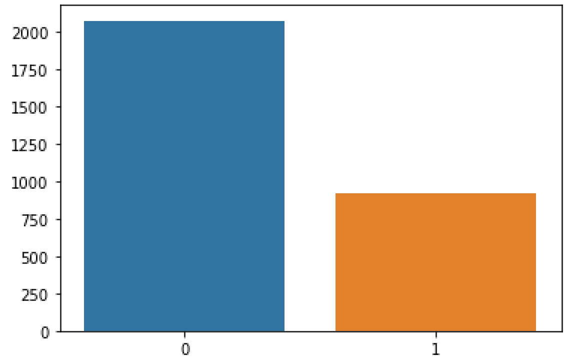


Fig.23 – Destination Uni-Bi Variant Analysis

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

- "Claimed" is our target variable.

Our Target Variable – Claimed




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


0 is for "No"
1 is for "Yes"
```

Fig.24 – Target Variable

- Claimed is our target variable and value 1 represent people who have claimed the tour insurance. Analysis of records containing 1 are of importance for the given dataset.
- Values of target variable are not perfectly balanced here.
- Records with "0" values are 69% and "1" values are 30%.

Dataset info after label Encoding

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Age           3000 non-null   int64  
 1   Agency_Code   3000 non-null   int8   
 2   Type          3000 non-null   int8   
 3   Claimed       3000 non-null   int8   
 4   Commision     3000 non-null   float64
 5   Channel       3000 non-null   int8   
 6   Duration      3000 non-null   int64  
 7   Sales         3000 non-null   float64
 8   Product Name  3000 non-null   int8   
 9   Destination   3000 non-null   int8   
dtypes: float64(2), int64(2), int8(6)
memory usage: 111.5 KB
```

Fig.25 – After Label Encoding

- Info of dataset after we have performed label encoding to all object type data to make them numeric category wise for analysis.

Dataset sample after label Encoding

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	0	0	0	0.70	1	7	2.51	2	0
1	36	2	1	0	0.00	1	34	20.00	2	0
2	39	1	1	0	5.94	1	3	9.90	2	1
3	36	2	1	0	0.00	1	4	26.00	1	0
4	33	3	0	0	6.30	1	53	18.00	0	0

Table .15

Train-Test Split

- We are splitting the data into 70% for training and 30% for testing and using random state = 123.
  Now as per our split dataset contains:
- x_train has 2100 rows and 9 columns
  y_train has labels 2100
  x_test has 900 rows and 9 columns
  y_test has labels 900

- We make Decision Tree Model, Random Forest and ANN, all using the same x_train and x_test datasets and their labels.

Decision Tree Classifier Model

- We build a DTC model using DecisionTreeClassifier function, taking criterion as “gini” and random state – 123.
- Grid search is applied to find the best parameters for making the DTC model. Cross validation values we are using is 3.
- Using the dictionary of parameters and array values for each parameter, we perform the grid search to DTC model.

```
grid_array_dtc = {'max_depth': [4,5,6],
                  'min_samples_leaf': [10,15,20,25],
                  'min_samples_split': [40,50,60]}
```

Table .16

- After fitting training data to our Grid search CV for DTC model we get the following:

```
GridSearchCV(cv=3, estimator=DecisionTreeClassifier(random_state=123),
             param_grid={'max_depth': [4, 5, 6],
                         'min_samples_leaf': [10, 15, 20, 25],
                         'min_samples_split': [40, 50, 60]})
```

Table .17

- We can find best parameters and best estimators.

```
DecisionTreeClassifier(max_depth=4, min_samples_leaf=20, min_samples_split=60,
                       random_state=123)
```

Table .18

- Now we take best estimators for the model and make prediction for training and testing data separately using these best estimators.

Features Importance for Decision Tree Classifier

	Imp_dtc
Type	0.000000
Channel	0.000000
Destination	0.000000
Age	0.005684
Duration	0.028131
Commision	0.037378
Product Name	0.103371
Sales	0.229098
Agency_Code	0.596338

Table .19

- Features that are of importance for the DTC model is presented above in form of a data frame and sorting done according to their value of importance in model evaluation in ascending order.

Random Forest Classifier Model

- We build a RFC model using RandomForestClassifier function and random state – 123.
- Grid search is applied to find the best parameters for making the RFC model. Cross validation values we are using is 3.
- Using the dictionary of parameters and array values for each parameter, we perform the grid search to RFC model.

```
grid_array_rfc = {'max_depth': [3,4,5],
                  'max_features': [5,6,7],
                  'min_samples_leaf': [25,30,35],
                  'min_samples_split': [90,100,110],
                  'n_estimators': [300,400,500]}
```

Table .20

- After fitting training data to our Grid search CV for RFC model we get the following:

```
GridSearchCV(cv=3, estimator=RandomForestClassifier(random_state=123),
             param_grid={'max_depth': [3, 4, 5], 'max_features': [5, 6, 7],
                         'min_samples_leaf': [25, 30, 35],
                         'min_samples_split': [90, 100, 110],
                         'n_estimators': [300, 400, 500]})
```

Table .21

- We can find best parameters and best estimators.

```
RandomForestClassifier(max_depth=5, max_features=7, min_samples_leaf=30,
                       min_samples_split=90, n_estimators=400,
                       random_state=123)
```

Table .22

- Now we take best estimators for the model and make prediction for training and testing data separately using these best estimators.

Features Importance for Decision Tree Classifier

	Imp_rfc
Channel	0.000000
Destination	0.004409
Type	0.010575
Age	0.032082
Commision	0.044188
Duration	0.052916
Product Name	0.198411
Sales	0.202642
Agency_Code	0.454777

Table .23

- Features that are of importance for the RFC model is presented above in form of a data frame and sorting done according to their value of importance in model evaluation in ascending order.

Artificial Neural Network Model

- We build an ANN model using MLPClassifier function and random state – 123.

- Before making the model we first need to scale our dataset:
  We perform fit_transform on training data.
  And only transform on testing data.
  This need to be done so that testing data is not known to our model while we are training the model. So as to ensure anomity is there in the model and the model does not get info about the test data we perform scaling after splitting the data into train and test for our ANN model.

- Grid search is applied to find the best parameters for making the ANN model. Cross validation values we are using is 3.
- Using the dictionary of parameters and array values for each parameter, we perform the grid search to ANN model.

```
grid_array_mlp = {'hidden_layer_sizes' : [100,200,300],
                  'max_iter' : [100,200,300],
                  'solver' : ['sgd', 'adam'],
                  'tol' : [0.01,0.001],}
```

Table .24

- After fitting training data to our Grid search CV for ANN model we get the following:

```
GridSearchCV(cv=3, estimator=MLPClassifier(random_state=123),
             param_grid={'hidden_layer_sizes': [100, 200, 300],
                         'max_iter': [100, 200, 300], 'solver': ['sgd', 'adam'],
                         'tol': [0.01, 0.001]})
```

Table .25

- We can find best parameters and best estimators.

```
MLPClassifier(hidden_layer_sizes=300, max_iter=100, random_state=123, tol=0.001)
```

Table .26

- Now we take best estimators for the model and make prediction for training and testing data separately using these best estimators.
- Feature Importance is not a part for ANN model.

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.

General Rules for performance Metrics:

- A model is said to perform well when it runs well in train as well as test data both
- The auc_ruc_score of both train and test should not differ more than 10% for the model to be valid.
- Higher the auc_ruc_score the better the model.
- If the difference between the auc_ruc_score for train and test set is greater than 10% then problem for overfitting and under fitting may arise.
  'Overfitting' is when model perform well in train but not in test set.
  'Under-fitting' is when model does not perform well in train but do well in test set.
  'Best Fit' is when model perform well in train as well as in test to a similar level.

- Confusion matrix and classification report is made for all the models.
- In classification report - '1' is our value of importance for model i.e. people who have claimed insurance.
  - Recall indicates how many of the actual data points are identified as True data points by the model.
  - Precision indicates the points that are identified as positive by the model, how many are really positive.
  - The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0.

  We will focus on recall and precision value in Classification Report for each model.

- An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:
  True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:
  $$TPR=TP/TP+FN$$
  False Positive Rate (FPR) is defined as follows:
  $$FPR=FP/FP+TN$$
- An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives.

Performance Metrics - DTC

- **Train Set**

```
Accuracy for DecisionTreeClassifier model for train data is: 0.7823809523809524


Classification report for DecisionTreeClassifier model for train data is:
              precision    recall  f1-score   support

           0       0.82      0.87      0.85      1444
           1       0.67      0.59      0.63       656

    accuracy                           0.78      2100
   macro avg       0.75      0.73      0.74      2100
weighted avg       0.78      0.78      0.78      2100


Confusion Matrix for DecisionTreeClassifier model for train data is:
```

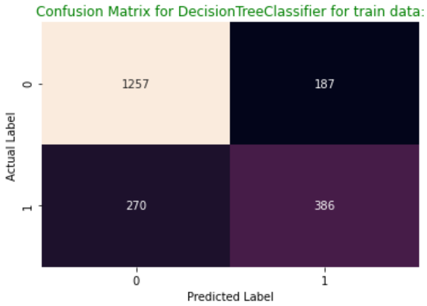


Table .27

AUC Score of DTC for Train set is – 0.82

- **Test Set**

```
Accuracy for DecisionTreeClassifier model for test data is: 0.7977777777777778


Classification report for DecisionTreeClassifier model for test data is:
              precision    recall  f1-score   support

           0       0.84      0.88      0.86       632
           1       0.68      0.62      0.64       268

    accuracy                           0.80       900
   macro avg       0.76      0.75      0.75       900
weighted avg       0.79      0.80      0.79       900


Confusion Matrix for DecisionTreeClassifier model for test data is:
```

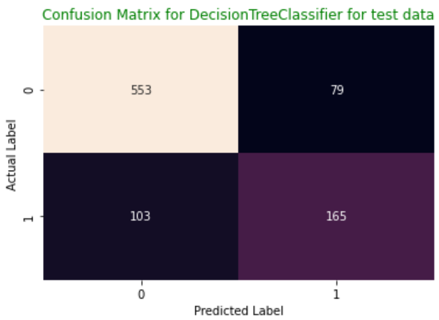


Table .28

AUC Score of DTC for Test set is – 0.83

- **ROC Curve for Train and Test set - DTC**

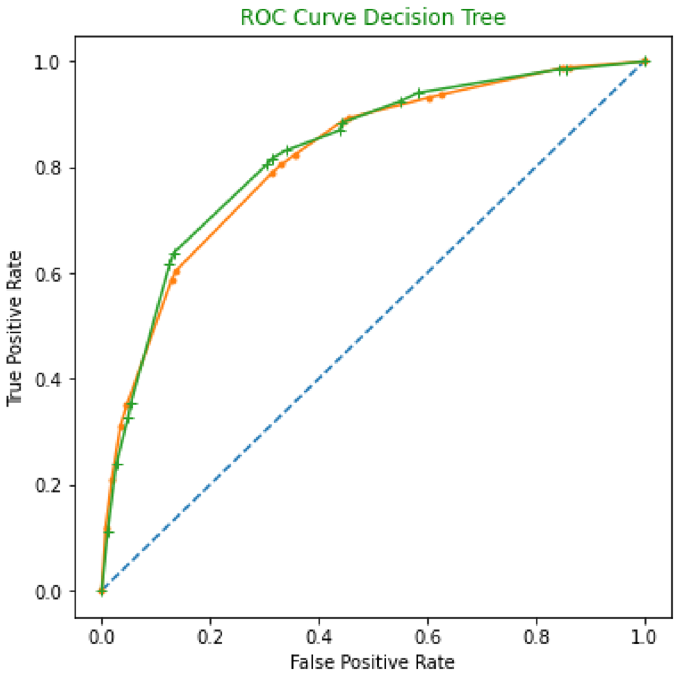


Fig .26

Line plot with dot "." marker is for train set and plus"+" marker is for test set.

Inferences:

- DTC model is performing well in both train and test set so it is a valid model.
  AUC score for train is 0.82
  AUC score for test is 0.83
- For test set for value '1' we have :
  Recall – 0.62
  Precision – 0.68
  This is model is performing well in test data and is able to correctly identify the labels for test set with acceptable values of recall and precision.

Performance Metrics - RFC

- **Train Set**

```
Accuracy for RandomForestClassifier model for train data is: 0.7914285714285715


Classification report for RandomForestClassifier model for train data is:
               precision    recall  f1-score   support

           0       0.82      0.90      0.86      1444
           1       0.71      0.56      0.63       656

    accuracy                           0.79      2100
   macro avg       0.76      0.73      0.74      2100
weighted avg       0.78      0.79      0.78      2100


Confusion Matrix for RandomForestClassifier model for train data is:
```

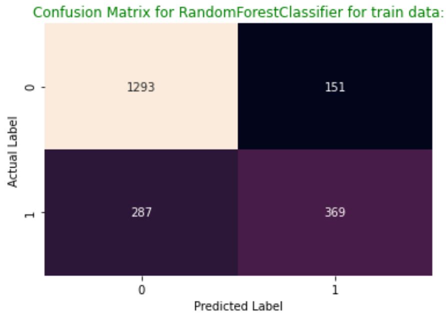


Table .29

AUC Score of DTC for Train set is – 0.84

- **Test Set**

```
Accuracy for RandomForestClassifier model for test data is: 0.7922222222222223


Classification report for RandomForestClassifier model for test data is:
               precision    recall  f1-score   support

           0       0.83      0.88      0.86       632
           1       0.68      0.58      0.63       268

    accuracy                           0.79       900
   macro avg       0.75      0.73      0.74       900
weighted avg       0.79      0.79      0.79       900


Confusion Matrix for RandomForestClassifier model for test data is:
```

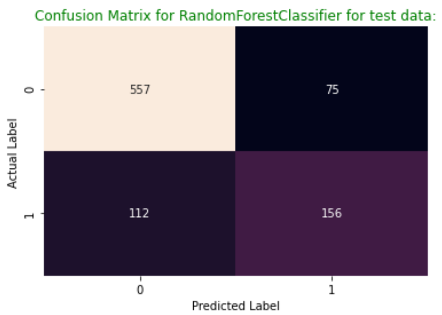


Table .30

AUC Score of DTC for Test set is – 0.83

- **ROC Curve for Train and Test set - RFC**

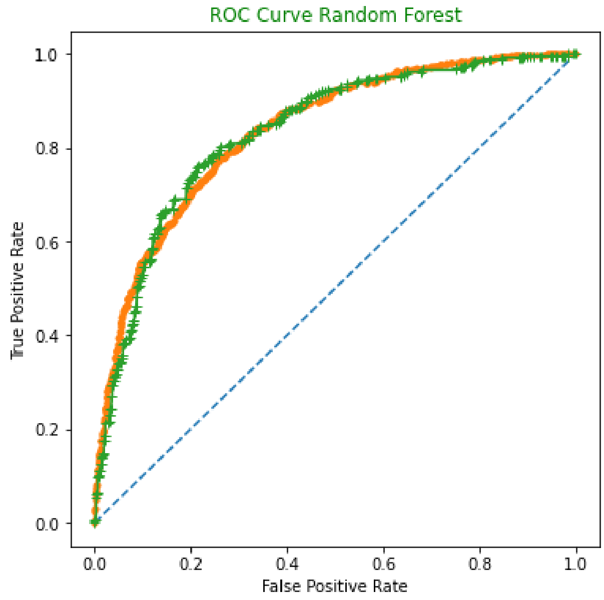


Fig .27

Line plot with dot "." marker is for train set and plus"+" marker is for test set.

Inferences:

- RFC model is performing well in both train and test set so it is a valid model.
  AUC score for train is 0.84
  AUC score for test is 0.83
- For test set for value '1' we have :
  Recall – 0.58
  Precision – 0.68
  This is model is performing well in test data and is able to correctly identify the labels for test set with acceptable values of recall and precision.

Performance Metrics - ANN

- **Train Set**

```
Accuracy for ANN model for train data is: 0.7452380952380953

Classification report for ANN model for train data is:
              precision    recall  f1-score   support

           0       0.82      0.81      0.81      1444
           1       0.59      0.60      0.60       656

    accuracy                           0.75      2100
   macro avg       0.70      0.71      0.71      2100
weighted avg       0.75      0.75      0.75      2100


Confusion Matrix for ANN model for train data is:
```

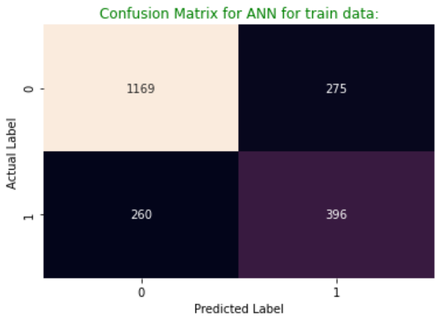


Table .31

AUC Score of DTC for Train set is – 0.76

- **Test Set**

```
Accuracy for ANN model for test data is: 0.7655555555555555

Classification report for ANN model for test data is:
              precision    recall  f1-score   support

           0       0.84      0.82      0.83       632
           1       0.60      0.64      0.62       268

    accuracy                           0.77       900
   macro avg       0.72      0.73      0.73       900
weighted avg       0.77      0.77      0.77       900


Confusion Matrix for ANN model for test data is:
```

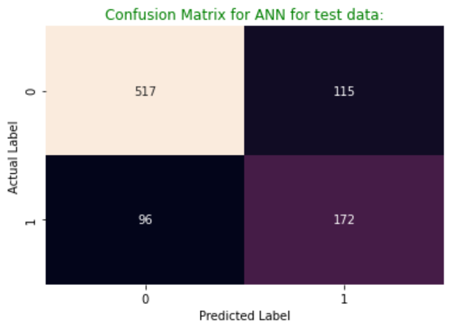


Table .32

AUC Score of DTC for Test set is – 0.79

- **ROC Curve for Train and Test set - ANN**

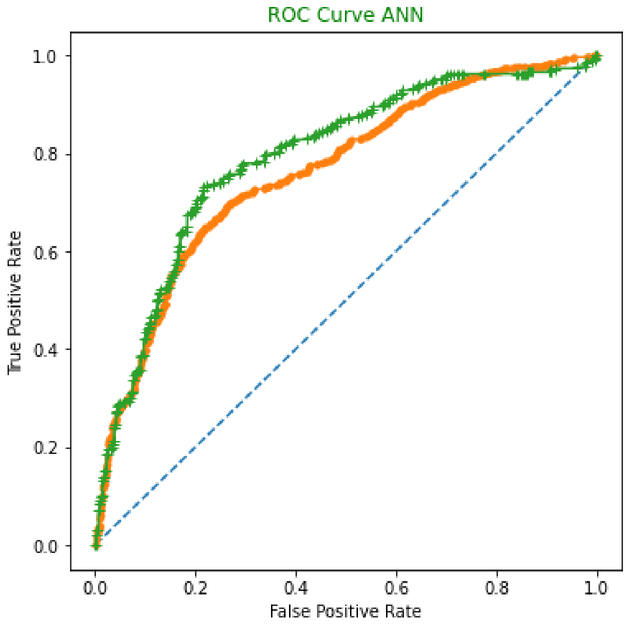


Fig .28

Line plot with dot "." marker is for train set and plus"+" marker is for test set.

Inferences:

- ANN model is performing well in both train and test set so it is a valid model.
  AUC score for train is 0.76
  AUC score for test is 0.79
- For test set for value '1' we have :
  Recall – 0.64
  Precision – 0.60
  This is model is performing well in test data and is able to correctly identify the labels for test set with acceptable values of recall and precision.

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

```
Area under the curve for Decision Tree Classification Model is 0.83
Area under the curve for Random Forest Classification Model is 0.83
Area under the curve for Artificial Neural Network Model is 0.79
```

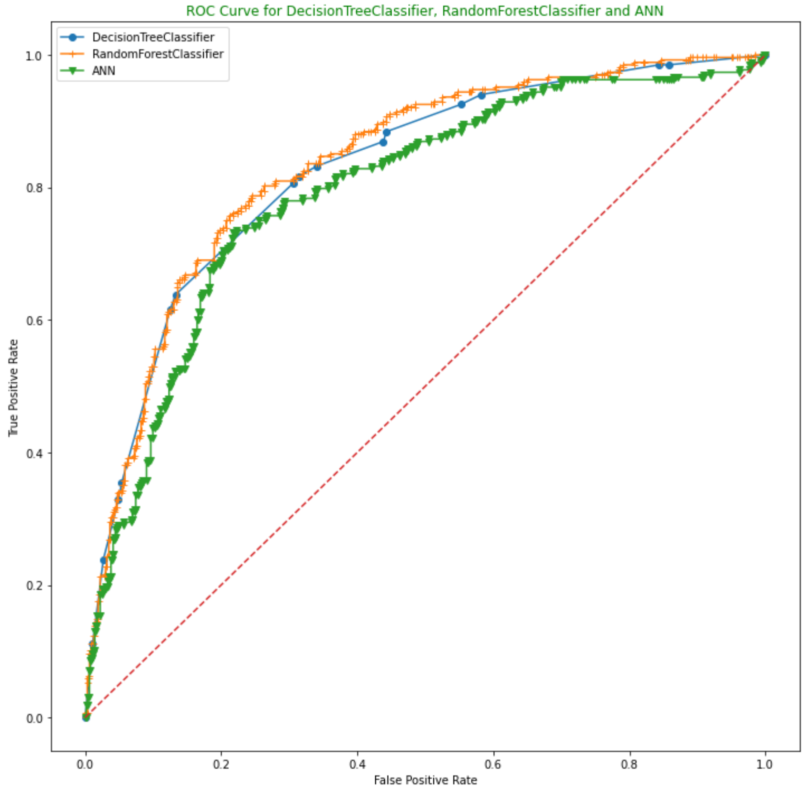


Fig .29 ROC Curve for all models for test set

	MODEL NAME					
	Decision Tree		Random Forest		ANN	
Accuracy Score -Train	0.78		0.79		0.74	
Accuracy Score -Test	0.79		0.79		0.76	
AUC - Train	0.82		0.84		0.76	
AUC - Test	0.83		0.83		0.79	
	Training Data Performance					
	0	1	0	1	0	1
Precision	0.82	0.67	0.82	0.71	0.82	0.59
Recall	0.87	0.59	0.9	0.56	0.81	0.6
F1-Score	0.85	0.63	0.86	0.63	0.81	0.6
	Testing Data Performance					
	0	1	0	1	0	1
Precision	0.84	0.68	0.83	0.68	0.84	0.6
Recall	0.88	0.62	0.88	0.58	0.82	0.64
F1-Score	0.86	0.64	0.86	0.63	0.83	0.62

Table .33 Comparison Metric Table of all models

Inferences:

- Please refer to ROC curve and Comparison metrics tables for all the model as shown above to the following inferences:

- Above graph shows roc curve for all the models for test data set to compare among them which model is best suited for our dataset.

  Visually area under curve for DTC and RFC model is very similar and least for ANN model

- Even though AUC score for DTC and RFC model is both same, RFC perform better because
  AUC score for RFC in train is 0.84 and in test is 0.83 while
  AUC score for DTC in train is 0.82 and in test is 0.83
  So RFC perform better in train and test set slightly than DTC.
  Hence, RFC performs best among all the three.
  Precision for value '1' is also highest for RFC model.

- ANN model performs worst among the three models with least accuracy and AUC score score for both train and test data.

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

Business insights:

- Since our target variable is not that balance:
  Records with "0" values are 69% and "1" values are 30%.
  More data will make our target variable balanced and all our model more accurate in prediction.
- As per our DTC model and RFC in dataset key features for our problem are:
  Agency_code
  Sales
  Product Name
  Duration
- As per dataset agency 'C2B' has highest no's of claims.
- People travelling with airlines has highest claim registered than people travelling with travel agency.
- As per dataset almost all insurance is done via online channel
- People in silver plan has highest claims registered.

Recommendations:

- Increase customer satisfaction to increase revenue generated.
- Reduce cost in handling insurance that are registered.
- Identify and detection of fraud claims across all verticals in company by analysing meaningful connection.

THE END