2020

Mr.Jaydeep Bhaskar Ashtekar
ashtekarjaydeep@yahoo.in
PGPDSBA:Group 9

[DATA MINING]

This report gives report of two case studies. First case study based on is the clustering techniques and insights of it.Second is for model building using CART,Random forest and ANN(MLPClassifier)

Content

PROBLEM STATEMENT 1:CART

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 EDA

The data set provides details of spending in different formats done by customers of one of the leading bank. As per the task different segments cannot be made just by seeing the data. We need to take help of the different clustering techniques.

Before doing that we first visualize and explore the data to see its insights and patterns.

1.Imported different libraries

2.Data set is loaded as df.df.head() gave first five entries which shows csv file is loaded without any issue.

3.Check for any null values df.isna() =0 and df.isnull()=0 so null values are present.

4.Check for duplicates no duplicates are present.(df.duplicated()=0)

5.Data types of variables:

spending	float64
advance_payments	float64
probability_of_full_payment	float64
current_balance	float64
credit_limit	float64
min_payment_amt	float64
max_spent_in_single_shopping	float64

All data types are float64.

6. Shape / size of df

Data set has 210 rows and 7 columns.

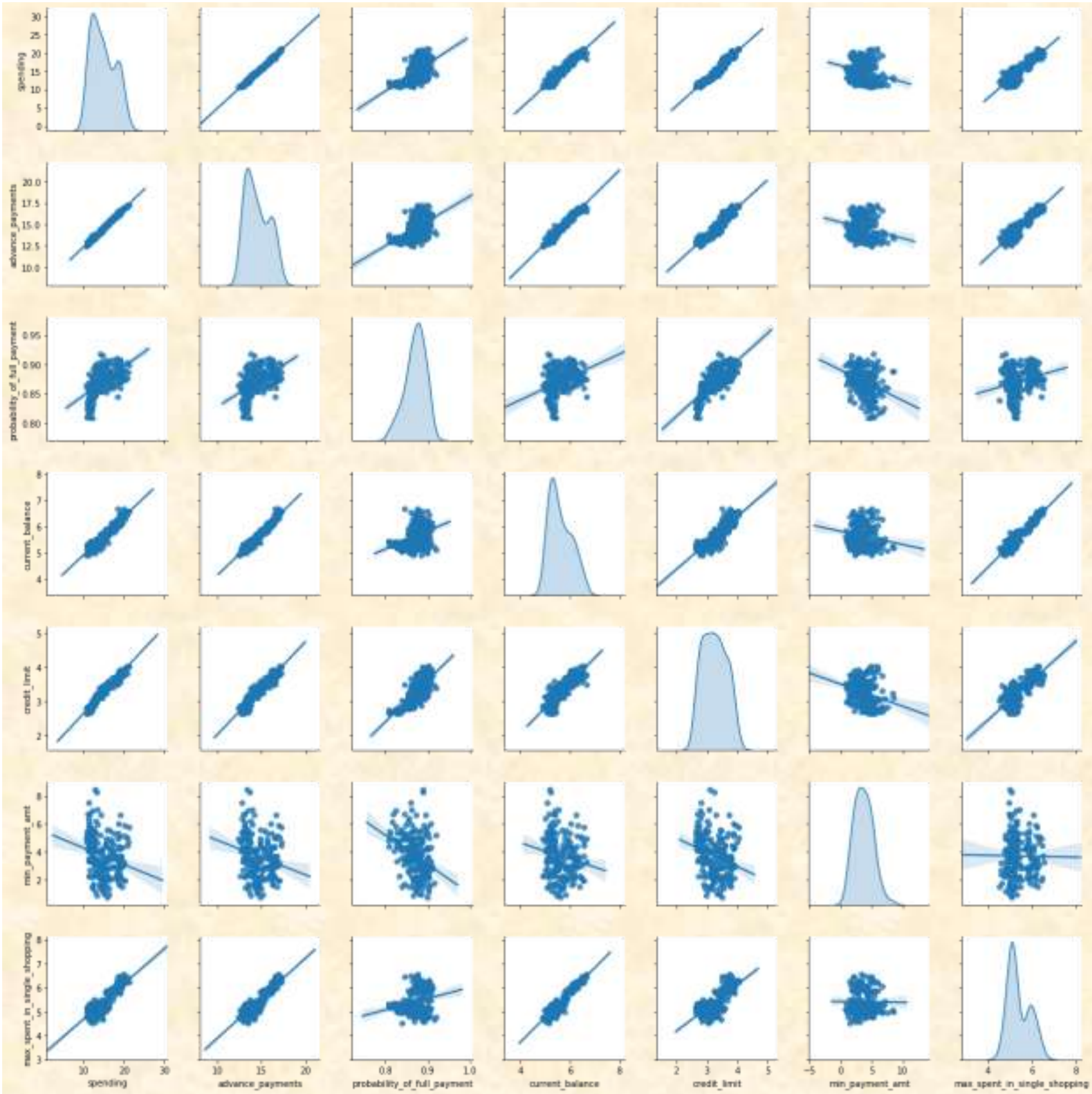
7.Data Description:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	210.00	210.00	210.00	210.00	210.00	210.00	210.00
mean	14.85	14.56	0.87	5.63	3.26	3.70	5.41
std	2.91	1.31	0.02	0.44	0.38	1.50	0.49
min	10.59	12.41	0.81	4.90	2.63	0.77	4.52
0.25	12.27	13.45	0.86	5.26	2.94	2.56	5.05
0.50	14.36	14.32	0.87	5.52	3.24	3.60	5.22
0.75	17.31	15.72	0.89	5.98	3.56	4.77	5.88
max	21.18	17.25	0.92	6.68	4.03	8.46	6.55

8.Median and mode

Variables	**Median**	**Mode**
spending	14.355	11.23,14.11,15.38
advance_payments	14.32	13.47
probability_of_full_payment	0.87345	0.8823
current_balance	5.5235	5.236
credit_limit	3.237	3.026
min_payment_amt	3.599	2.129,2.221
max_spent_in_single_shopping	5.223	5.001

9. Pair plot: Pair plot gives graph of each variable with respect to each other and itself.



9.Skewness:

Variable	**Skewness**
spending	0.399889
advance_payments	0.386573
probability_of_full_payment	-0.537954
current_balance	0.525482
credit_limit	0.134378
min_payment_amt	0.401667
max_spent_in_single_shopping	0.561897

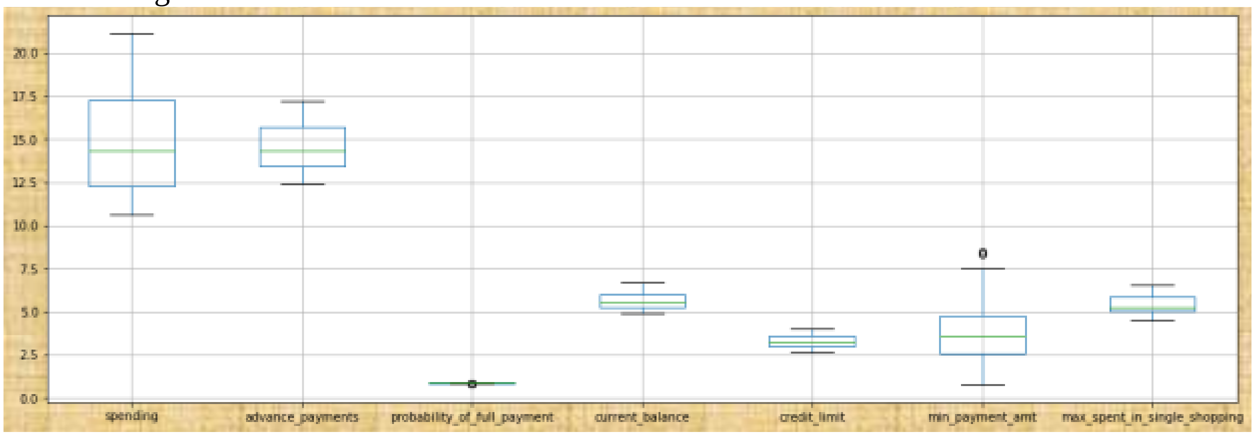
Probability_of_full_payment is negatively skewed (**Mode> Median> Mean**).Significance of skewness is explained below.

All other variables are positively skewed (**Mode< Median< Mean).**

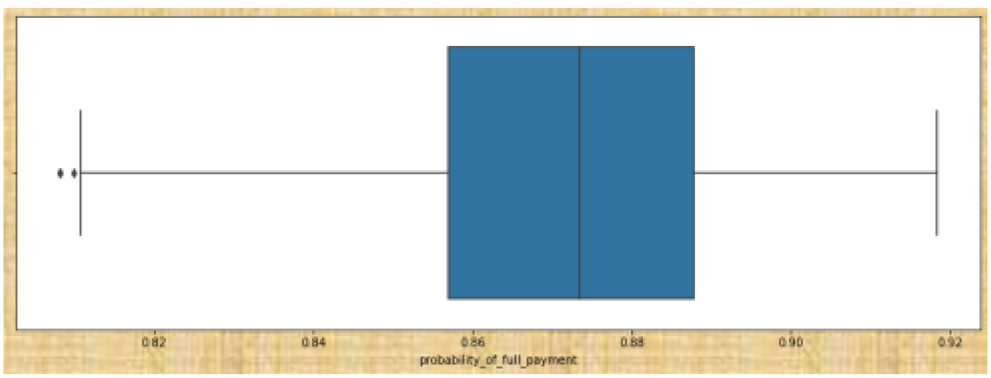
Let's observe diagonals of pair plot. It can be concluded that:

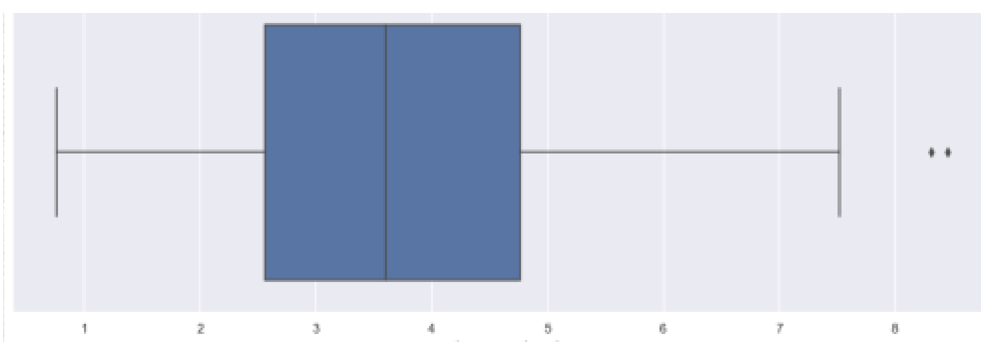
1. Spending is multimodal and normally distributed.
2. Advanced payments are multimodal and normally distributed.
3. All variables are approximately normally distributed.

10. Checking outliers:



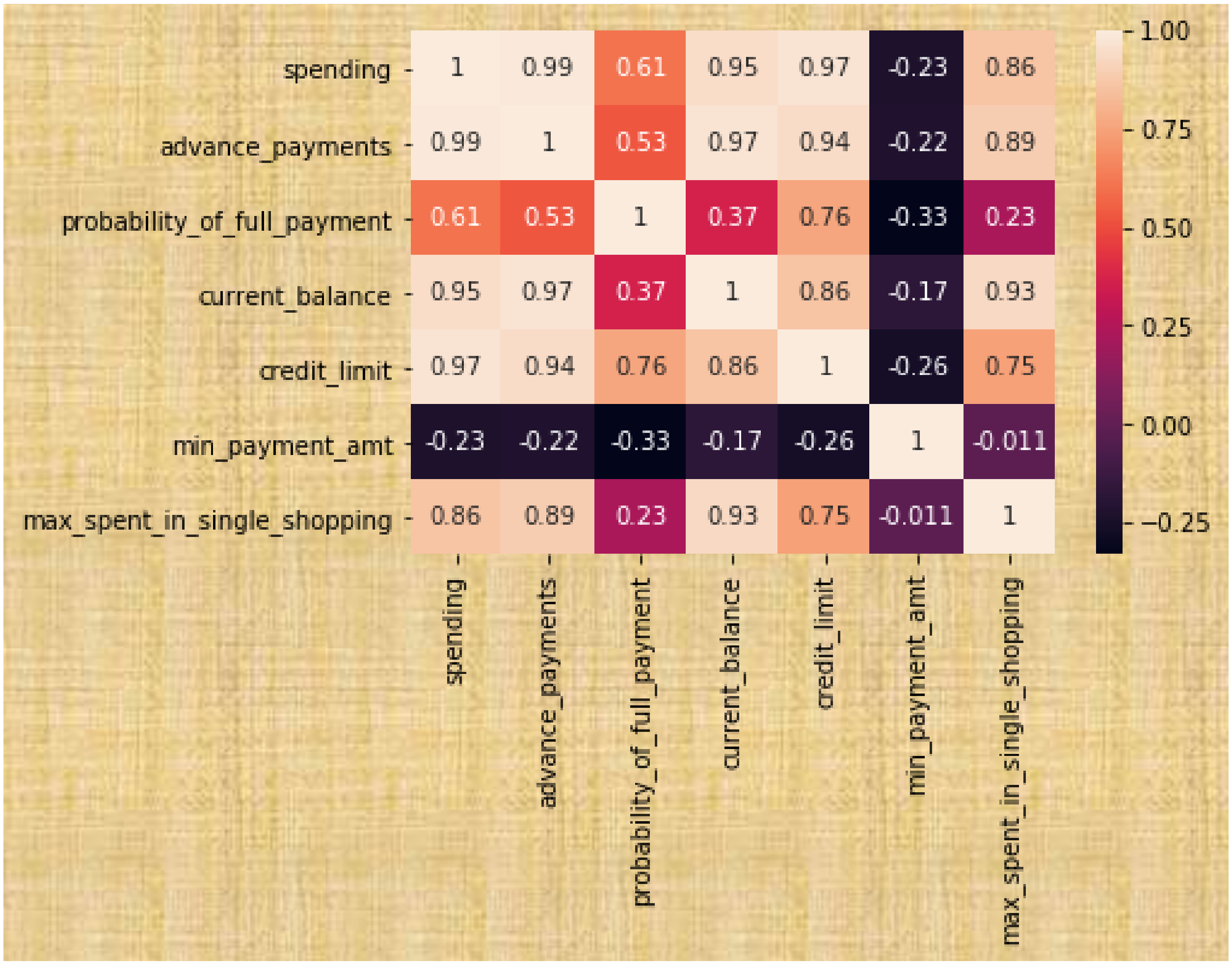
So probalility and min payment amt are having two outlires.





Number of outliers are very less so not removed.

11. Correlation:
Here exists good correlation amongst all variables except min_payment_amt.



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

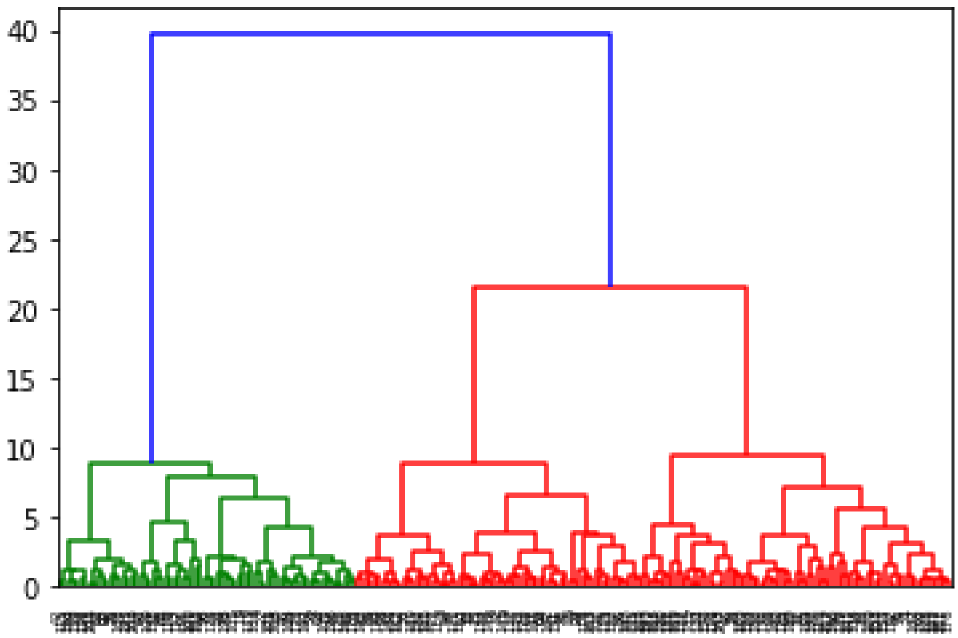
Ans:
Yes, I think scaling is necessary for clustering. Because, units are different for different variables. To bring them in same range irrespective of units,scaling is necessary.
While calculating distances also scaling will be more advantageous.

If we don't scale data then it may happen that we may give attributes which have larger magnitudes more importance.

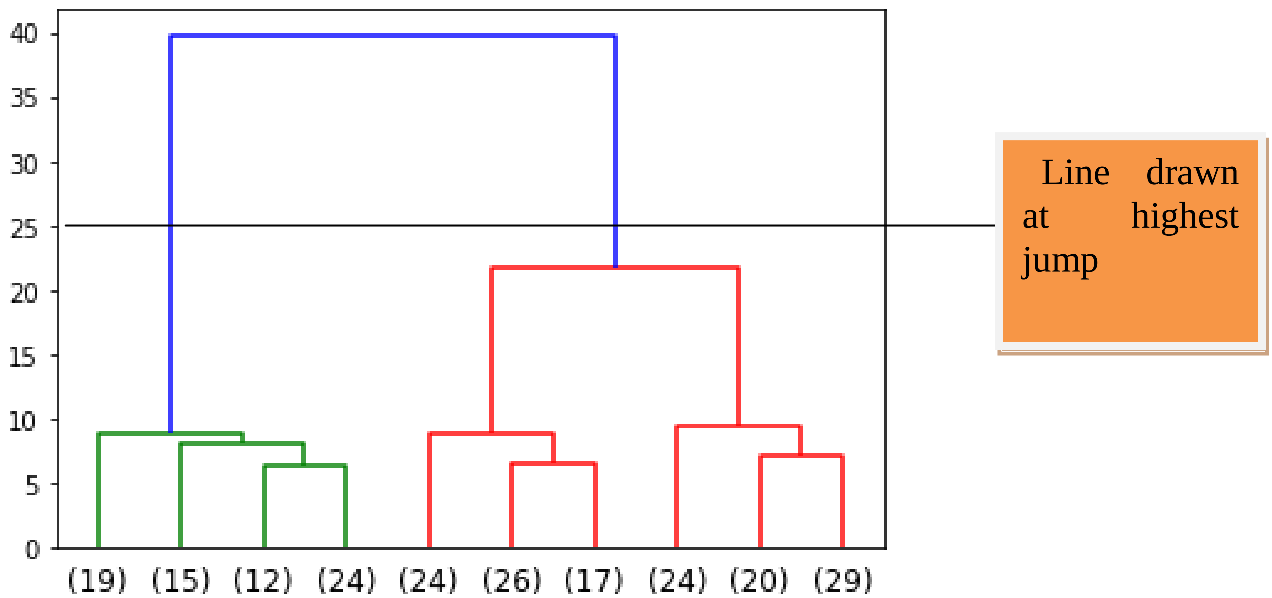
But if all data has same unit or scale then we may neglect data scaling.

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

Ans: Dendogram obtained for the hierarchical clustering using wardlink is as follows:



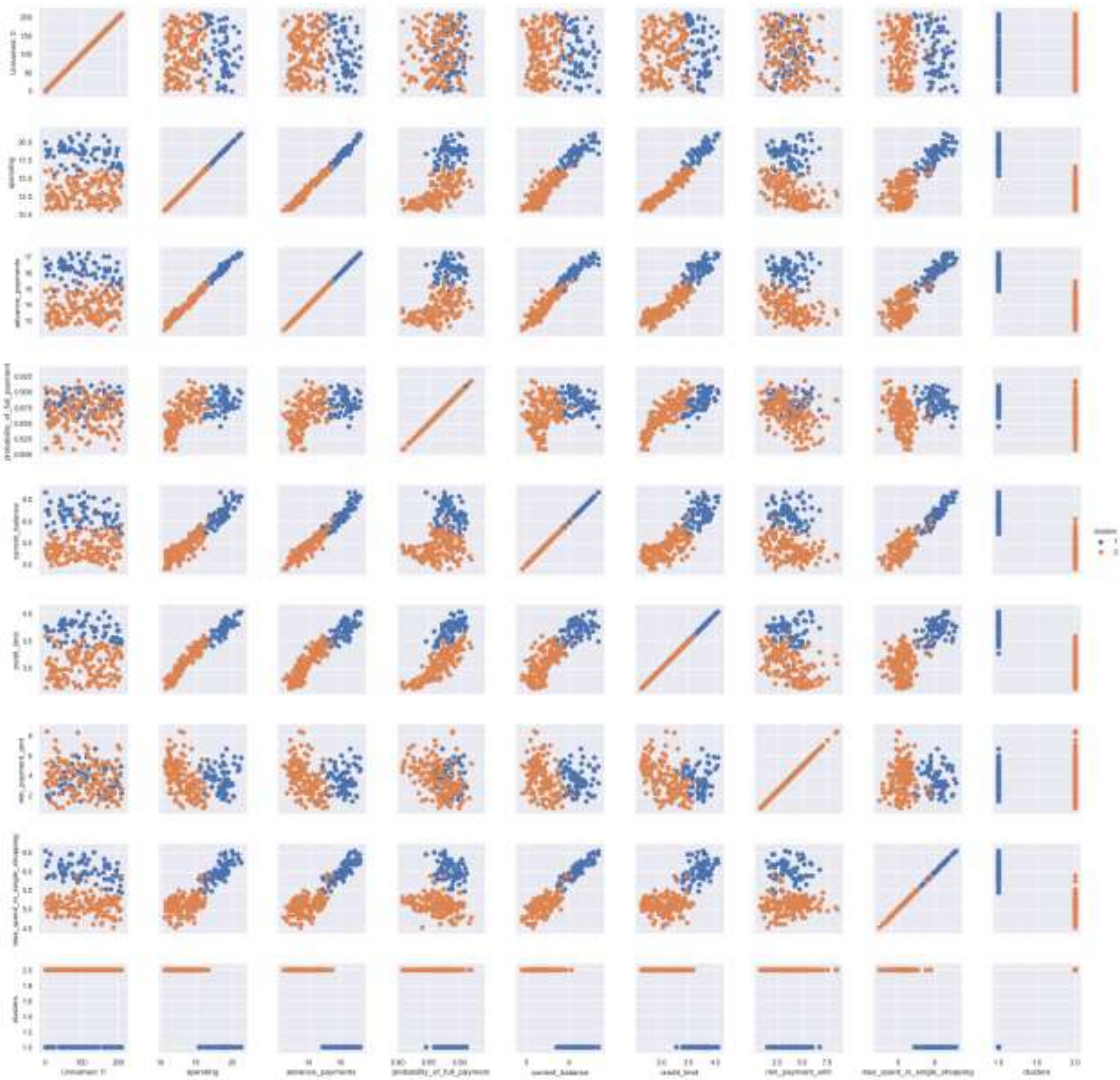
By considering last 10 vertical lines following dendogram is obtained:



So from this dendogram two optimum clusters are suggested.(Red and Green)
Also, if we draw horizontal line from highest jump of vertical line is intersecting at two lines as shown above so number of optimum clusters are 2.
Scatter plot of the same is shown below: (Neglect Variable Unknown)

```
sns.set()
g = sns.PairGrid(df2,hue="clusters")
g = g.map(plt.scatter,s=40)
g = g.add_legend()
```

(*df2 is dataframe with clusters 1 and 2)

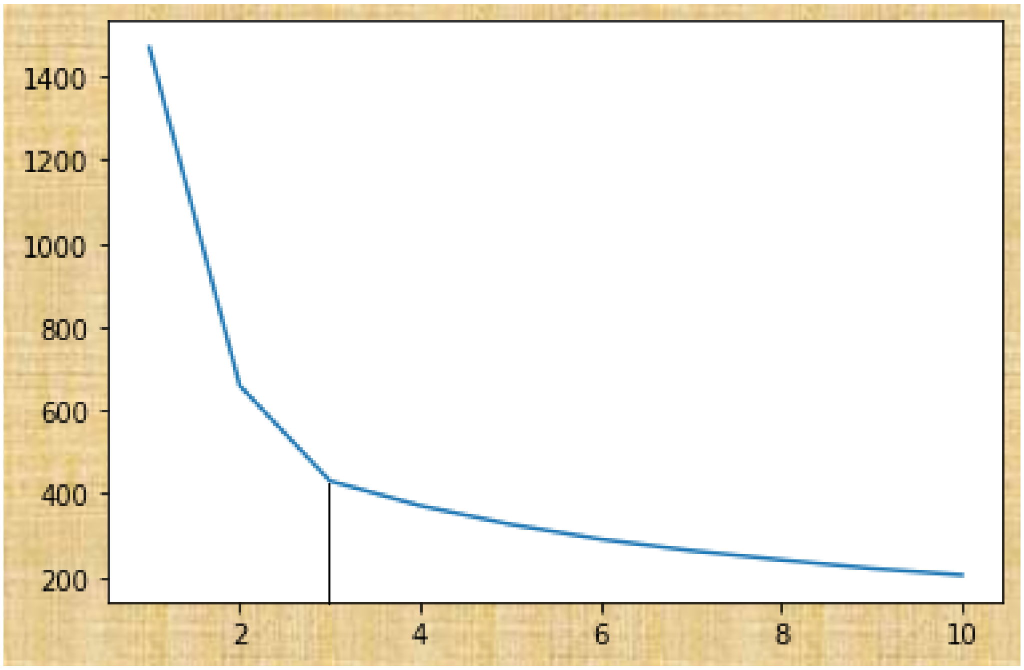


So it can be seen from the above graph that two colors shows two different clusters.

Cluster no.	Spending (1000s)	advance_payments (100s)	probability_of_full_payment	current_balance (1000s)	credit_limit (10000s)	min_payment_amt (100s)	max_spent_in_single_shopping(1000s)	clusters_size
1	18.37	16.15	0.8844	6.16	3.68	3.64	6.02	70
2	13.08	13.77	0.8643	5.36	3.05	3.73	5.1	140

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

Ans:

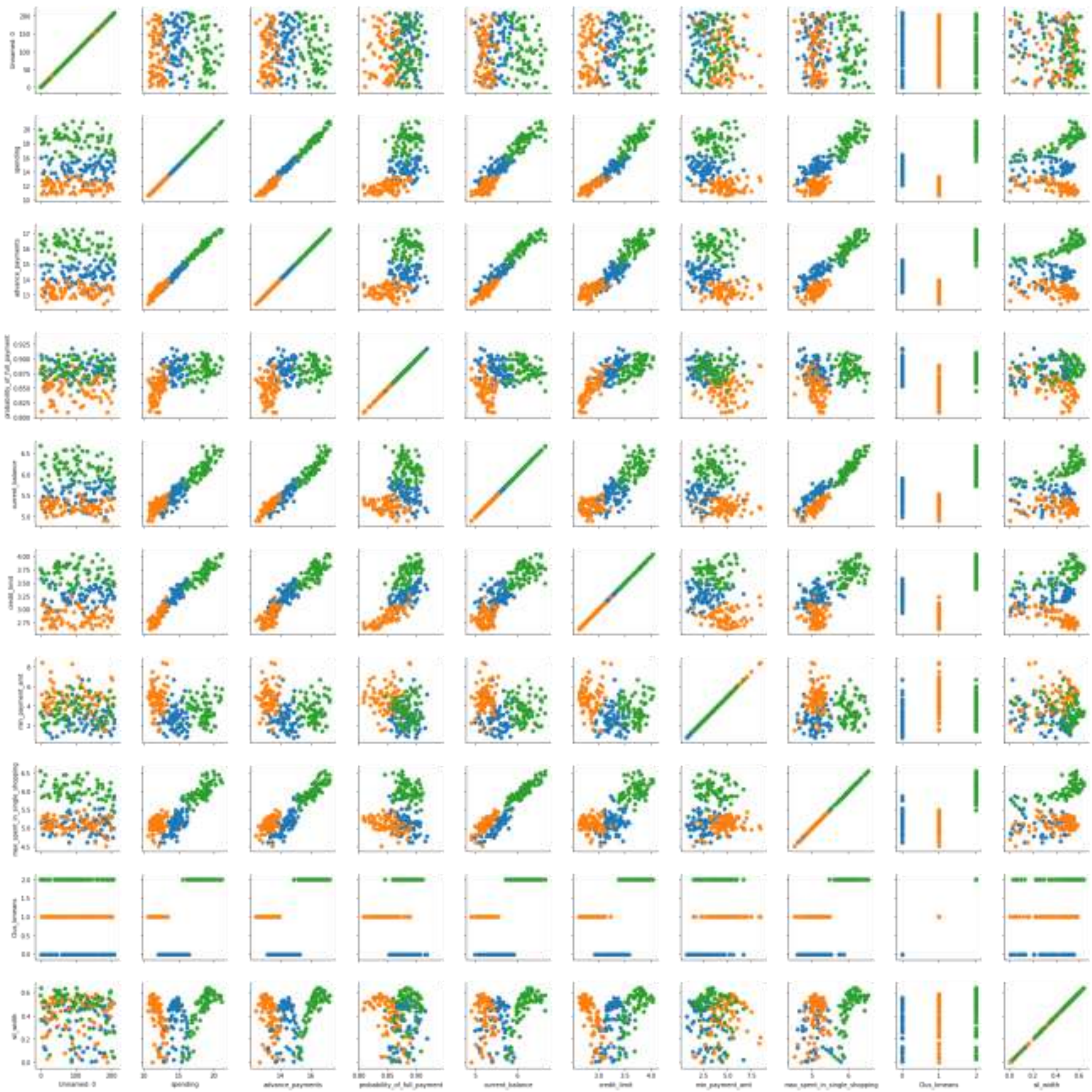


silhouette score vs number of clusters

So graph shows sil.score vs clusters. So slope of score sharply decreases after 3 so optimum number of clusters are selected as 3.

Following graph shows three clusters by three different colors.(*Plz neglect unkonows,clust_kmeans,sil_width columns)

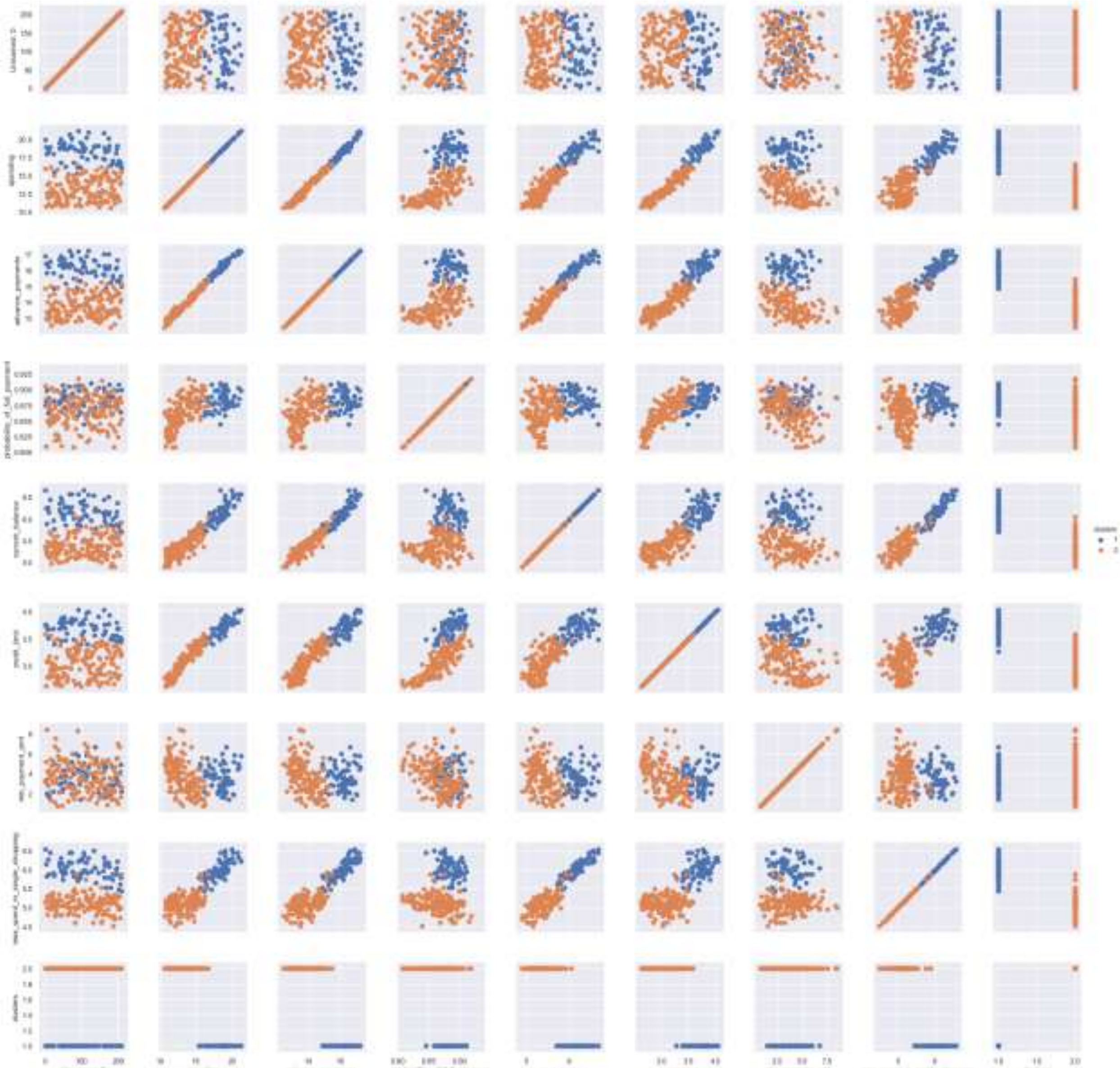
Cluster no.	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	clusters_size
0	14.44	14.34	0.8816	5.51	3.2592	2.7073	5.1208	71
1	11.86	13.25	0.8482	5.23	2.85	4.74	5.1	72
2	18.49	16.2	0.8842	6.18	3.7	3.63	6.04	67



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

Ans:

Scatter plots for hierarchical clustering gives better presentation of clustering.
Less points are overlapping with each other.



Cluster no.	Spending (1000s)	advance_payments (100s)	probability_of_full_payment	current_balance (1000s)	credit_limit (10000s)	min_payment_amt (100s)	max_spent_in_single_shopping(1000s)	clusters_size
1	18.37	16.15	0.8844	6.16	3.68	3.64	6.02	70
2	13.08	13.77	0.8643	5.36	3.05	3.73	5.1	140

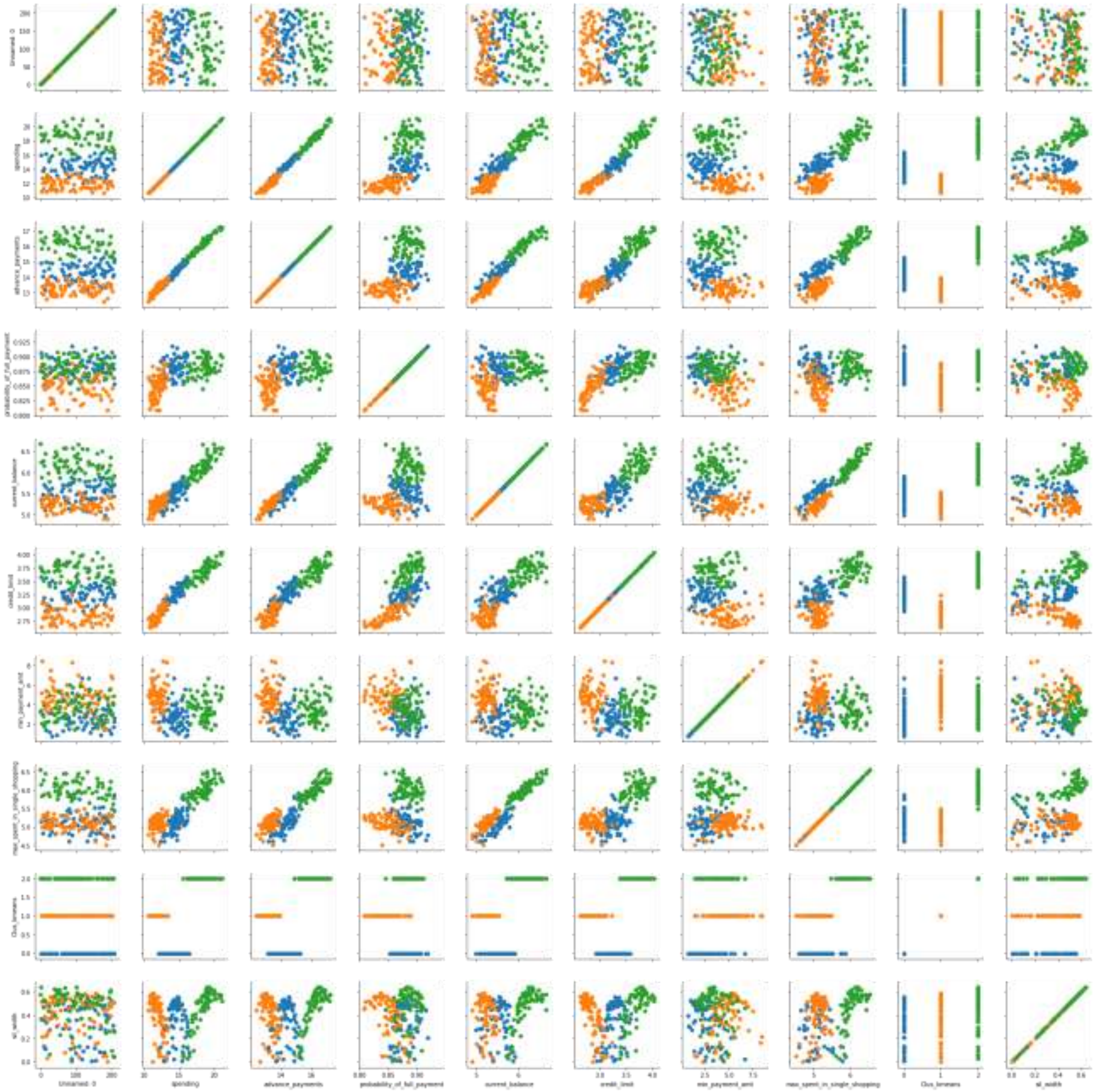
Promotional strategies:

If **two** clusters are considered then two classes can be made depending on spending capacity i.e. Platinum (1) and Gold (2).
For platinum:
1. Platinum class is spending more (average 18.37 units). So, their credit limit can be raised (now it is 3.64 units).
2. Due to more spending capacity costlier items can be promoted to this class first then to other class.
3. For Gold class min_payment_amt is higher that can be reduced little as the gold class size is 140(double that of the platinum).

If **three** clusters are considered then three classes can be made depending on spending capacity i.e. Platinum (2) and Gold (0) and silver (1).

Costlier items should be promoted in above sequence (2,0,1)

Cluster no.	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	clusters_size
0	14.44	14.34	0.8816	5.51	3.2592	2.7073	5.1208	71
1	11.86	13.25	0.8482	5.23	2.85	4.74	5.1	72
2	18.49	16.2	0.8842	6.18	3.7	3.63	6.04	67



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.

Ans: Imported necessary libraries

1. No null values in data set.

2. Data types

```
Data columns (total 10 columns):
Age                3000 non-null int64
Agency_Code        3000 non-null object
Type               3000 non-null object
Claimed            3000 non-null object
Commision          3000 non-null float64
Channel            3000 non-null object
Duration           3000 non-null int64
Sales              3000 non-null float64
Product Name       3000 non-null object
Destination        3000 non-null object
```

```
3.No. of duplicate rows
data.duplicated().sum()=139
```

```
4.Data mean:
Age             36.00
Commision        4.63
Duration        26.50
Sales           33.00
```

```
5.Data mode:
```

Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
36	EPX	Travel Agency	No	0.0	Online	8	20.0	Customised Plan	ASIA

6. Data Description

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
count	3000.00	3000	3000.00	3000.00	3000.00	3000.00	3000.00	3000.00	3000.00	3000.00
unique	NaN	4.00	2.00	2.00	NaN	2.00	NaN	NaN	5.00	3.00
top	NaN	EPX	Travel Agency	No	NaN	Online	NaN	NaN	Customised Plan	ASIA
freq	NaN	1365.00	1837.00	2076.00	NaN	2954.00	NaN	NaN	1136.00	2465.00
mean	38.09	NaN	NaN	NaN	14.53	NaN	70.00	60.25	NaN	NaN
std	10.46	NaN	NaN	NaN	25.48	NaN	134.05	70.73	NaN	NaN
min	8.00	NaN	NaN	NaN	0.00	NaN	-1.00	0.00	NaN	NaN
0.25	32.00	NaN	NaN	NaN	0.00	NaN	11.00	20.00	NaN	NaN
0.50	36.00	NaN	NaN	NaN	4.63	NaN	26.50	33.00	NaN	NaN
0.75	42.00	NaN	NaN	NaN	17.24	NaN	63.00	69.00	NaN	NaN
max	84.00	NaN	NaN	NaN	210.21	NaN	4580.00	539.00	NaN	NaN

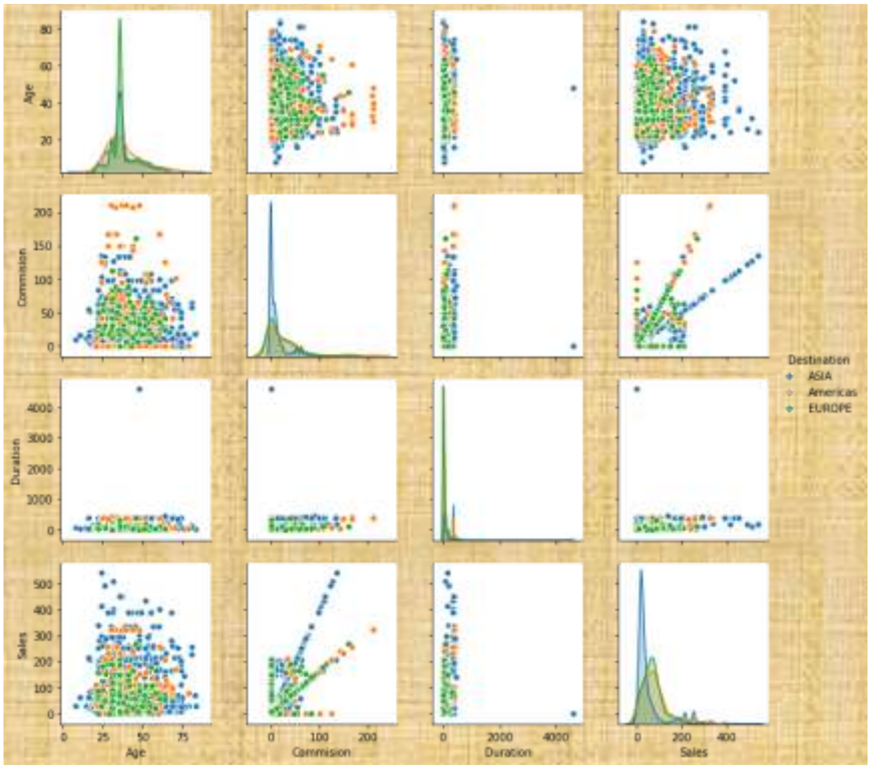
7.Data range:

```
Age              76.00
Commision        210.21
Duration         4581.00
Sales            539.00
```

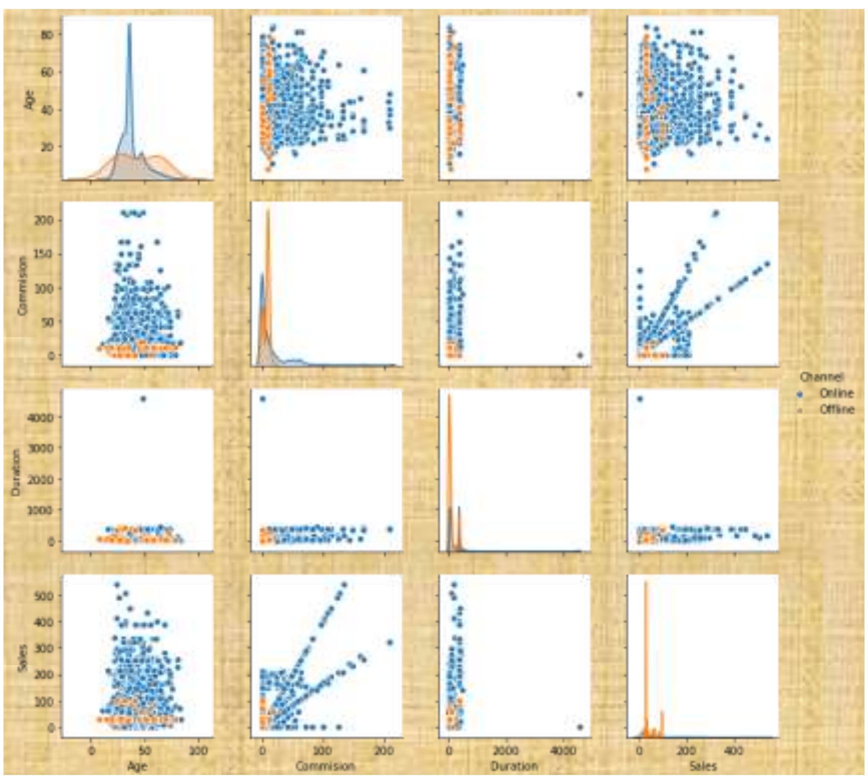
7. data.skew()

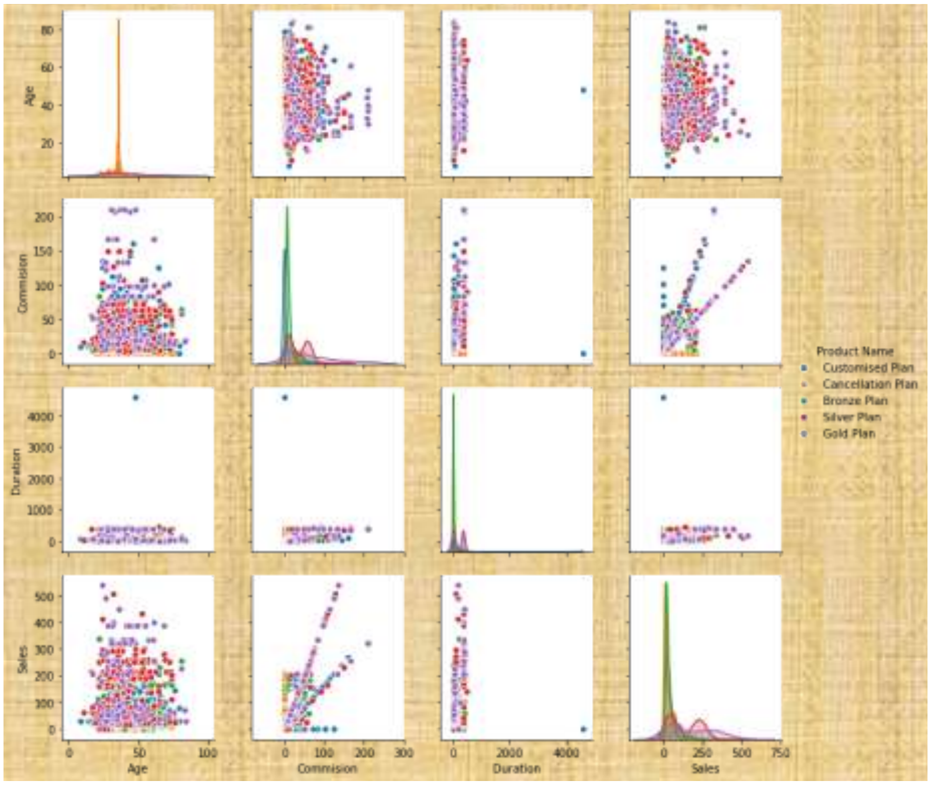
```
Age              1.149713
Commision        3.148858
Duration         13.784681
Sales            2.381148
```

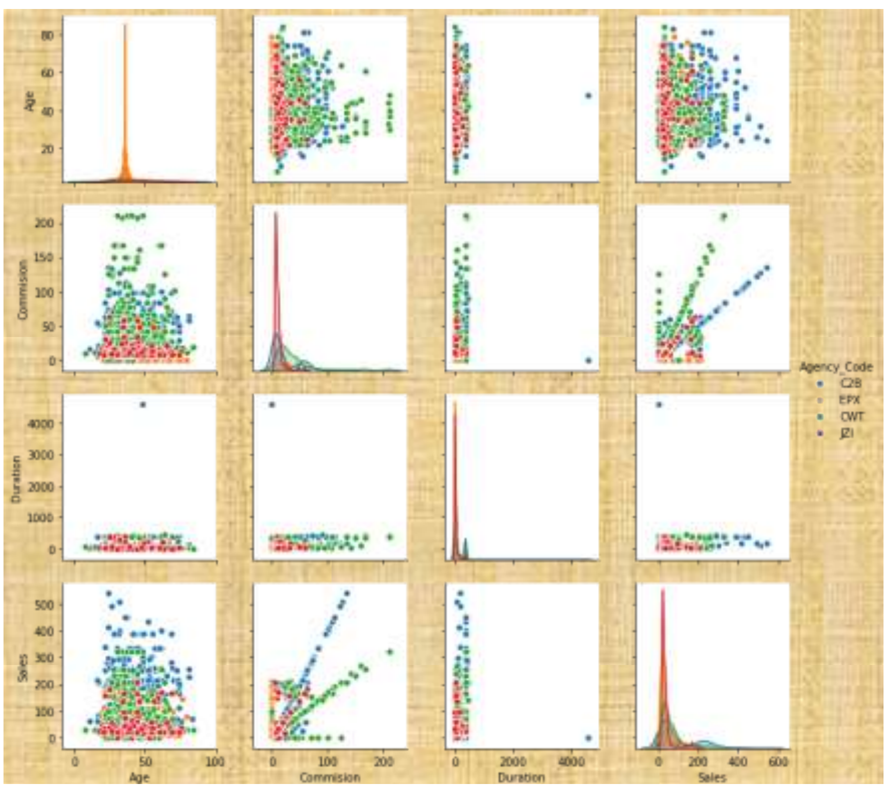
Skew is positive so, (**Mode< Median< Mean).**

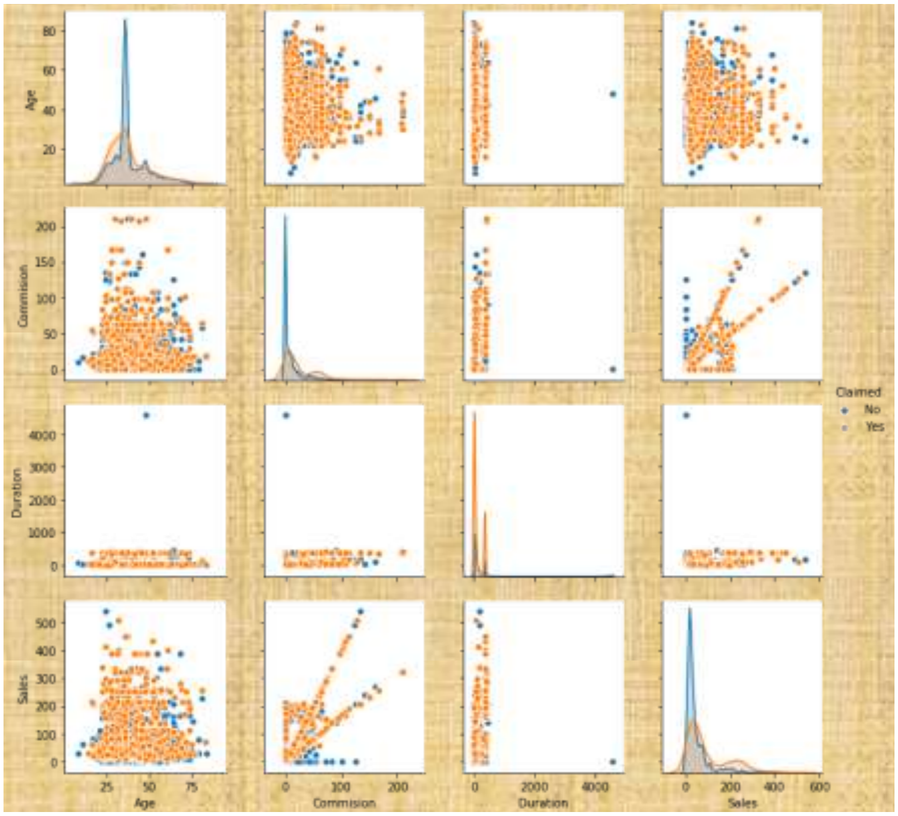


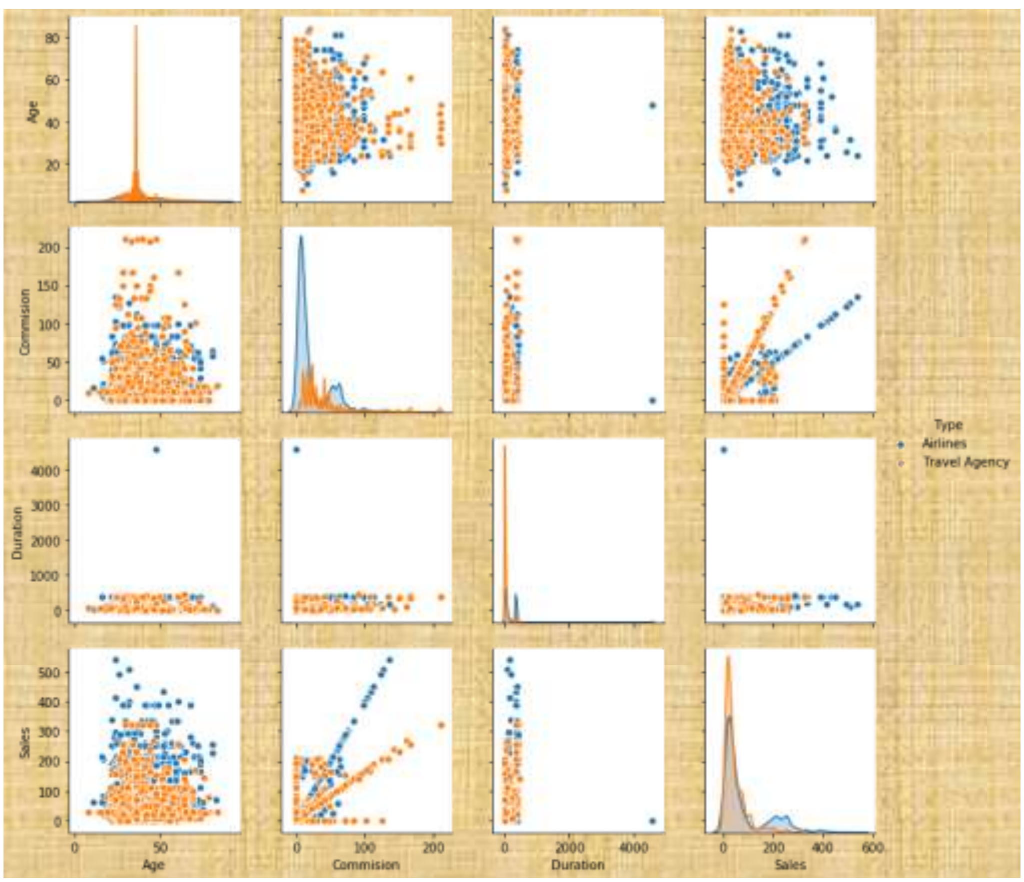
Top customers are from ASIA.(Same can be interpretated from description).



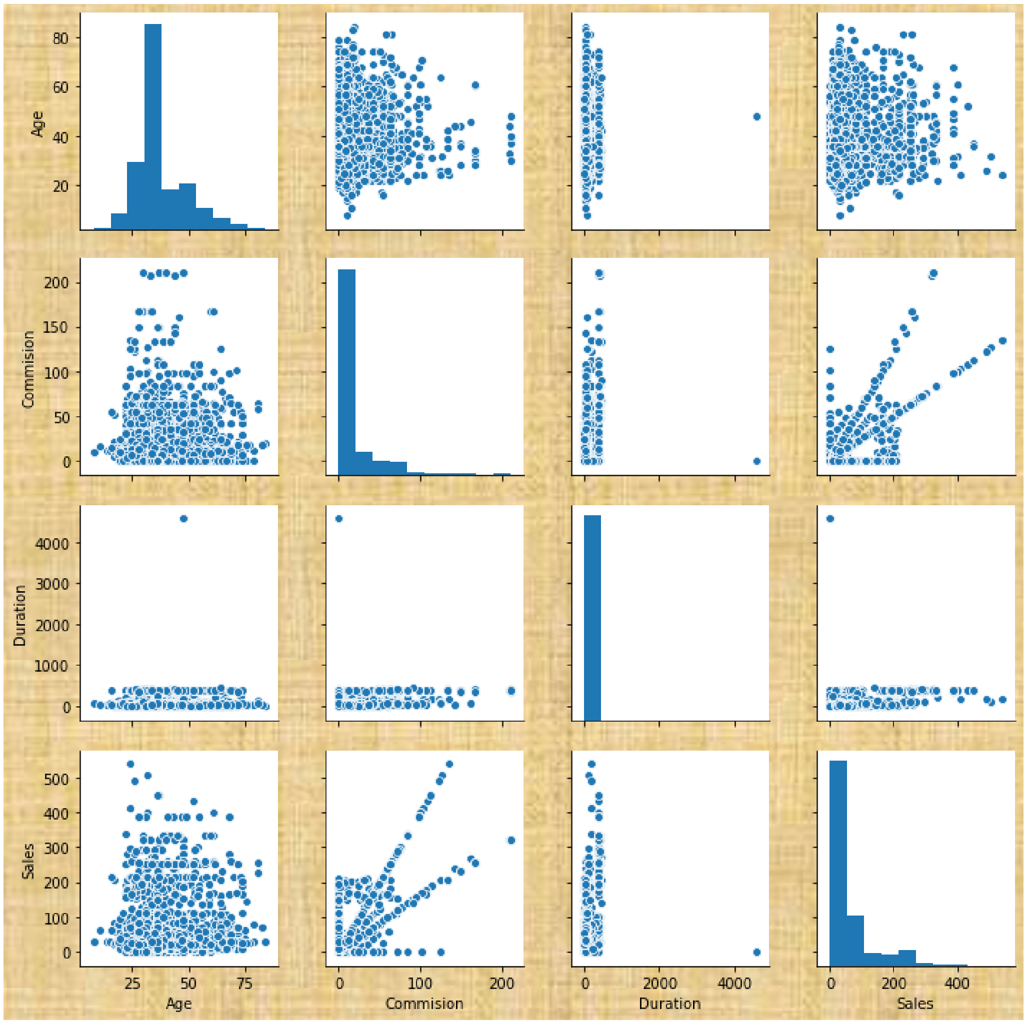

Age
Commision
Duration
Sales
Product Name
Customised Plan
Cancellation Plan
Bronze Plan
Silver Plan
Gold Plan


Age
Commision
Duration
Sales
Agency_Code
C2B
EPX
CWT
JZI

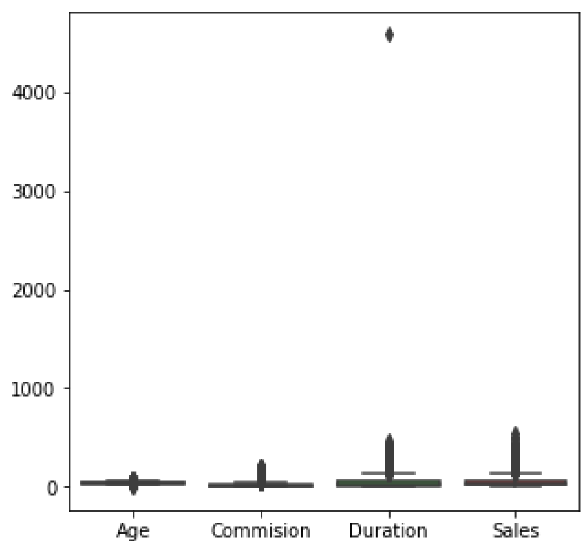

Age
Commision
Duration
Sales
Claimed
No
Yes


Age
Commision
Duration
Sales
Type
Airlines
Travel Agency

Pairplot()



Boxplot()



Boxplot shows outliers are associated with all variables.

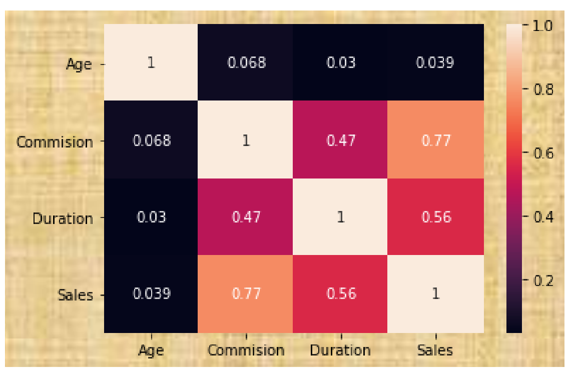
Crosstab:

Destination	ASIA					Americas					EUROPE					All
Product Name	**Bronze Plan**	**Cancellation Plan**	**Customised Plan**	**Gold Plan**	**Silver Plan**	**Bronze Plan**	**Cancellation Plan**	**Customised Plan**	**Gold Plan**	**Silver Plan**	**Bronze Plan**	**Cancellation Plan**	**Customised Plan**	**Gold Plan**	**Silver Plan**	
Claimed																
No	379	528	634	31	119	10	65	149	6	2	10	42	99	2	0	2076
Yes	243	30	143	56	302	6	7	61	11	3	2	6	50	3	1	924
All	622	558	777	87	421	16	72	210	17	5	12	48	149	5	1	3000

Correlation:

	Age	Commision	Duration	Sales
Age	1	0.067717	0.030425	0.039455
Commision	0.067717	1	0.471389	0.766505
Duration	0.030425	0.471389	1	0.55893
Sales	0.039455	0.766505	0.55893	1

Correlation heatmap:



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

Ans:

Imported liabraries required for splitting.

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=.30, random_state=1)

70 % data is used for training and 30 % for testing

A.CART

1. Model:

dt_model = DecisionTreeClassifier (criterion = 'gini')

2. fit model to train and test data:

dt _model.fit(x_train, y_train)

3.Preparing world file for tree in webgraphviz:

from sklearn import tree

train_char_label = ['No', 'Yes']

ins_Tree_File = open('d:\ins_tree.dot','w')

dot_data = tree.export_graphviz(dt_model, out_file=ins_Tree_File, feature_names = list(x_train), class_names = list(train_char_label))

ins_Tree_File.close()

4. Finding best pruning parameters of tree using grid search:

from sklearn.model_selection import GridSearchCV

param_grid = {

    'max_depth': [10,11,12,13],

    'min_samples_leaf': [15, 20, 25],

    'min_samples_split': [45, 60, 75]

}

dt_model = DecisionTreeClassifier()

grid_search = GridSearchCV(estimator = dt_model, param_grid = param_grid, cv = 3)

5. Best pruning parameters

grid_search.best_params_

{'max_depth': 10, 'min_samples_leaf': 25, 'min_samples_split': 60}

6. best_grid = grid_search.best_estimator_

7. Apply best grid to train and test data to get predicted test and train data:

ytrain_predict = best_grid.predict(x_train)

ytest_predict = best_grid.predict(x_test)

8. Check performance of model.

B. Random Forest

1.Build model for random forest:

```
from sklearn.model_selection import GridSearchCV
param_grid = {
    'max_depth': [10, 11,12],
    'max_features': [5,6,7],
    'min_samples_leaf': [20, 25],
    'min_samples_split': [60, 75],
    'n_estimators': [101, 301]
}
rfcl = RandomForestClassifier()
grid_search = GridSearchCV(estimator = rfcl, param_grid = param_grid, cv = 3)
```

2.Fit data to x and y training data:

grid_search.fit(x_trains, y_train)

3.Best parameters:

grid_search.best_params_

```
{'max_depth': 11,
 'max_features': 5,
 'min_samples_leaf': 20,
 'min_samples_split': 60,
 'n_estimators': 101}
```

4.Apply best grid to training data

best_grid = grid_search.best_estimator_

5. Classification report and performance checking.

C.ANN

1.Scale training and test data:

x_trains = sc.fit_transform(x_train)

x_tests = sc.transform (x_test)

2.Classifier:

```
param_grid = {
    'hidden_layer_sizes': [(100,100,100)],
    'activation': ['logistic', 'relu'],
    'solver': ['sgd', 'adam'],
    'tol': [0.1,0.01],
    'max_iter' : [10000]
}
rfcl = MLPClassifier()
grid_search = GridSearchCV(estimator = rfcl, param_grid = param_grid, cv = 3)
```

3. Apply grid search to x_test and x_train data

best_grid = grid_search.best_estimator_

4. Predict for x_test and x_train

```
ytrain_predict = best_grid.predict(x_trains)
ytest_predict = best_grid.predict(x_tests)
```

5. Classification report and performance of model.

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.

1. Decision tree

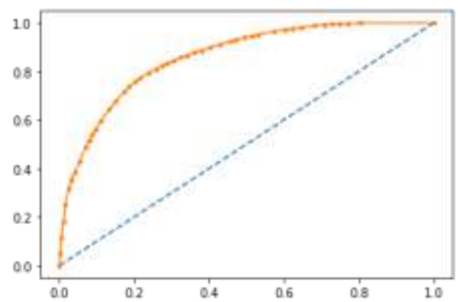
1. Train Data:

Best parameters:

{'max_depth': 10, 'min_samples_leaf': 25, 'min_samples_split': 60}

	precision	recall	f1-score	support
0	0.83	0.90	0.86	1471
1	0.70	0.57	0.63	629
accuracy			0.80	2100
macro avg	0.77	0.73	0.75	2100
weighted avg	0.79	0.80	0.79	2100

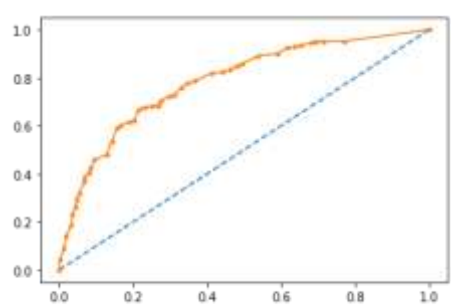
AUC: 0.859



2. Test Data:

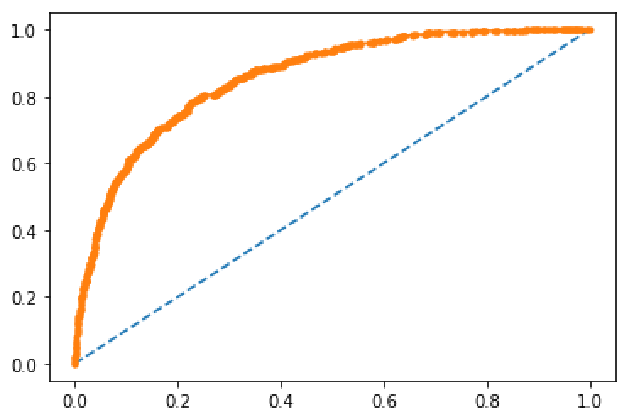
	precision	recall	f1-score	support
0	0.77	0.92	0.84	605
1	0.72	0.44	0.54	295
accuracy			0.76	900
macro avg	0.75	0.68	0.69	900
weighted avg	0.75	0.76	0.74	900

AUC: 0.785

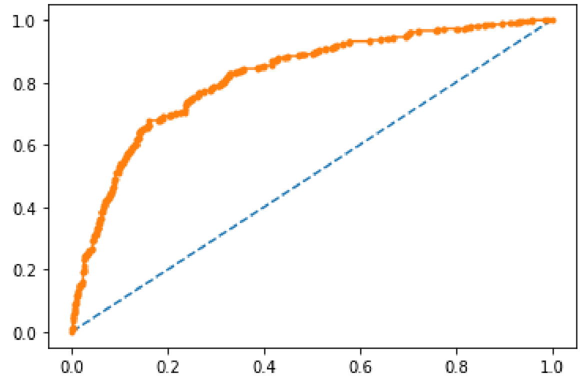


2. Random forest

```
Best parameters
{'max_depth': 11,
 'max_features': 5,
 'min_samples_leaf': 20,
 'min_samples_split': 75,
 'n_estimators': 101}

Train Data:


              precision    recall  f1-score   support

           0       0.83      0.91      0.87      1471
           1       0.72      0.58      0.64       629

    accuracy                           0.81      2100
   macro avg       0.78      0.74      0.76      2100
weighted avg       0.80      0.81      0.80      2100

AUC: 0.856
```




```
Test Data:
              precision    recall  f1-score   support

           0       0.78      0.92      0.84       605
           1       0.73      0.46      0.57       295

    accuracy                           0.77       900
   macro avg       0.75      0.69      0.70       900
weighted avg       0.76      0.77      0.75       900

AUC: 0.820
```



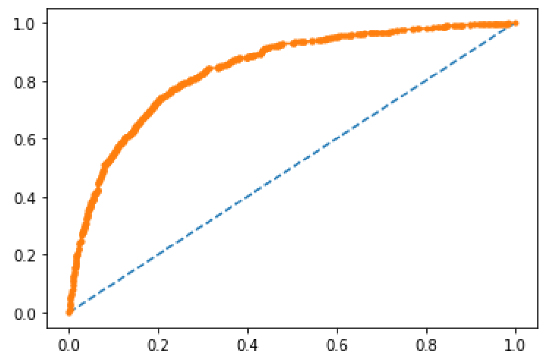
3.MLP Classifier

```
Best Parameters:
{'activation': 'relu',
 'hidden_layer_sizes': (100, 100, 100),
 'max_iter': 10000,
 'solver': 'adam',
 'tol': 0.01}
```

A.Train data:

	precision	recall	f1-score	support
0	0.81	0.91	0.86	1471
1	0.72	0.51	0.60	629
accuracy			0.79	2100
macro avg	0.77	0.71	0.73	2100
weighted avg	0.79	0.79	0.78	2100

AUC:0.845

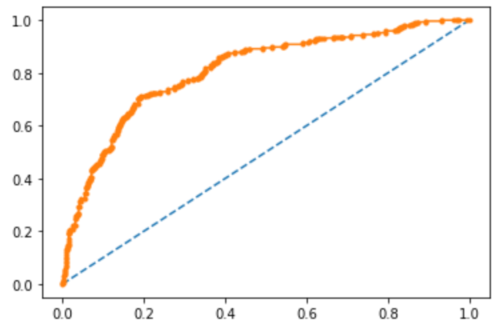


B.Test Data:

	precision	recall	f1-score	support
0	0.77	0.93	0.84	605
1	0.74	0.43	0.54	295

```
    accuracy                           0.76       900
   macro avg       0.75      0.68      0.69       900
weighted avg       0.76      0.76      0.74       900
```

```
AUC:0.812
```



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

Ans:

Type		precision	recall	f1-score	AUC
Decision trtee	Train	0.70	0.57	0.63	0.859
	Test	0.72	0.44	0.54	0.785
Random forest	Train	0.72	0.58	0.64	0.856
	Test	0.73	0.46	0.57	0.820
MLP classifier	Train	0.72	0.51	0.60	0.845
	Test	0.74	0.43	0.54	0.812

Best/Optimized model is **Random forest.**

Because all parameters (precision/recall/f1 score) are not varying much. The values for AUC are maximum for train data and test data are 0.856 and 0.820.

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

Ans:

Random forest model can be used for predicting claim status of a particular customer with 82% of correctness.

Random forest algorithm has following advantages:

Classifications and regression both can be done using random forest.

Gives higher accuracy.

Random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data.

It won’t allow over fitting trees in the model.

It has the power to handle a large data set with higher dimensionality.