DATA MINING
BUSINESS REPORT

----------------------******-----------------------

BY:- ANIKET HIRGUDE

INDEX

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

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. Explain the results properly. Interpret and write inferences on the finalized clusters.

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

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

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

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.

2.4 Final Model: Compare all the models 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

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

Data Dictionary for Market Segmentation:

1. *spending: Amount spent by the customer per month (in 1000s)*
2. *advance_payments: Amount paid by the customer in advance by cash (in 100s)*
3. *probability_of_full_payment: Probability of payment done in full by the customer to the bank*
4. *current_balance: Balance amount left in the account to make purchases (in 1000s)*
5. *credit_limit: Limit of the amount in credit card (10000s)*
6. *min_payment_amt : minimum paid by the customer while making payments for purchases made monthly (in 100s)*
7. *max_spent_in_single_shopping: Maximum amount spent in one purchase (in 1000s)*

1.1 Read the data and do exploratory data analysis (3 pts). Describe the data briefly. Interpret the inferences for each (3 pts). Initial steps like head() .info(), Data Types, etc . Null value check. Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Summary stats, Skewness, Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.

:--

	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

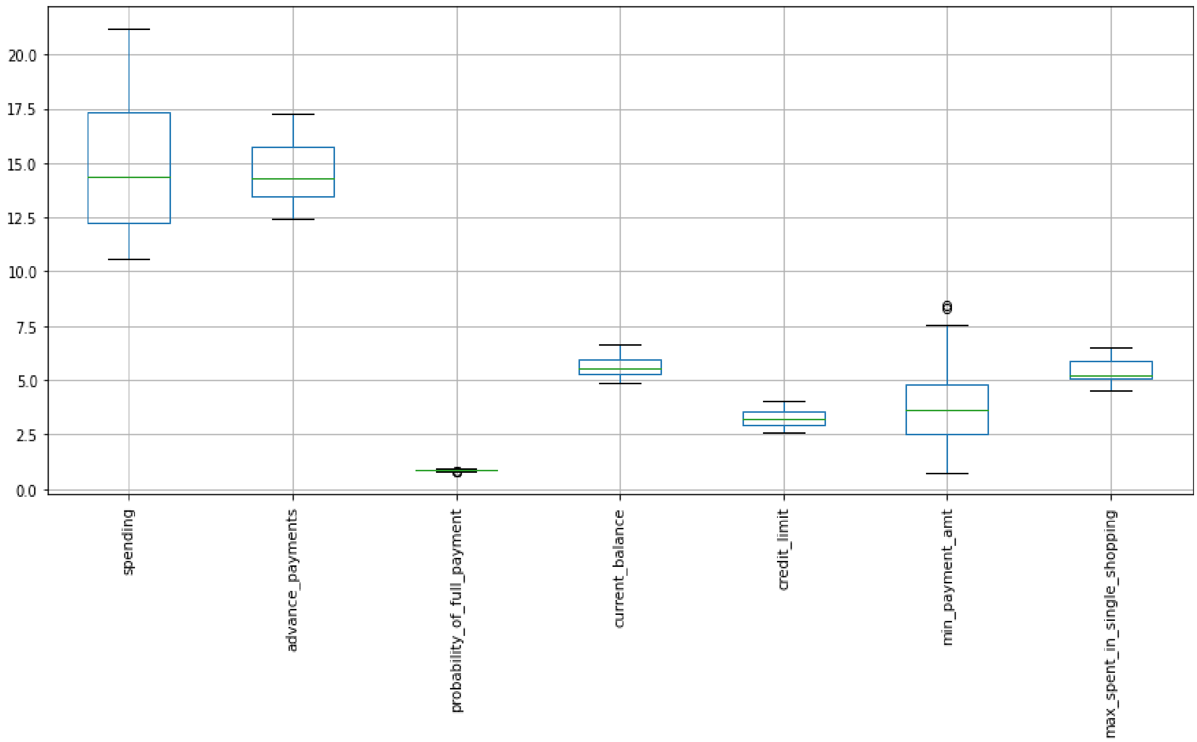
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   spending                      210 non-null    float64
 1   advance_payments              210 non-null    float64
 2   probability_of_full_payment   210 non-null    float64
 3   current_balance               210 non-null    float64
 4   credit_limit                  210 non-null    float64
 5   min_payment_amt               210 non-null    float64
 6   max_spent_in_single_shopping  210 non-null    float64
dtypes: float64(7)
memory usage: 11.6 KB
```

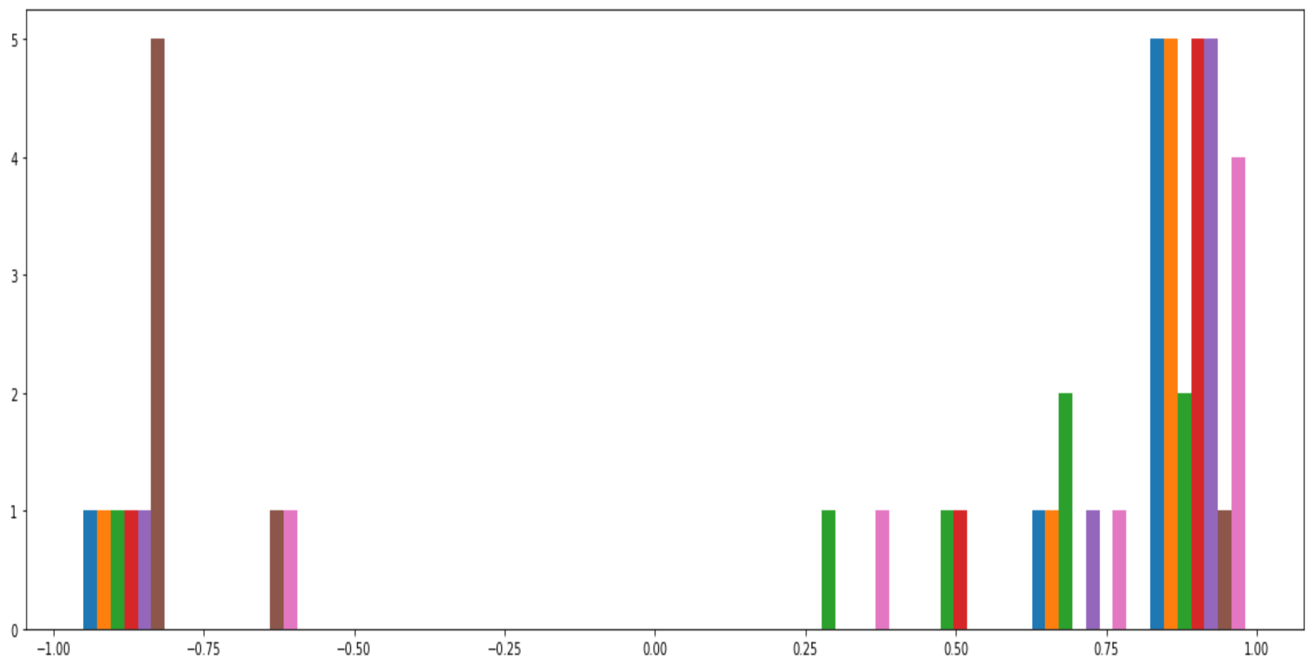
---

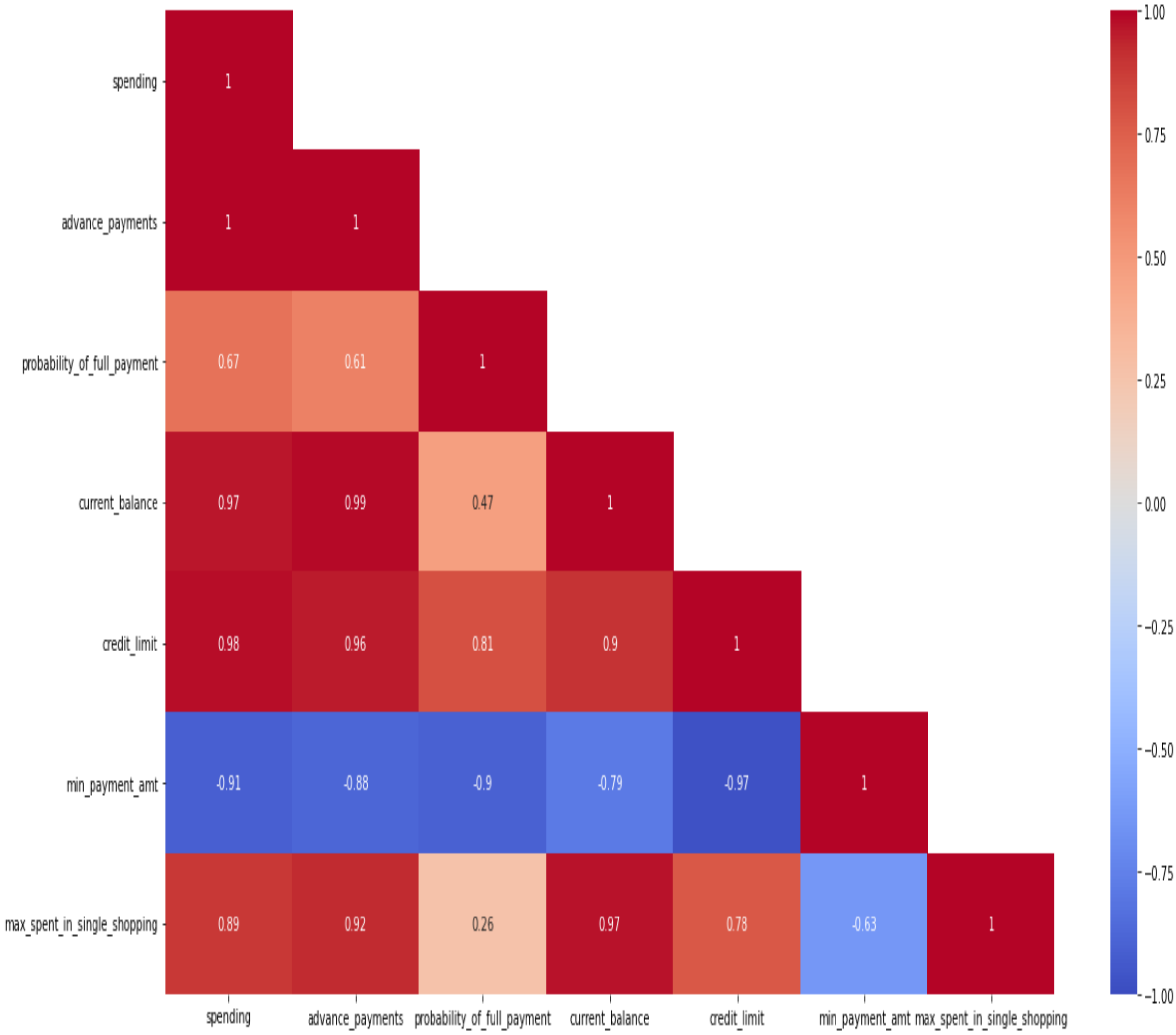
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000	210.000000
mean	14.847524	14.559286	0.870999	5.628533	3.258605	3.700201	5.408071
std	2.909699	1.305959	0.023629	0.443063	0.377714	1.503557	0.491480

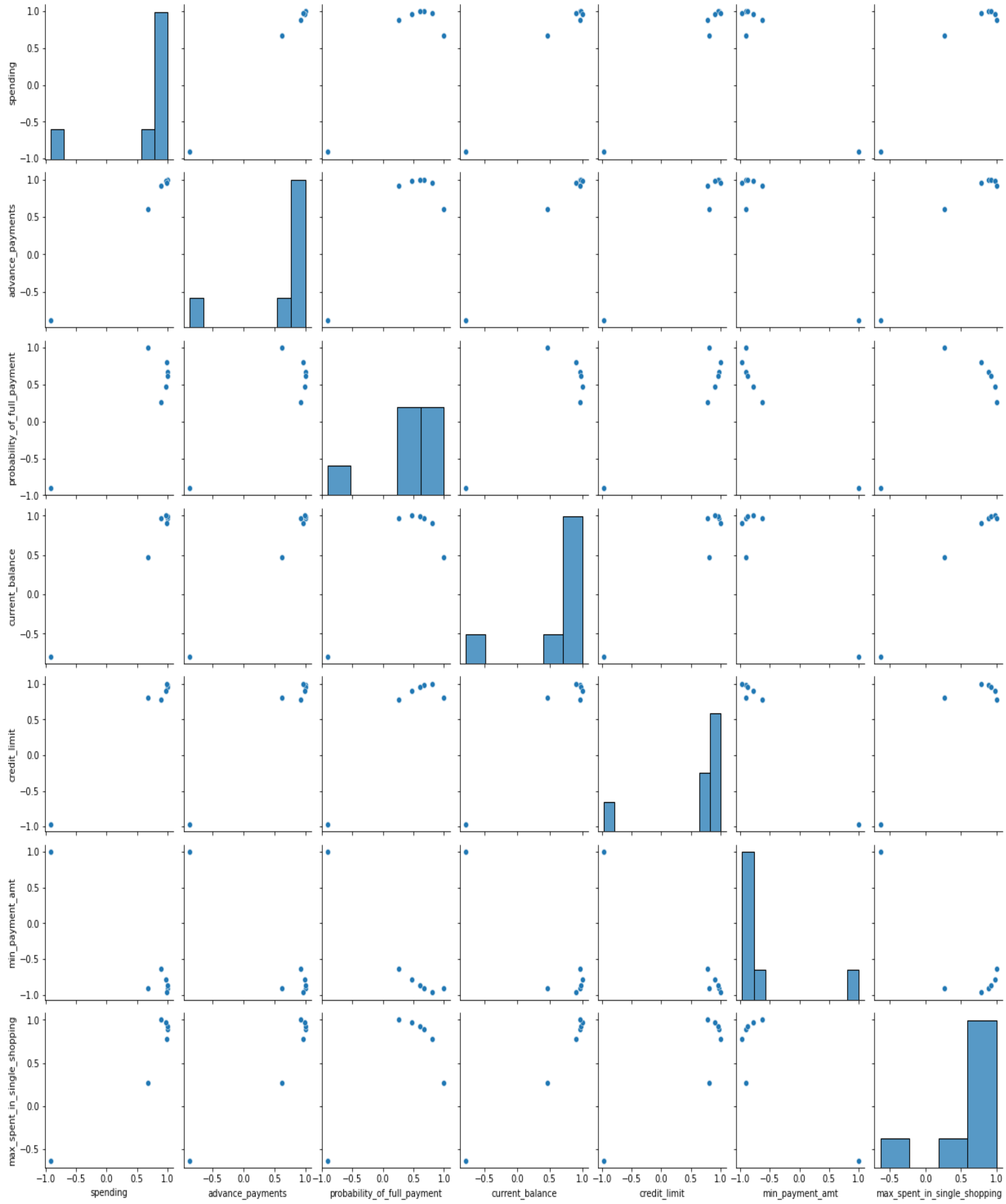
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
min	10.590000	12.410000	0.808100	4.899000	2.630000	0.765100	4.519000
25%	12.270000	13.450000	0.856900	5.262250	2.944000	2.561500	5.045000
50%	14.355000	14.320000	0.873450	5.523500	3.237000	3.599000	5.223000
75%	17.305000	15.715000	0.887775	5.979750	3.561750	4.768750	5.877000
max	21.180000	17.250000	0.918300	6.675000	4.033000	8.456000	6.550000

```
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
```


20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
spending
advance_payments
probability_of_full_payment
current_balance
credit_limit
min_payment_amt
max_spent_in_single_shopping


5
4
3
2
1
0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


spending
advance_payments
probability_of_full_payment
current_balance
credit_limit
min_payment_amt
max_spent_in_single_shopping
1
1
1
0.67
0.61
1
0.97
0.99
0.47
1
0.98
0.96
0.81
0.9
1
-0.91
-0.88
-0.9
-0.79
-0.97
1
0.89
0.92
0.26
0.97
0.78
-0.63
1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


spending
advance_payments
probability_of_full_payment
current_balance
credit_limit
min_payment_amt
max_spent_in_single_shopping

1.2 Do you think scaling is necessary for clustering in this case? Justify The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for example std dev, variance, etc. Should justify whether there is a necessity for scaling and which method is he/she using to do the scaling. Can also comment on how that method works.

:--

```
## Scaling the data
```

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
spending	0.529356	0.531655	0.441078	0.532221	0.519821	-0.501083	0.521327
advance_payments	0.524191	0.536902	0.333344	0.560191	0.490386	-0.447525	0.582262
probability_of_full_payment	0.025858	-0.072691	1.002730	-0.287385	0.256203	-0.491734	-0.605056
current_balance	0.480869	0.514342	0.089652	0.583829	0.398605	-0.311786	0.673809
credit_limit	0.498791	0.475419	0.669331	0.419983	0.549850	-0.590106	0.327086

```
# the StandardScaler Module
```

	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.811968	0.178230	2.367533	1.338579	-0.298806	2.328998
1	0.253840	1.501773	-0.600744	0.858236	-0.242805	-0.538582
2	1.428192	0.504874	1.401485	1.317348	-0.221471	1.509107
3	-1.227533	-2.591878	-0.793049	-1.639017	0.987884	-0.454961
4	0.998364	1.196340	0.591544	1.155464	-1.088154	0.874813
...	...	...	...	...	...	...
205	-0.413929	0.721222	-0.428801	-0.158181	0.190536	-1.366631
206	0.814152	-0.305372	0.675253	0.476084	0.813214	0.789153
207	-0.306472	0.364883	-0.431064	-0.152873	-1.322158	-0.830235
208	0.338271	1.230277	0.182048	0.600814	-0.953484	0.071238
209	0.453403	-0.776248	0.659416	-0.073258	-0.706813	0.960473

1.3 Apply hierarchical clustering to scaled data (3 pts). Identify the number of optimum clusters using Dendrogram and briefly describe them (4). Students are expected to apply hierarchical clustering. It can be obtained via Fclusters or Agglomerative Clustering. Report should talk about the used criterion, affinity and linkage. Report must contain a Dendrogram and a logical reason behind choosing the optimum number of clusters and Inferences on the dendrogram. Customer segmentation can be visualized using limited features or whole data but it should be clear, correct and logical. Use appropriate plots to visualize the clusters.

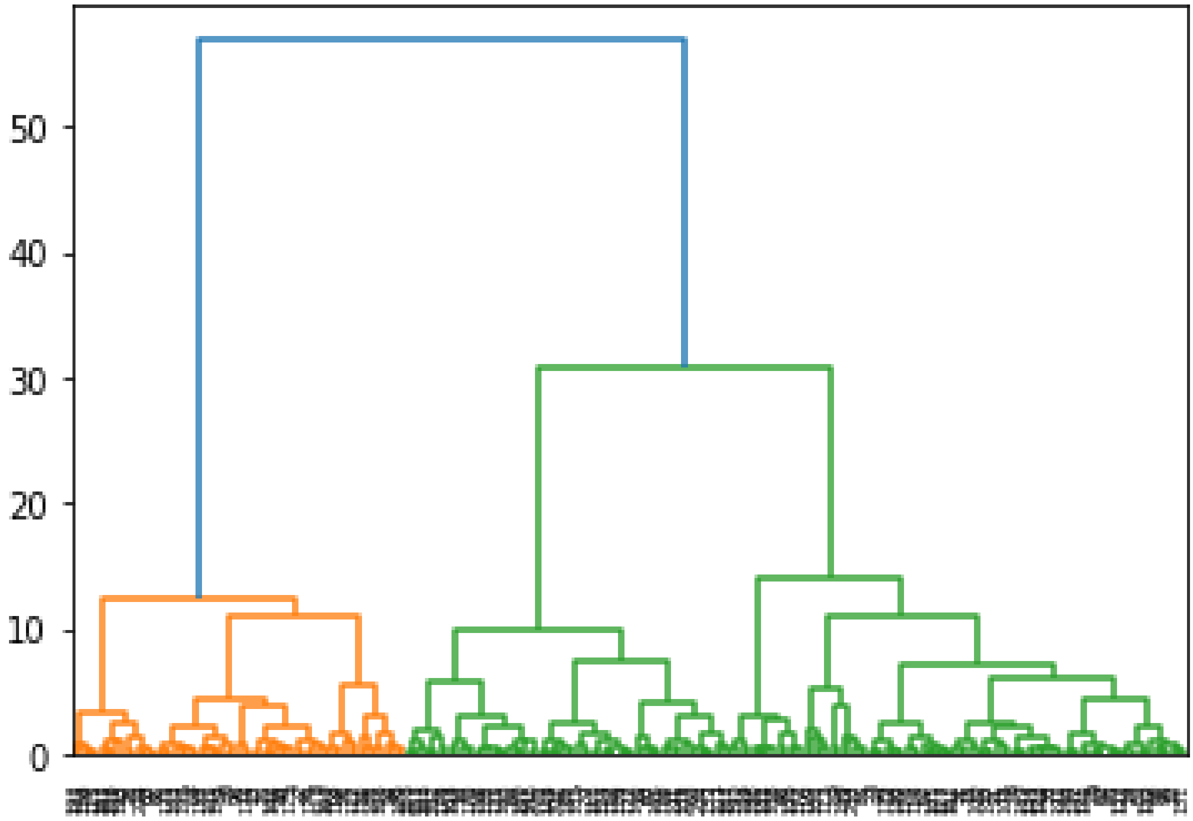
:--

iloc

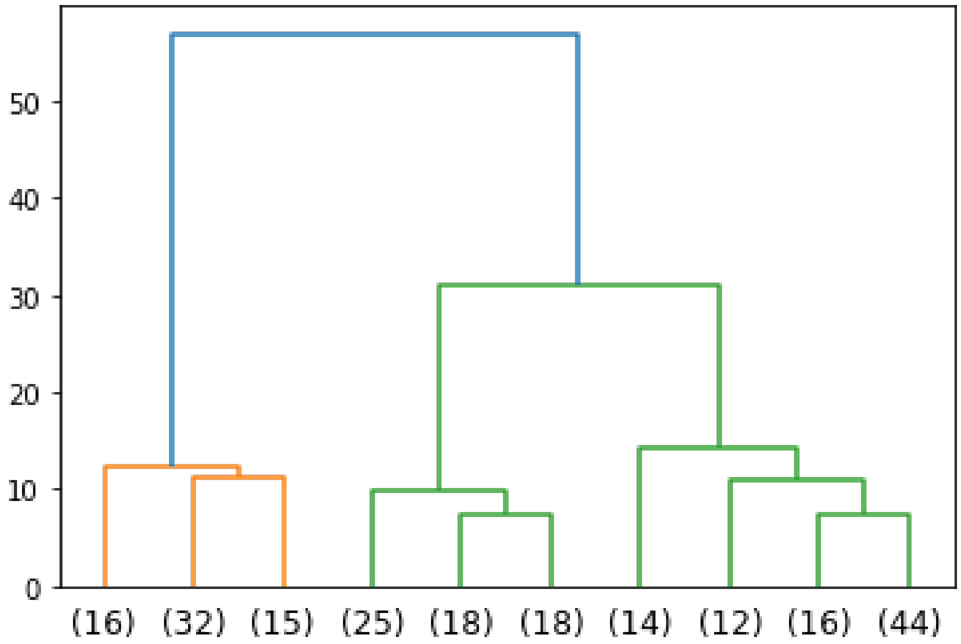
	advance_payments	probability_of_full_payment	current_balance	credit_limit
count	210.000000	210.000000	210.000000	210.000000
mean	14.559286	0.870999	5.628533	3.258605
std	1.305959	0.023629	0.443063	0.377714
min	12.410000	0.808100	4.899000	2.630000
25%	13.450000	0.856900	5.262250	2.944000
50%	14.320000	0.873450	5.523500	3.237000
75%	15.715000	0.887775	5.979750	3.561750
max	17.250000	0.918300	6.675000	4.033000

New table

	advance_payments	probability_of_full_payment	current_balance	credit_limit
0	16.92	0.8752	6.675	3.763
1	14.89	0.9064	5.363	3.582
2	16.42	0.8829	6.248	3.755
3	12.96	0.8099	5.278	2.641
4	15.86	0.8992	5.890	3.694



P10



Fclusters

Method 1

```
Fcluster (wardlink, 3, criterion='maxclust')
```

```
array([1, 2, 1, 3, 1, 3, 3, 2, 1, 3, 1, 2, 3, 1, 3, 3, 2, 3, 3, 3, 3,
3, 1, 3, 2, 2, 3, 3, 3, 3, 2, 3, 3, 2, 3, 3, 3, 3, 3, 1, 1, 2, 1, 1, 3,
3, 3, 1, 1, 1, 3, 1, 1, 1, 1, 1, 3, 3, 3, 1, 2, 3, 3, 2, 2, 1, 1, 2, 1,
3, 2, 3, 1, 1, 3, 1, 2, 3, 1, 2, 2, 3, 2, 1, 3, 2, 2, 2, 1, 3, 3, 1, 2,
3, 3, 1, 1, 1, 3, 1, 3, 1, 2, 1, 2, 1, 1, 3, 3, 1, 2, 2, 1, 3, 3, 1, 2,
3, 3, 1, 3, 3, 3, 3, 2, 2, 1, 3, 2, 2, 3, 2, 3, 1, 3, 1, 1, 3, 1, 2, 2,
2, 3, 3, 3, 3, 1, 3, 2, 3, 2, 3, 2, 2, 3, 2, 3, 3, 2, 1, 1, 3, 1, 1, 1,
3, 2, 2, 3, 3, 2, 3, 2, 1, 1, 1, 2, 3, 2, 3, 2, 2, 3, 2, 2, 1, 3, 2, 3,
3, 3, 2, 3, 1, 2, 1, 1, 3, 1, 3, 2, 2, 2, 3, 1, 2, 1, 2, 2, 2],
dtype=int32)
```

method 2

```
fcluster (wardlink, 23, criterion='distance')

array([1, 2, 1, 3, 1, 3, 3, 2, 1, 3, 1, 2, 3, 1, 3, 3, 2, 3, 3, 3, 3,
3, 1, 3, 2, 2, 3, 3, 3, 3, 2, 3, 3, 2, 3, 3, 3, 3, 3, 1, 1, 2, 1, 1, 3,
3, 3, 1, 1, 1, 3, 1, 1, 1, 1, 1, 3, 3, 3, 1, 2, 3, 3, 2, 2, 1, 1, 2, 1,
3, 2, 3, 1, 1, 3, 1, 2, 3, 1, 2, 2, 3, 2, 1, 3, 2, 2, 2, 1, 3, 3, 1, 2,
3, 3, 1, 1, 1, 3, 1, 3, 1, 2, 1, 2, 1, 1, 3, 3, 1, 2, 2, 1, 3, 3, 1, 2,
3, 3, 1, 3, 3, 3, 3, 2, 2, 1, 3, 2, 2, 3, 2, 3, 1, 3, 1, 1, 3, 1, 2, 2,
2, 3, 3, 3, 3, 1, 3, 2, 3, 2, 3, 2, 2, 3, 2, 3, 3, 2, 1, 1, 3, 1, 1, 1,
3, 2, 2, 3, 3, 2, 3, 2, 1, 1, 1, 2, 3, 2, 3, 2, 2, 3, 2, 2, 1, 3, 2, 3,
3, 3, 2, 3, 1, 2, 1, 1, 3, 1, 3, 2, 2, 2, 3, 1, 2, 1, 2, 2, 2],
dtype=int32)
```

1.4 Apply K-Means clustering on scaled data and determine optimum clusters (2 pts). Apply elbow curve and silhouette score (3 pts). Interpret the inferences from the model (2.5 pts). K-means clustering code application with different number of clusters. Calculation of WSS(inertia for each value of k) Elbow Method must be applied and visualized with different values of K. Reasoning behind the selection of the optimal value of K must be explained properly. Silhouette Score must be calculated for the same values of K taken above and commented on. Report must contain logical and correct explanations for choosing the optimum clusters using both elbow method and silhouette scores. Append cluster labels obtained from K-means clustering into the original data frame. Customer Segmentation can be visualized using appropriate graphs.

:--

```
# Get the labels
```

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

```
k_means.inertia
1011.712345315119
```

```
Cluster for 1,3,4,5,6
```

1259.9999999999998

398.47257138104794

349.74033704424596

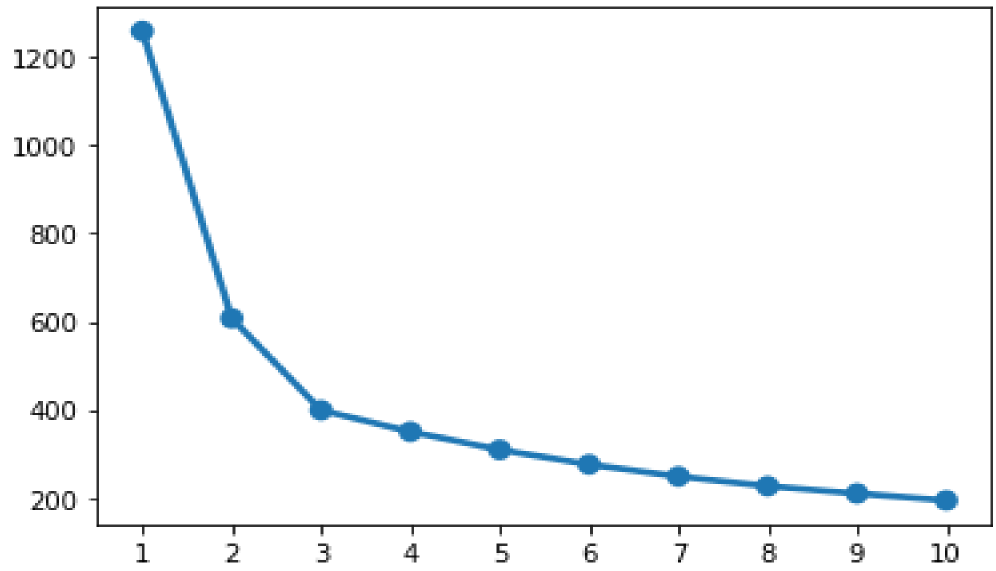
308.6442514797732

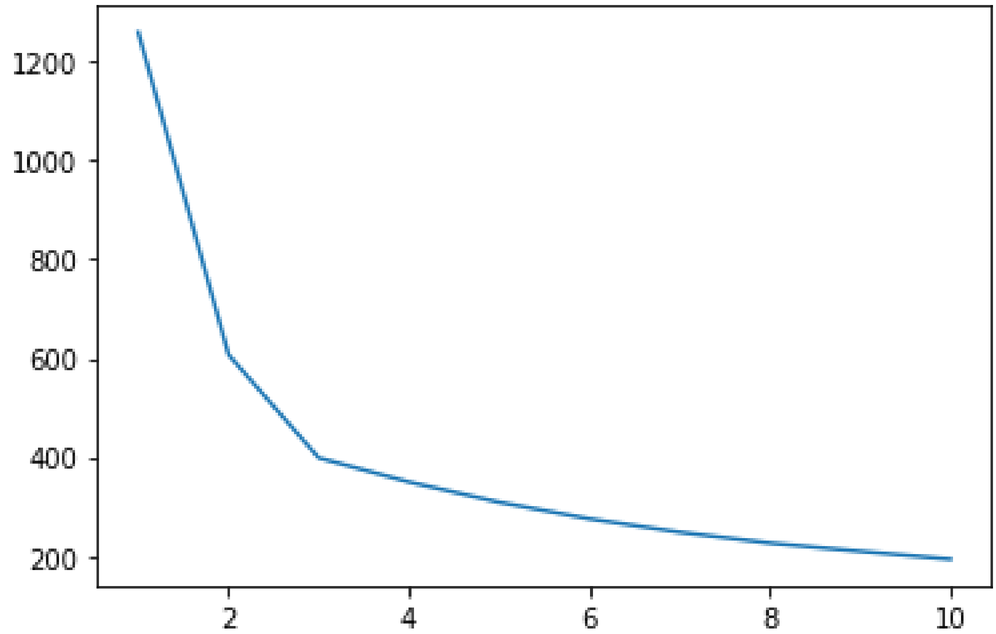
275.11057239157833

Calculating WSS for other values of K - Elbow Method

wss

```
[1259.9999999999998, 607.2643170652891, 398.47257138104794,
349.74033704424596, 308.6442514797732, 275.11057239157833,
248.01182685031293, 225.98173591960415, 209.29194926314042,
193.42428405885454]
```





the silhouette score

for cluster 3

0.4001619756799544

0.3235236917702016

---

silhouette score is better for 3 clusters than for 4 clusters. So, the final clusters will be 3

1.5 Describe cluster profiles for the clusters defined (2.5 pts). Recommend different promotional strategies for different clusters in context to the business problem in-hand (2.5 pts). After adding the final clusters to the original dataframe, do the cluster profiling. Divide the data in the finalyzed groups and check their means. Explain each of the group briefly. There should be at least 3-4 Recommendations. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks will only be allotted if the recommendations are correct and business specific. variable means. Students to explain the profiles and suggest a mechanism to approach each cluster. Any logical explanation is acceptable.

:--

```
# Appending Clusters to the original dataset
```

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Clus_kmeans3
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	0
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	0
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	1
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837	0

```
0 50 1 64 2 65 3 31 Name: Clus_kmeans3, dtype: int64
```

Problem 2: CART-RF-ANN

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

Attribute Information:

1. Target: Claim Status (Claimed)
2. Code of tour firm (Agency_Code)
3. Type of tour insurance firms (Type)
4. Distribution channel of tour insurance agencies (Channel)
5. Name of the tour insurance products (Product)
6. Duration of the tour (Duration in days)
7. Destination of the tour (Destination)
8. Amount worth of sales per customer in procuring tour insurance policies in rupees (in 100's)
9. The commission received for tour insurance firm (Commission is in percentage of sales)
10.Age of insured (Age)

2.1 Read the data and do exploratory data analysis (4 pts). Describe the data briefly. Interpret the inferences for each (2 pts). Initial steps like head() .info(), Data Types, etc . Null value check. Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Summary stats, Skewness, Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.

:--

	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

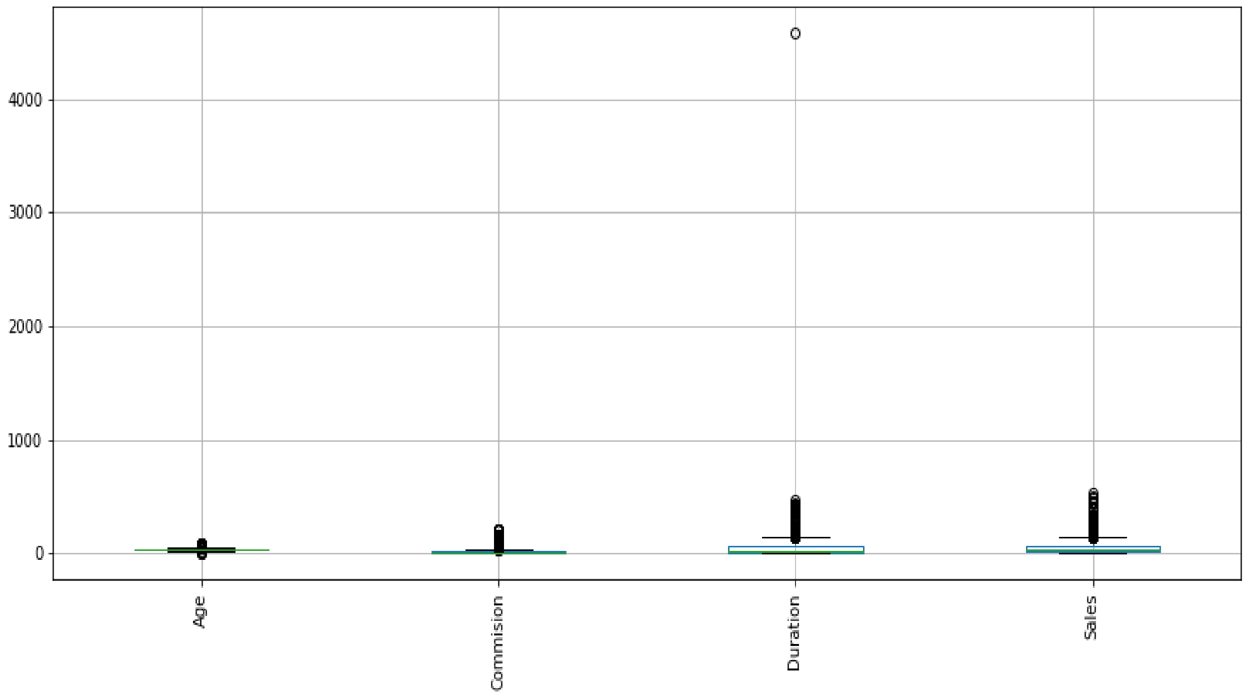
After object to int (`Categorical)`

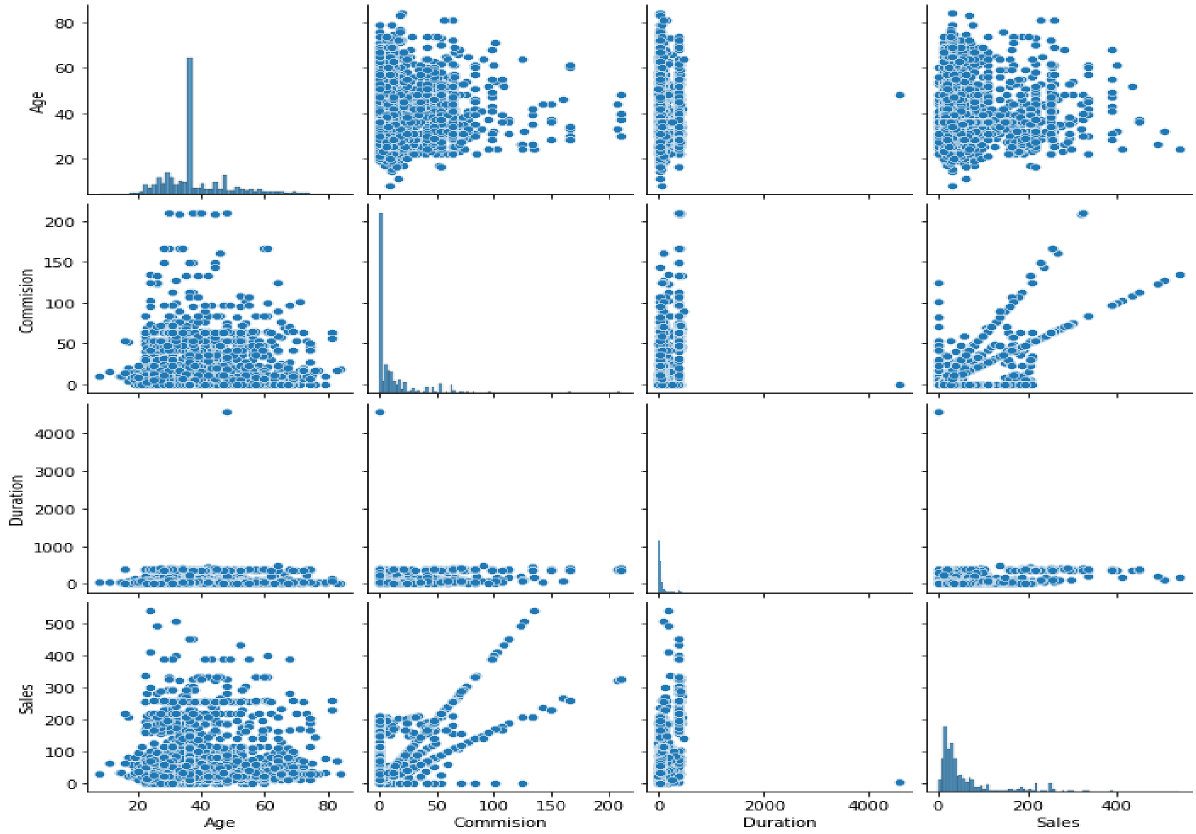
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Agency_Code   3000 non-null   int8   
 1   Type          3000 non-null   int8   
 2   Claimed       3000 non-null   int8   
 3   Commision     3000 non-null   float64
 4   Channel       3000 non-null   int8   
 5   Duration      3000 non-null   int64  
 6   Sales         3000 non-null   float64
 7   Product Name  3000 non-null   int8   
 8   Destination   3000 non-null   int8   
dtypes: float64(2), int64(1), int8(6)
memory usage: 88.0 KB
```

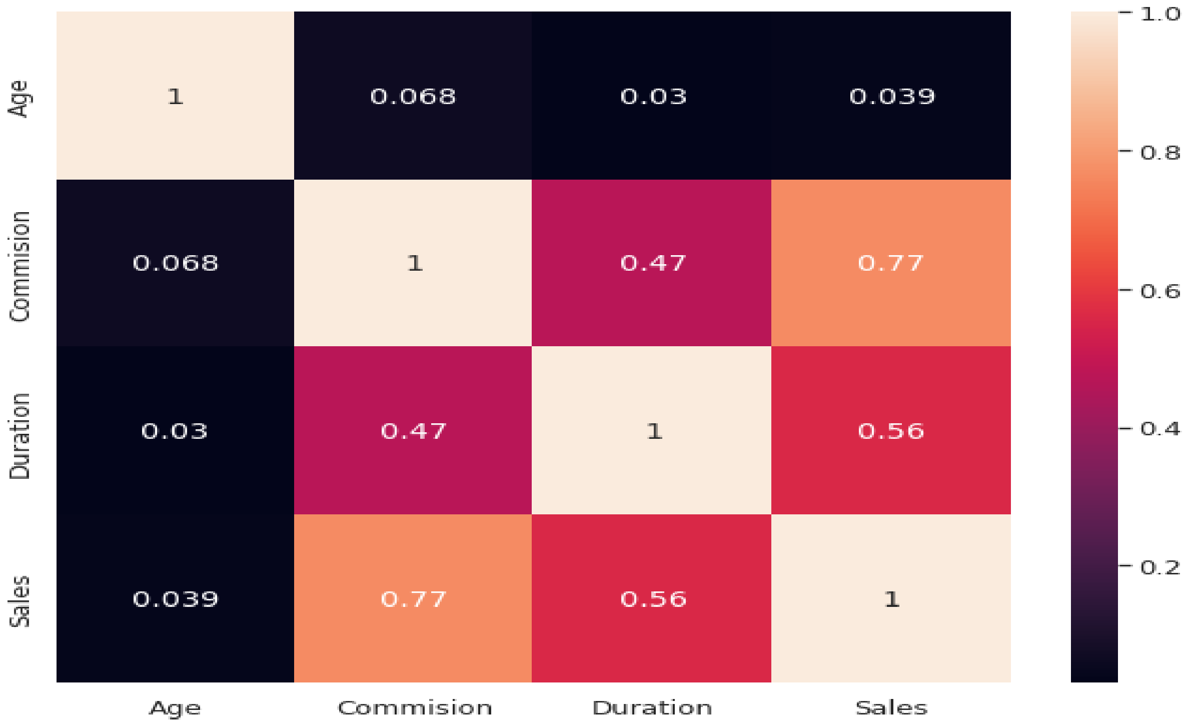
	Age	Commision	Duration	Sales
count	3000.000000	3000.000000	3000.000000	3000.000000
mean	38.091000	14.529203	70.001333	60.249913
std	10.463518	25.481455	134.053313	70.733954
min	8.000000	0.000000	-1.000000	0.000000
25%	32.000000	0.000000	11.000000	20.000000
50%	36.000000	4.630000	26.500000	33.000000
75%	42.000000	17.235000	63.000000	69.000000
max	84.000000	210.210000	4580.000000	539.000000

Data type

```
0 Airlines 1 Travel Agency 2 Travel Agency 3 Travel Agency 4 Airlines
... 2995 Travel Agency 2996 Airlines 2997 Travel Agency 2998 Airlines
2999 Airlines Name: Type, Length: 3000, dtype: object
```




Age
Commision
Duration
Sales


Age
Commision
Duration
Sales
1
0.068
0.03
0.039
0.47
0.77
0.56
1.0
0.8
0.6
0.4
0.2

2.2 Data Split: Split the data into test and train(1 pts), build classification model CART (1.5 pts), Random Forest (1.5 pts), Artificial Neural Network(1.5 pts). Object data should be converted into categorical/numerical data to fit in the models. (pd.categorical().codes(), pd.get_dummies(drop_first=True)) Data split, ratio defined for the split, train-test split should be discussed. Any reasonable split is acceptable. Use of random state is mandatory. Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Apply grid search for each model and make models on best_params. Feature importance for each model.

:--

Droping the (Product Name) Xhead

	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Destination
0	0	0	0	0.70	1	7	2.51	0
1	2	1	0	0.00	1	34	20.00	0
2	1	1	0	5.94	1	3	9.90	1
3	2	1	0	0.00	1	4	26.00	0
4	3	0	0	6.30	1	53	18.00	0

Yhead

```
0    2
1    2
2    2
3    1
4    0
Name: Product Name, dtype: int8
```

Split

```
X_train (2100, 8)
X_test (900, 8)
train_labels (2100,)
test_labels (900,)
```

grid search

{'criterion': 'gini', 'max_depth': 10, 'min_samples_leaf': 250, 'min_samples_split': 750}
DecisionTreeClassifier(max_depth=10, min_samples_leaf=250,
min_samples_split=750)

```
GridSearchCV(cv=10, estimator=RandomForestClassifier(),
param_grid={'max_depth': [10], 'max_features': [5], 'min_samples_leaf':
[250], 'min_samples_split': [750], 'n_estimators': [100]})
```

```
{'max_depth': 10, 'max_features': 5, 'min_samples_leaf': 250,
'min_samples_split': 750, 'n_estimators': 100}
```

```
DecisionTreeClassifier(max_depth=10, min_samples_leaf=250,
min_samples_split=750)
```

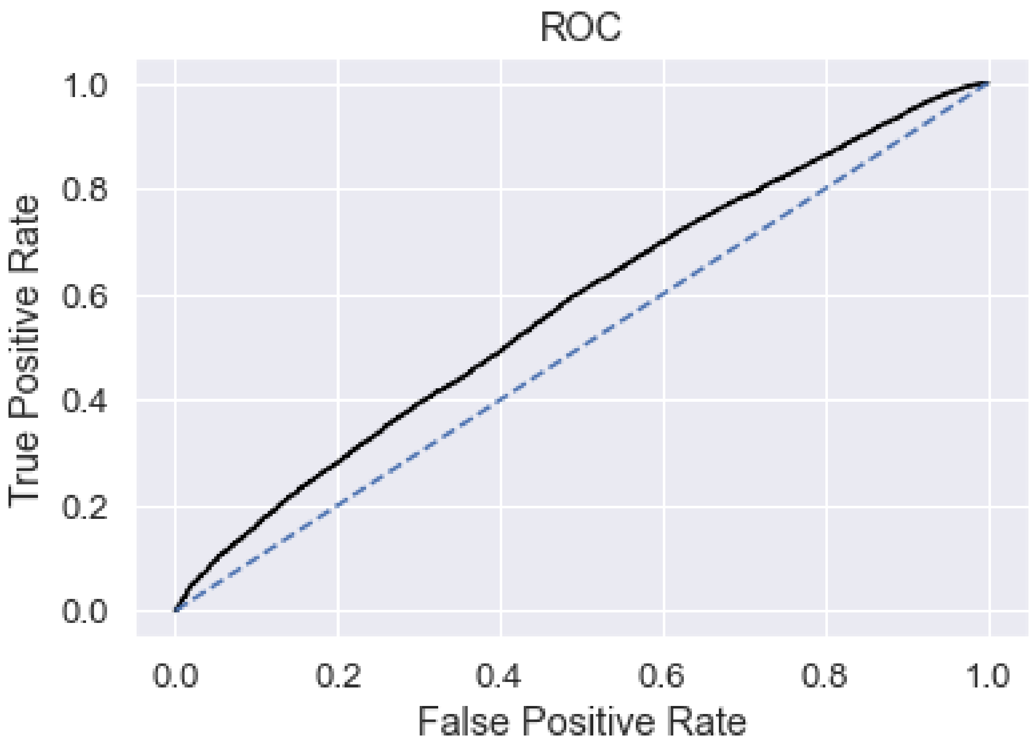
{'hidden_layer_sizes': 32, 'max_iter': 200, 'solver': 'adam'}

```
Best grid
```

MLPClassifier(early_stopping=True, hidden_layer_sizes=32, tol=0.01)

2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy (1 pts), Confusion Matrix (2 pts), Plot ROC curve and get ROC_AUC score for each model (2 pts), Make classification reports for each model. Write inferences on each model (2 pts). Calculate Train and Test Accuracies for each model. Comment on the validness of models (overfitting or underfitting) Build confusion matrix for each model. Comment on the positive class in hand. Must clearly show obs/pred in row/col Plot roc_curve for each model. Calculate roc_auc_score for each model. Comment on the above calculated scores and plots. Build classification reports for each model. Comment on f1 score, precision and recall, which one is important here.

```
:--
```




```
#Accuracy on the Training Data: 83%
Accuracy on the Test Data: 82%

AUC on the Training Data: 87.9%
AUC on the Test: 88.1%

Accuracy, AUC, Precision and Recall for test data is almost inline with
 training data.
This proves no overfitting or underfitting has happened, and overall th
e model is a good model for classification

FICO, term and gender (in same order of preference) are the most import
ant variables in determining if a borrower will get into a delinquent s
tage
```

2.4 Final Model - Compare all models on the basis of the performance metrics in a structured tabular manner (2.5 pts). Describe on which model is best/optimized (1.5 pts). A table containing all the values of accuracies, precision, recall, auc_roc_score, f1 score. Comparison between the different models(final) on the basis of above table values. After comparison which model suits the best for the problem in hand on the basis of different measures. Comment on the final model.

:--

Confusion matrix train

```
array([[  0,   6,   4, 420,   9],
       [  0, 452,  17,   0,   1],
       [  0, 456,  38, 289,  19],
       [  0,   0,   0,  82,   0],
       [  0,   3,   1, 300,   3]])
```

Precision, Recall, F1-score (train)

```
precision    recall  f1-score   support

           0       0.00      0.00      0.00       439
           1       0.49      0.96      0.65       470
           2       0.63      0.05      0.09       802
           3       0.08      1.00      0.14        82
           4       0.09      0.01      0.02       307

    accuracy                           0.27      2100
   macro avg       0.26      0.40      0.18      2100
weighted avg       0.37      0.27      0.19      2100
```

nn_train_precision 0.49
nn_train_recall 0.96
nn_train_f1 0.65

Confusion matrix (test)

```
array([[ 0, 5, 3, 200, 3],
 [ 0, 200, 7, 0, 1],
[ 0, 216, 12, 101, 5],
[ 0, 0, 0, 27, 0],
[ 0, 0, 0, 119, 1]])
```

Precision, Recall, F1-score (test)

```
precision    recall  f1-score   support

           0       0.00      0.00      0.00       439
           1       0.49      0.96      0.65       470
           2       0.63      0.05      0.09       802
           3       0.08      1.00      0.14        82
           4       0.09      0.01      0.02       307

    accuracy                           0.27      2100
   macro avg       0.26      0.40      0.18      2100
weighted avg       0.37      0.27      0.19      2100
```

```
nn_test_precision  0.48
nn_test_recall  0.96
nn_test_f1  0.64
```

```
best grid score
```

<bound method ClassifierMixin.score of MLPClassifier(early_stopping=True, hidden_layer_sizes=32, tol=0.01)>

***