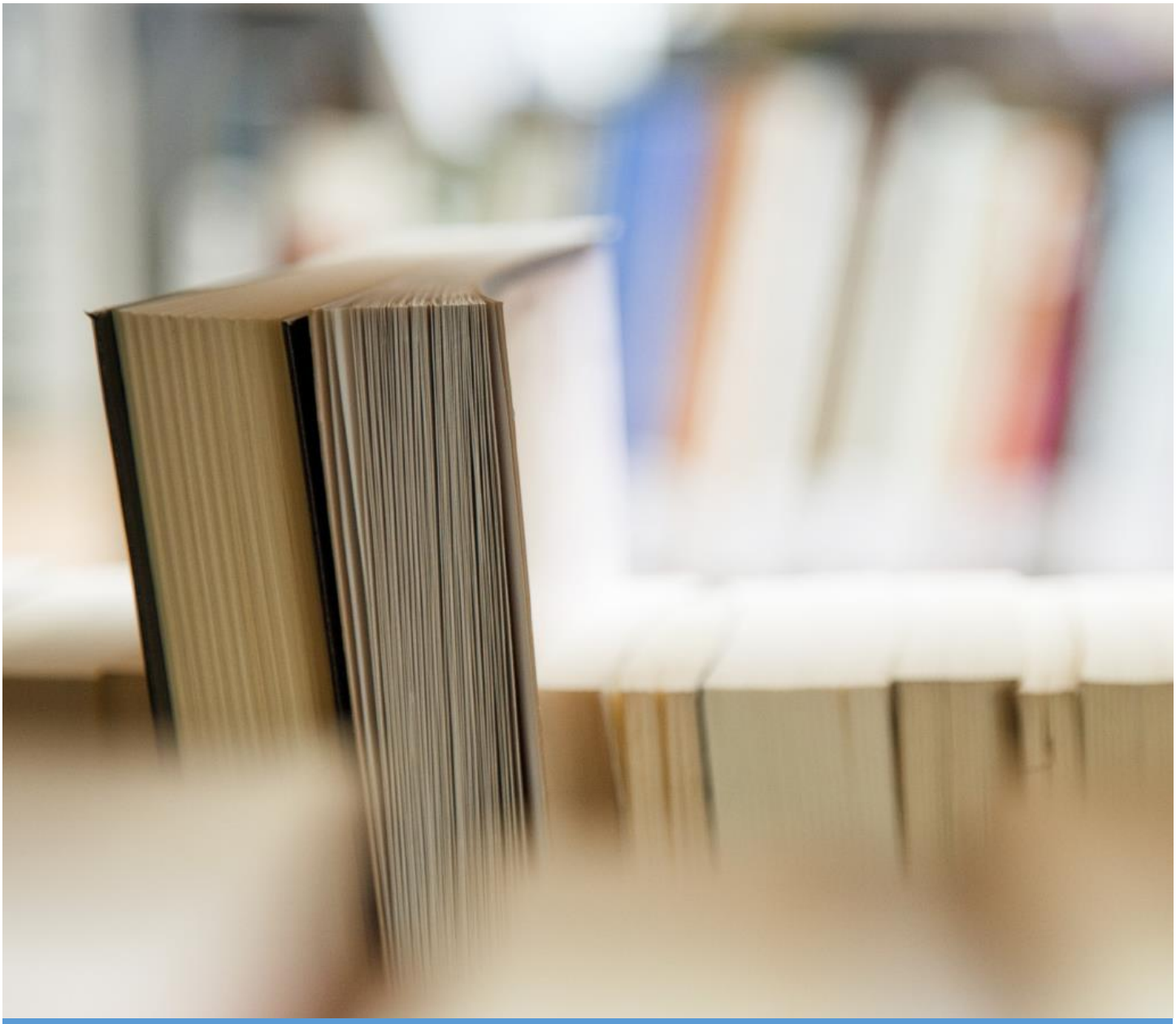


DATA-MINING BUSINESS REPORT

PGP-DSBA

Srikanthpr.27@gmail.com

10/25/20

Data-Mining Project

Table of Contents

1 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.

Dataset: bank_marketing_part1_Data.csv

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. Describe the data briefly.

Solution:

- Load the required packages and read the dataset.
- Dataset contains total 210 Observations and 7 variables in the dataset. All the vari ables are of continuous numerical features.

	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

Figure-1: Data-Frame head with 5 rows

As mentioned in the data dictionary, all the variables are in different units and hence before we start with any algorithm, we need to ensure that all the features are in the accepted unit level. Let's look at the first 5 records again to be sure that all the features have the accepted unit now.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	19940.0	1692.0	0.8752	6675.0	37630.0	325.2	6550.0
1	15990.0	1489.0	0.9064	5363.0	35820.0	333.6	5144.0
2	18950.0	1642.0	0.8829	6248.0	37550.0	336.8	6148.0
3	10830.0	1296.0	0.8099	5278.0	26410.0	518.2	5185.0
4	17990.0	1586.0	0.8992	5890.0	36940.0	206.8	5837.0

Figure-2: Data-Frame head with 5 rows after unit change

Dataset does not have any null value. Also, there are no duplicate records in the dataset.

```
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)
```

Figure-3: Data Structure of the data Features

All the variable types are float64 continuous data type. Distribution plot shows that 'probability_of_full_payment' is negatively skewed and 'max_spent_in_single_shopping' is positively skewed.

The same is verified from the skew output as well, as detailed below:

Univariate Analysis:

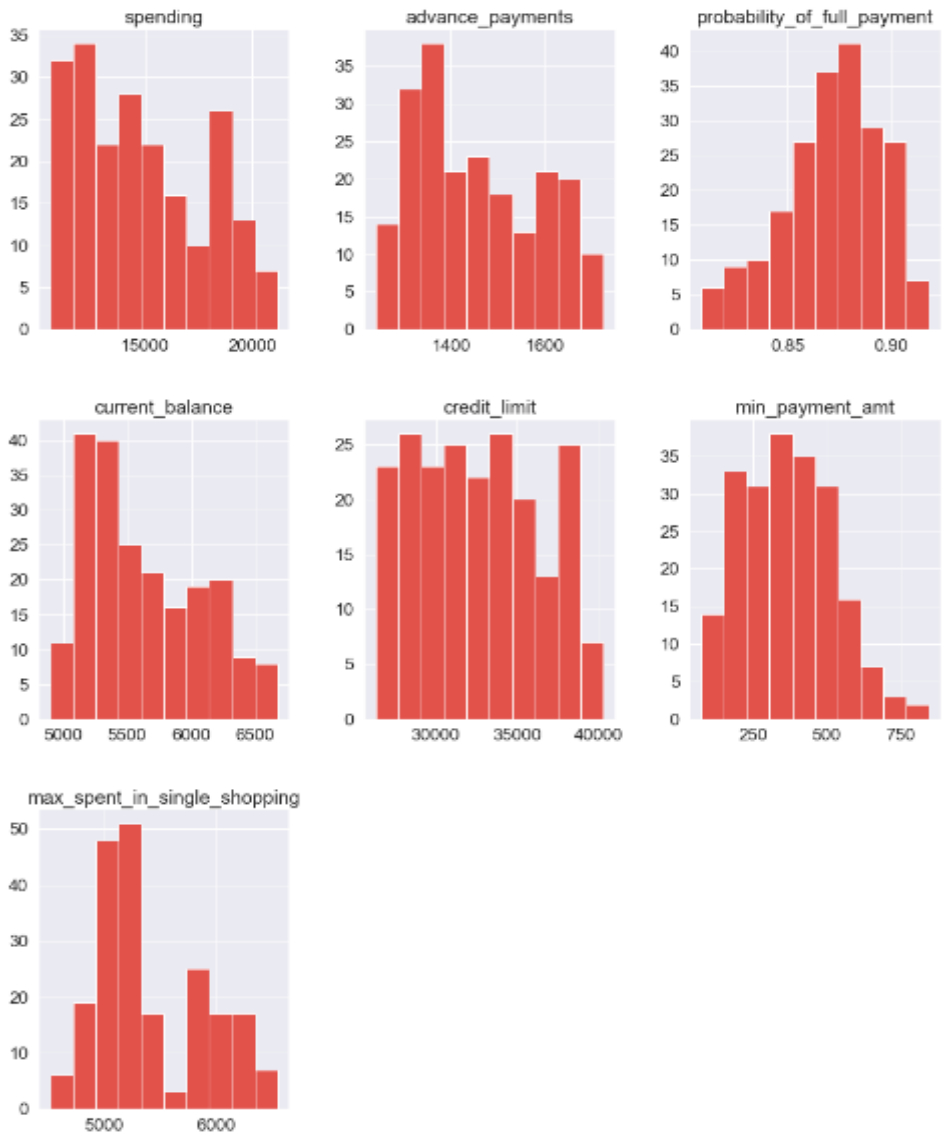


Figure-4: All variables represented using histogram

```
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
dtype: float64
```

Figure-5: Skewness

Pair plot among the following variables shows a strong relationship

'spending',advance_payments','current_balance','max_spent_in_single_shopping'

Multivariate Analysis:

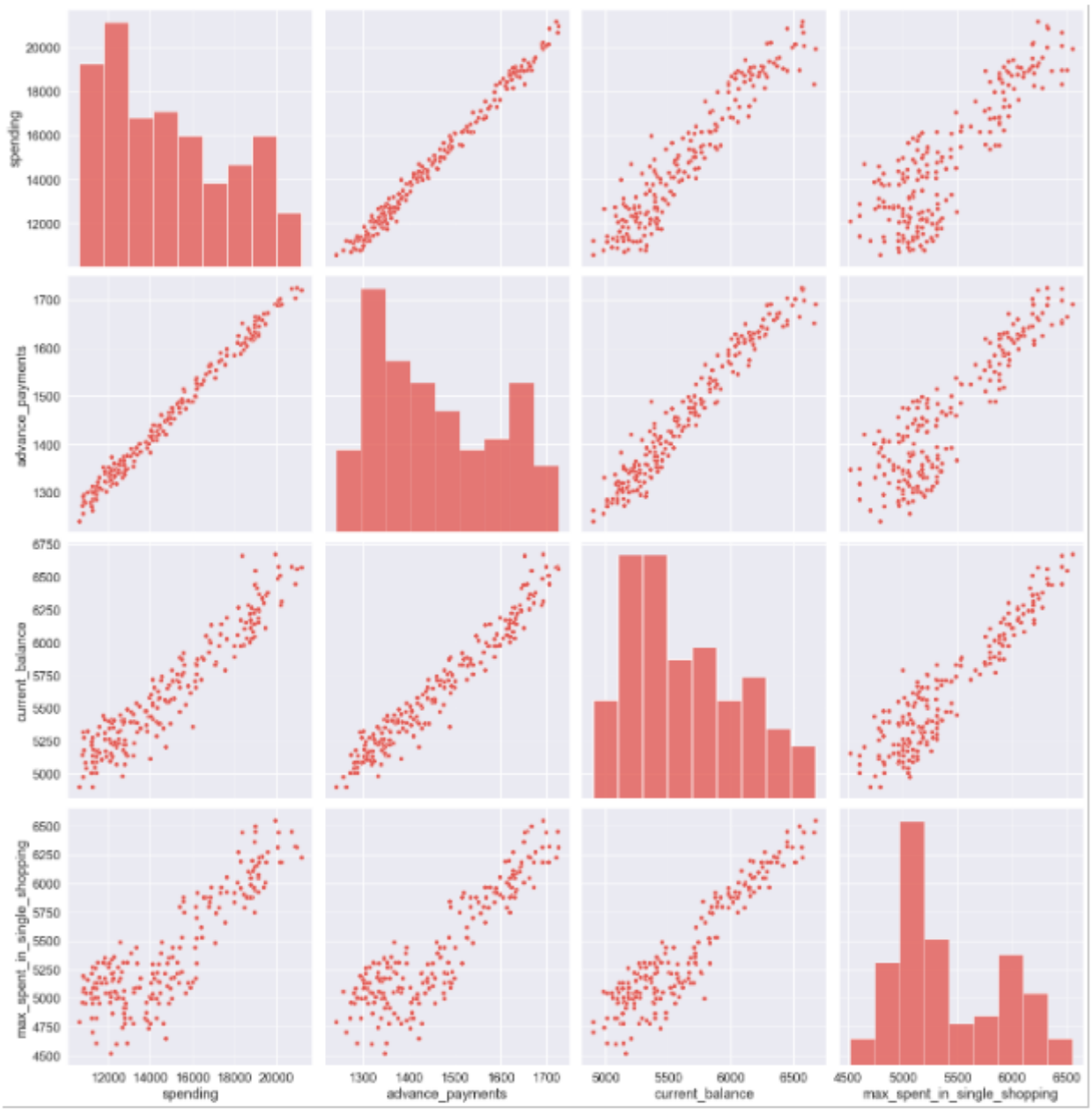


Figure-6: Pair plot of all variables

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

Solution:

Yes, scaling is required in this data set as all features have different weights and to ensure that none of the feature is identified as important only because of the weight, scaling is mandatory for this data set.

StandardScaler() function is used to scale the dataset. The scaled dataset is converted into data frame summary statistics are looked at to confirm if the data set has been scaled.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	2.100000e+02	2.100000e+02	2.100000e+02	2.100000e+02	2.100000e+02	2.100000e+02	2.100000e+02
mean	1.340198e-16	7.602384e-16	1.260896e-15	-9.886272e-16	1.799486e-16	8.617445e-17	2.955308e-16
std	1.002389e+00	1.002389e+00	1.002389e+00	1.002389e+00	1.002389e+00	1.002389e+00	1.002389e+00
min	-1.466714e+00	-1.649686e+00	-2.668236e+00	-1.650501e+00	-1.668209e+00	-1.956769e+00	-1.813288e+00
25%	-8.879552e-01	-8.514330e-01	-5.980791e-01	-8.286816e-01	-8.349072e-01	-7.591477e-01	-7.404953e-01
50%	-1.696741e-01	-1.836639e-01	1.039927e-01	-2.376280e-01	-5.733534e-02	-6.746852e-02	-3.774588e-01
75%	8.465989e-01	8.870693e-01	7.116771e-01	7.945947e-01	8.044956e-01	7.123789e-01	9.563941e-01
max	2.181534e+00	2.065260e+00	2.006586e+00	2.367533e+00	2.055112e+00	3.170590e+00	2.328998e+00

Figure-7: Summary statistics of scaled data

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

Solution:

Hierarchical clustering is applied on the scaled dataset. Clusters are created between the variables 'spending' and 'current balance' with a consideration that based on the spending and balance available, business can identify the customer profile from segmentation perspective.

Truncated dendrogram shows that best segmentation can be derived from 2 clusters.

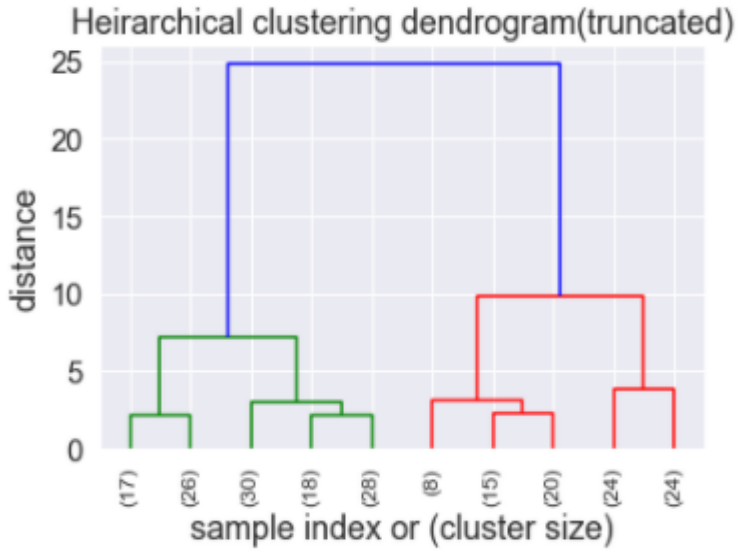


Figure-8: Truncated dendrogram

However as normally to understand the dataset, normally 2 clusters are not preferred because in most of the cases, business is already aware about the 2 classes in the dataset and hence to generate some more insights, segmentation with more than 2 clusters is preferred.

As an example, for a bank dataset, bank would like to know more than 'good' and 'not so good' customers and hence more insight we are able to generate with more than 2 clusters, better it is for business.

Hence let's consider 3 clusters and plot the clusters to confirm if the desired clusters are providing the required segmentation details.

First "f-cluster" technique was used to create the clusters. However, as we can see from the given below plot, segmentation was not very clear.

Figure-9: Scatterplot between Current Balance and Spending with cluster labels as hue using 'fclust' from SCIPY

Hence Agglomerative technique was used for the given dataset. Following is a plot with a clear segmentation of the dataset for the selected features.

Figure-10: Scatterplot between Current Balance and Spending with cluster labels as hue using 'Agglomerative from Sklearn

Also, clusters were tried with all the features, however spending and customer balance is giving the good customer segmentation.

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

Solution:

K-means clustering technique was used along with elbow curve to define the optimum clusters for this data set. Once again Spending and current Balance features were used to build the cluster. As per the Elbow method, 3 cluster were identified as an optimum number.

Elbow Plot:

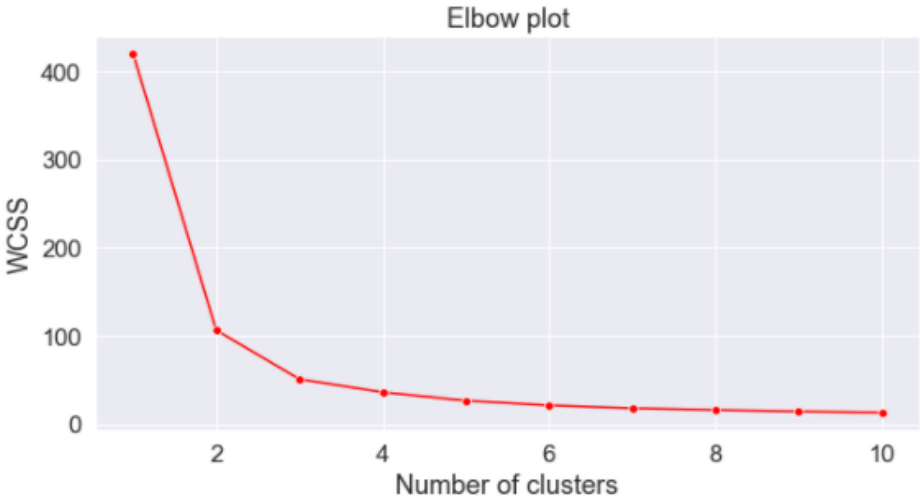


Figure-11: Elbow Plot

As per the silhouette score, optimal number of clusters are 2. However as mentioned earlier, 2 is not preferred way of profiling the dataset and hence the ideal number of clusters to be considered is 3. With a silhouette score of 0.5588.

Plot of the cluster with centroid is as follows:

Figure-12: Scatterplot between Current Balance and Spending with cluster labels as hue and cluster centroids

As hierarchical clustering has provided a better segmentation, the output obtained from the same would be used to define the customer strategies.

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

Solution:

In today's competitive world, it is crucial to understand customer behaviour and categorize customers based on their demography and buying behaviour. This is a critical aspect of customer segmentation that allows marketers to better tailor their marketing efforts to various audience subsets in terms of promotional marketing and product development strategies.

Based on the clusters obtained from hierarchical clustering strategy for different segment of customers is as follows:

Figure-13: Scatterplot between Current Balance and Spending with cluster labels as hue using 'Agglomerative from Sklearn

i. Type 1 Customers (Represented by purple Dots in the hierarchical cluster plot) Segment wherein spending is not that high, but there are good number of customers with a high current balance. So, for these customers, reasons need to be identified why the available balance is not being utilised. To start with survey could be conducted to understand if suitable offers/options are not available on the ecommerce sites wherein customer does purchase or have they faced any specific challenge while using the card because of which spending on the card are low etc.

ii. Type 2 Customers (Represented by purple Dots in the hierarchical cluster plot wherein spending is high and current balance is low) – Bank can analyse this segment to understand if these customers could be offered a different product and/ or if their existing credit limit could be increased based on their profile details as these customers might spend more if they have sufficient balance.

iii. Type 3 Customers (Represented by Red Dots in the hierarchical cluster plot wherein both spending and current balance are high). This is probably a high value customer and hence special discounted pricing based promotional campaigns for this group to increase their spending and use current balance.

iv. Type 4 Customers (Represented by Green Dots in the hierarchical cluster plot wherein both spending and current balance are moderate). Further analysis could be performed to understand what would make this segment to move into Red area.

2 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.

Dataset: insurance_part2_data-1.csv

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)
7. Destination of the tour (Destination)
8. Amount of sales of tour insurance policies (Sales)
9. The commission received for tour insurance firm (Commission)
10. Age of insured (Age)

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

Solution:

- Load the required packages and read the dataset.
- Dataset contains 3000 Observations and 10 variables including the target variable.
- Descriptive statistics of all the features is as follows:

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
count	3000.000000	3000	3000	3000	3000.000000	3000	3000.000000	3000.000000	3000	3000
unique	NaN	4	2	2	NaN	2	NaN	NaN	5	3
top	NaN	EPX	Travel Agency	No	NaN	Online	NaN	NaN	Customised Plan	ASIA
freq	NaN	1365	1837	2076	NaN	2954	NaN	NaN	1136	2465
mean	38.091000	NaN	NaN	NaN	14.529203	NaN	70.001333	60.249913	NaN	NaN
std	10.463518	NaN	NaN	NaN	25.481455	NaN	134.053313	70.733954	NaN	NaN
min	8.000000	NaN	NaN	NaN	0.000000	NaN	-1.000000	0.000000	NaN	NaN
25%	32.000000	NaN	NaN	NaN	0.000000	NaN	11.000000	20.000000	NaN	NaN
50%	36.000000	NaN	NaN	NaN	4.630000	NaN	26.500000	33.000000	NaN	NaN
75%	42.000000	NaN	NaN	NaN	17.235000	NaN	63.000000	69.000000	NaN	NaN
max	84.000000	NaN	NaN	NaN	210.210000	NaN	4580.000000	539.000000	NaN	NaN

Figure-13: Summary statistics of the data

Key takeaways from the descriptive statistics:

- Dataset pertains to 4 different agency firms, there are 2 types of insurance firm's i.e Airline and Travel agency, 5 different products for 3 destinations.

- 4 continuous variables i.e Age, commission, Duration and Sales out of which age is not skewed.
- Duration has a high dispersion.

Dataset does not have any null values.

Duration has lot of outliers. Two records with value as "-1" and "4580" seems to be incorrect and hence the same was dropped from the dataset.

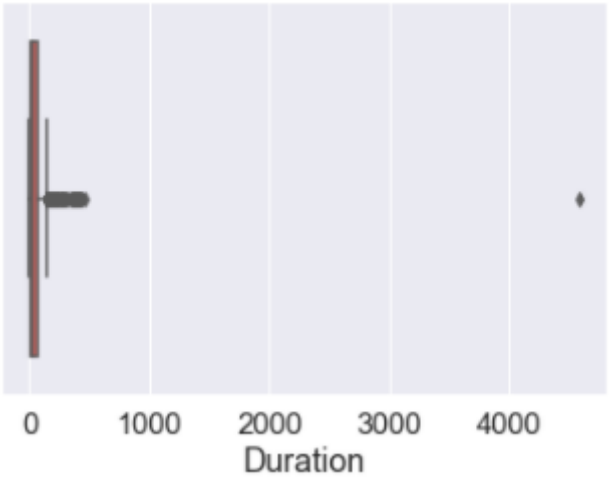


Figure-15: Boxplot for variable 'Duration'

Duration has a correlation with the commission and data points are more concentrated towards lower commission and lower duration.

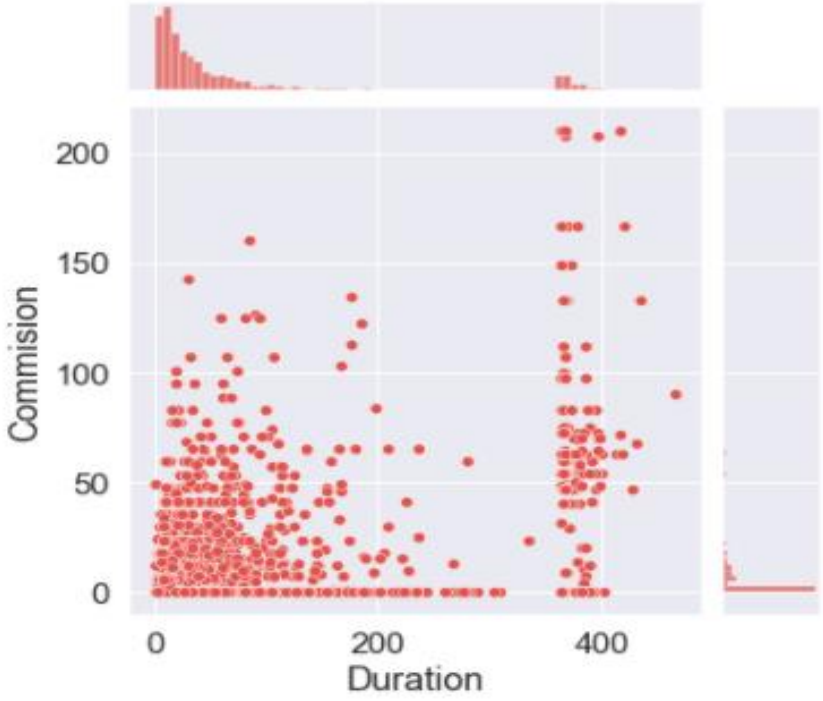


Figure-16: Joint plot between Commission and Duration

As expected, duration and sales also have a high correlation.

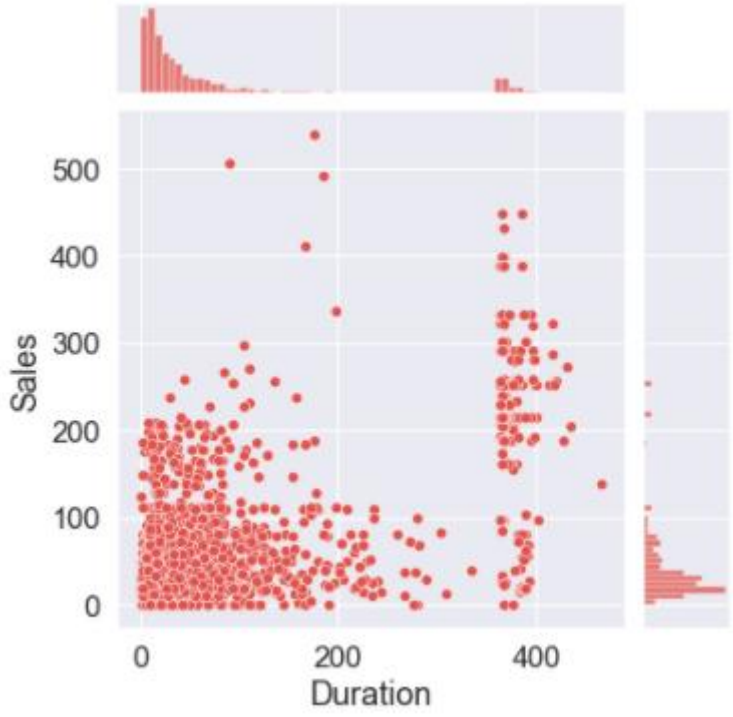


Figure-17: Joint plot between Sales and Duration

Sales and commission have a high correlation and linear relationship.

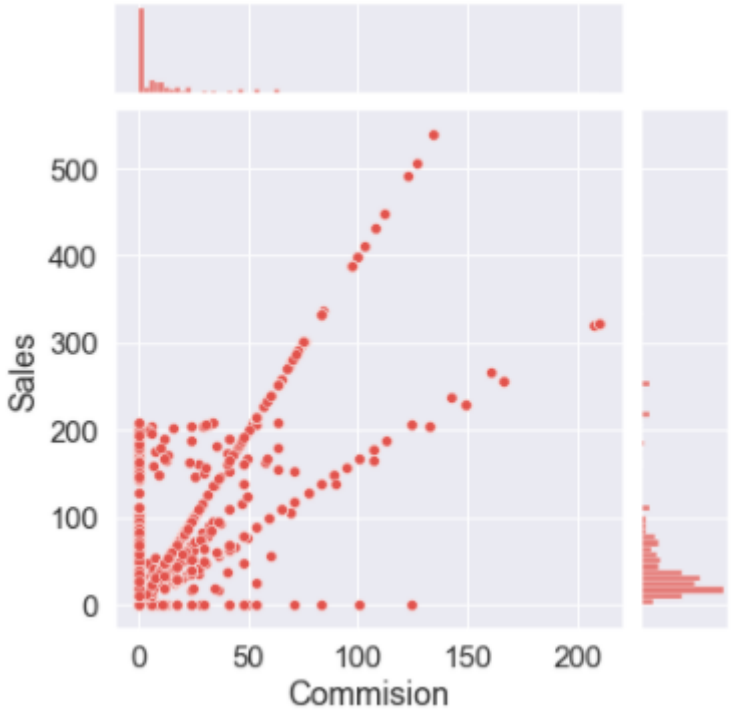


Figure-18: Joint plot between Sales and Commission

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

Solution:

Before doing the data split variables types were checked. As there are many columns with the type as object, these variables were converted into categorical type.

Then target variable was captured into separate vector for training and test data set.

Then the dataset was split into train and test in the ratio of 70:30.

Building a Decision Tree Classifier

- Decision tree classifier model built using below hyper parameters.

Hyper-parameter selected:

1. 'criterion': 'gini'
2. 'max_depth': 20
3. 'min_samples_leaf': 75
4. 'min_samples_split': 450

- Best grid estimators are stored into variable best_grid_dtcl
- Agency_Code is the most important variable for predicting claiming insurance.

	Imp
Agency_Code	0.667608
Sales	0.179399
Product Name	0.096666
Duration	0.045032
Commision	0.011294
Age	0.000000
Type	0.000000
Channel	0.000000
Destination	0.000000

- Prediction of train and test result are stored into ytrain_predict_dtcl and ytest_predict_dtcl respectively.

Head of the Predicted Classes and Probs

	0	1
0	0.435146	0.564854
1	0.240000	0.760000
2	0.911357	0.088643
3	0.845029	0.154971
4	0.640271	0.359729

Building a Random Forest Classifier

- Random forest classifier is built using below hyper-parameters:

Hyper-parameter selected:

1. 'max_depth': 20

2. 'max_features': 4
3. 'min_samples_leaf': 10
4. 'min_samples_split': 100
5. 'n_estimators': 300

- Best grid estimators are stored into variable best_grid_rfcl.
- Agency_Code is the most important variable for predicting claiming insurance.

	Imp
Agency_Code	0.309193
Product Name	0.221061
Sales	0.161085
Commision	0.123689
Duration	0.083913
Type	0.049422
Age	0.037402
Destination	0.013431
Channel	0.000804

- Prediction of train and test result are stored into ytrain_predict_rfcl and ytest_predict_rfcl respectively.

Head of the Predicted Classes and Probs

	0	1
0	0.407959	0.592041
1	0.219083	0.780917
2	0.890896	0.109104
3	0.971716	0.028284
4	0.584876	0.415124

Building a Neural Network Classifier

- Scaling is necessary before building a neural network classifier.
- After scaling independent train and test data the resulting output is stored on to Scaled_X_train and Scaled_X_test.
- Neural-network classifier is built using below hyper-parameters:

Hyper-parameter selected:

1. 'hidden_layer_sizes': 50
2. 'max_iter': 2500
3. 'solver': 'sgd'
4. 'tol': 0.01

- Best grid estimators are stored into variable best_grid_mlp.
- Prediction of train and test result are stored into ytrain_predict_mlp and ytest_predict_mlp respectively.

Head of the Predicted Classes and Probs

	0	1
0	0.547579	0.452421
1	0.239912	0.760088
2	0.811701	0.188299
3	0.810405	0.189595
4	0.497912	0.502088

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.

Solution:

Model performance measures for a Decision Tree Classifier

Training Dataset

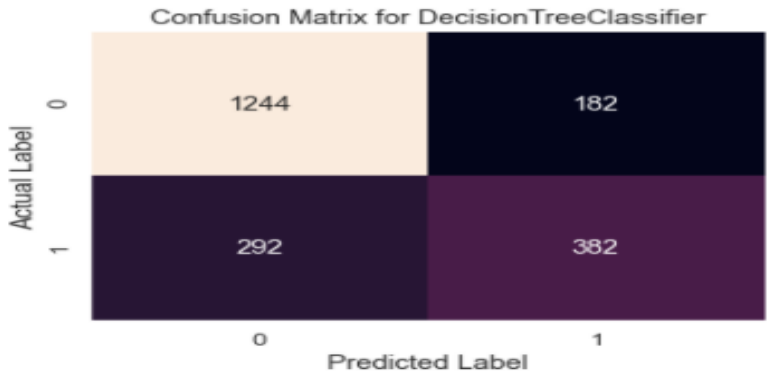
```
Accuracy for DecisionTreeClassifier model on Training set is
 0.774


Classification report for DecisionTreeClassifier model on Training set is
              precision    recall  f1-score   support

           0       0.81      0.87      0.84      1426
           1       0.68      0.57      0.62       674

    accuracy                           0.77      2100
   macro avg       0.74      0.72      0.73      2100
weighted avg       0.77      0.77      0.77      2100
```

Confusion Matrix for DecisionTreeClassifier model on Training set is




```
cart_train_precision  0.68
cart_train_recall  0.57
cart_train_f1  0.62
```

AUC and ROC for the training data

- Area under the cure [AUC] : 0.809 or 80.9%

Testing Dataset:

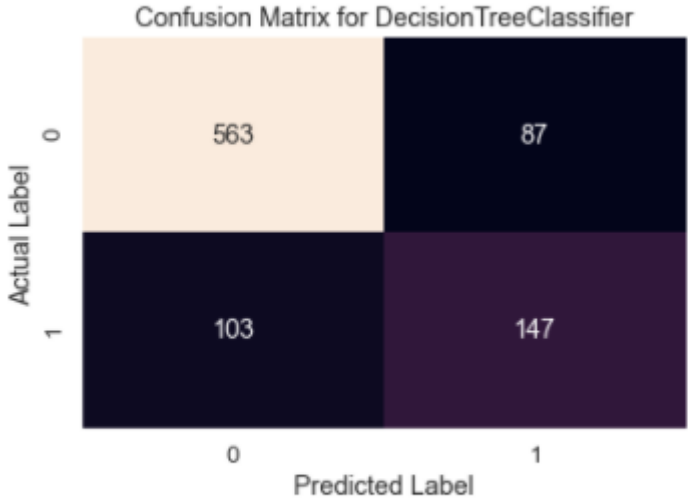
```
Accuracy for DecisionTreeClassifier model on Testing set is
 0.789
```

```
Classification report for DecisionTreeClassifier model on Testing set is
              precision    recall  f1-score   support

           0       0.85      0.87      0.86       650
           1       0.63      0.59      0.61       250

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

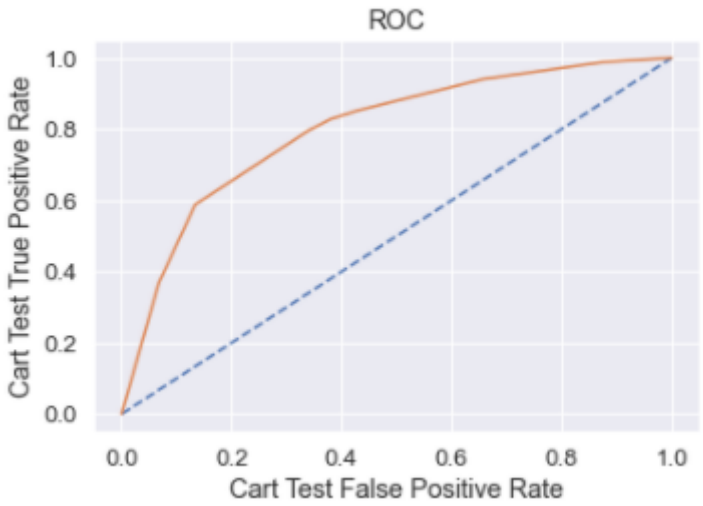
```
Confusion Matrix for DecisionTreeClassifier model on Testing set is
```




```
cart_test_precision  0.63
cart_test_recall  0.59
cart_test_f1  0.61
```

AUC and ROC for the testing data

- Area under the cure [AUC] : 0.799 or 79.9%



Cart Conclusion:

→ **Train Data:**

1. AUC: 80.9%
2. Accuracy: 77.4%
3. Precision: 68%
4. Recall: 57%
5. f1-Score: 62%

→ **Test Data:**

1. AUC: 79.9%
2. Accuracy: 78.9%
3. Precision: 63%
4. Recall: 59%
5. f1-Score: 61%

- Training and Test set results are almost similar, and with the overall measures high, the model is a good model.
- Agency_code is the most important variable for predicting insurance claimed.

Model performance measures for a Random Forest Classifier

Training Dataset

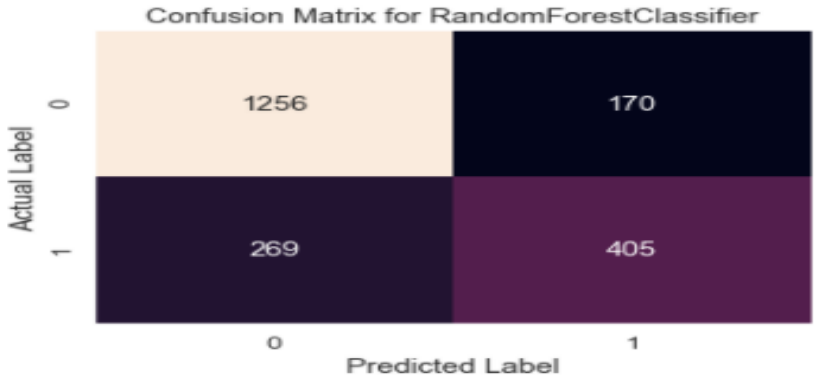
```
Accuracy for RandomForestClassifier model on Training set is
 0.791


Classification report for RandomForestClassifier model on Training set is
              precision    recall  f1-score   support

           0       0.82      0.88      0.85      1426
           1       0.70      0.60      0.65       674

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

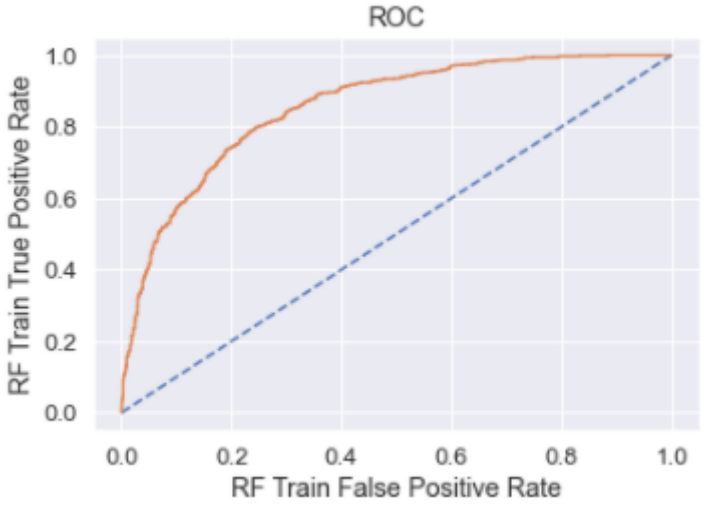
```
Confusion Matrix for RandomForestClassifier model on Training set is
```




```
rfcl_train_precision  0.7
rfcl_train_recall  0.6
rfcl_train_f1  0.65
```

AUC and ROC for the training data

- Area under the cure [AUC] : 0.855 or 85.5%



Testing Dataset:

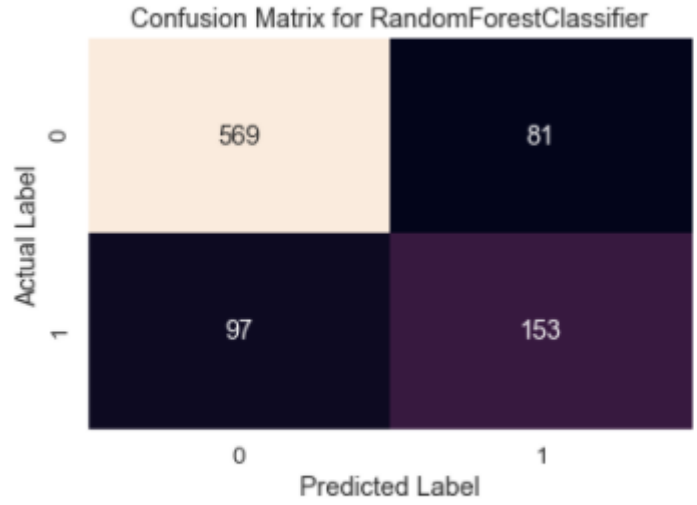
```
Accuracy for RandomForestClassifier model on Testing set is
 0.802
```

```
Classification report for RandomForestClassifier model on Testing set is
              precision    recall  f1-score   support

           0       0.85      0.88      0.86       650
           1       0.65      0.61      0.63       250

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

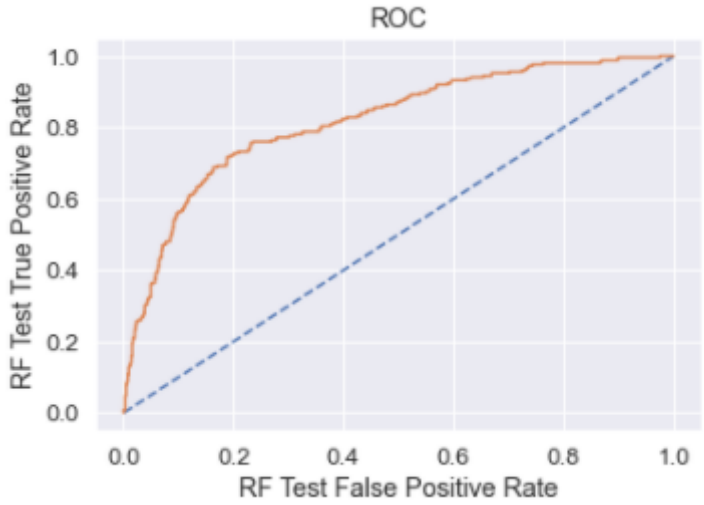
```
Confusion Matrix for RandomForestClassifier model on Testing set is
```




```
rfcl_test_precision  0.65
rfcl_test_recall  0.61
rfcl_test_f1  0.63
```

AUC and ROC for the testing data

- Area under the curve [AUC] : 0.818 or 81.8%



RF Conclusion:

→ **Train Data:**

1. AUC: 85.5%
2. Accuracy: 79.1%
3. Precision: 70%
4. Recall: 60%
5. f1-Score: 65%

→ **Test Data:**

1. AUC: 81.8%

2. Accuracy: 80.2%
3. Precision: 65%
4. Recall: 61%
5. f1-Score: 63%

- Training and Test set results are almost similar, and with the overall measures high, the model is a good model.
- Agency_code is the most important variable for predicting insurance claimed.

Model performance measures for a Neural Network Classifier

Training Dataset

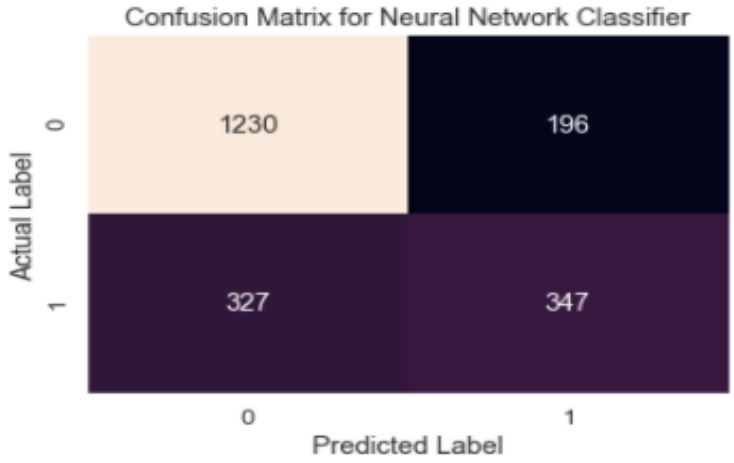
```
Accuracy for Neural Network Classifier model on Training set is
 0.751


Classification report for Neural Network Classifier model on Training set is
              precision    recall  f1-score   support

           0       0.79      0.86      0.82      1426
           1       0.64      0.51      0.57       674

    accuracy                           0.75      2100
   macro avg       0.71      0.69      0.70      2100
weighted avg       0.74      0.75      0.74      2100
```

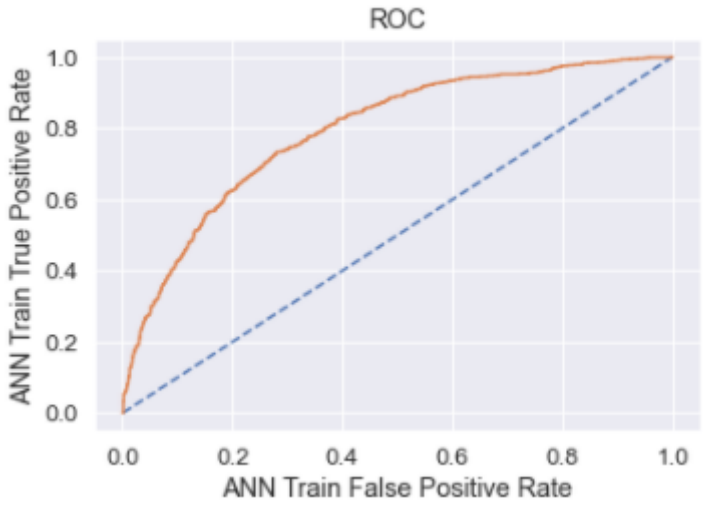
```
Confusion Matrix for Neural Network Classifier model on Training set is
```




```
ANN_train_precision  0.64
ANN_train_recall  0.51
ANN_train_f1  0.57
```

AUC and ROC for the training data

- Area under the cure [AUC] : 0.793 or 79.3%



Testing Dataset:

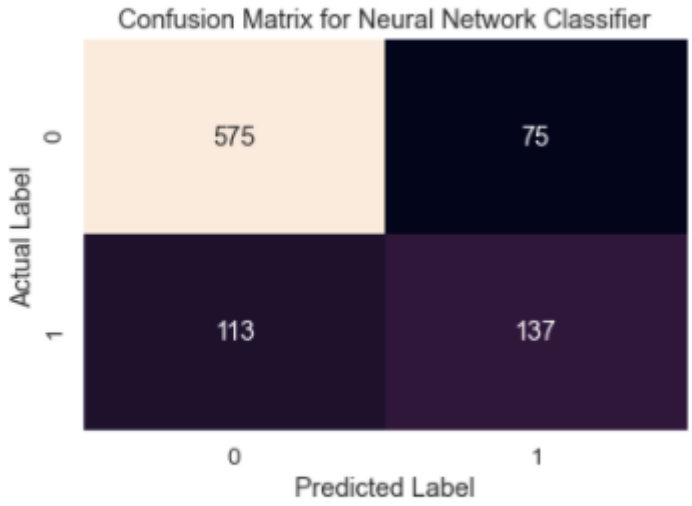
```
Accuracy for Neural Network Classifier model on Testing set is
 0.791
```

```
Classification report for Neural Network Classifier model on Testing set is
              precision    recall  f1-score   support

           0       0.84      0.88      0.86       650
           1       0.65      0.55      0.59       250

    accuracy                           0.79       900
   macro avg       0.74      0.72      0.73       900
weighted avg       0.78      0.79      0.79       900
```

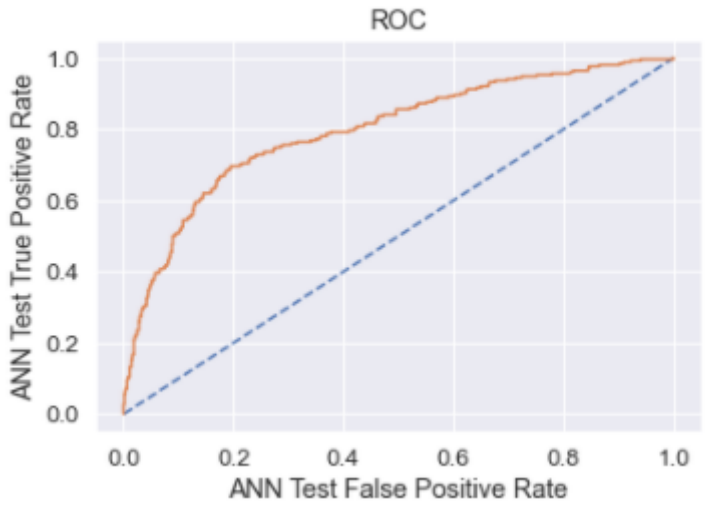
```
Confusion Matrix for Neural Network Classifier model on Testing set is
```




```
ANN_test_precision  0.65
ANN_test_recall  0.55
ANN_test_f1  0.59
```

AUC and ROC for the testing data

- Area under the curve [AUC] : 0.796 or 79.6%



Neural Networks Conclusion:

→ **Train Data:**

1. AUC: 79.3%
2. Accuracy: 75.1%
3. Precision: 64%
4. Recall: 51%
5. f1-Score: 57%

→ **Test Data:**

1. AUC: 79.6%

2. Accuracy: 79.1%
3. Precision: 65%
4. Recall: 55%
5. f1-Score: 59%

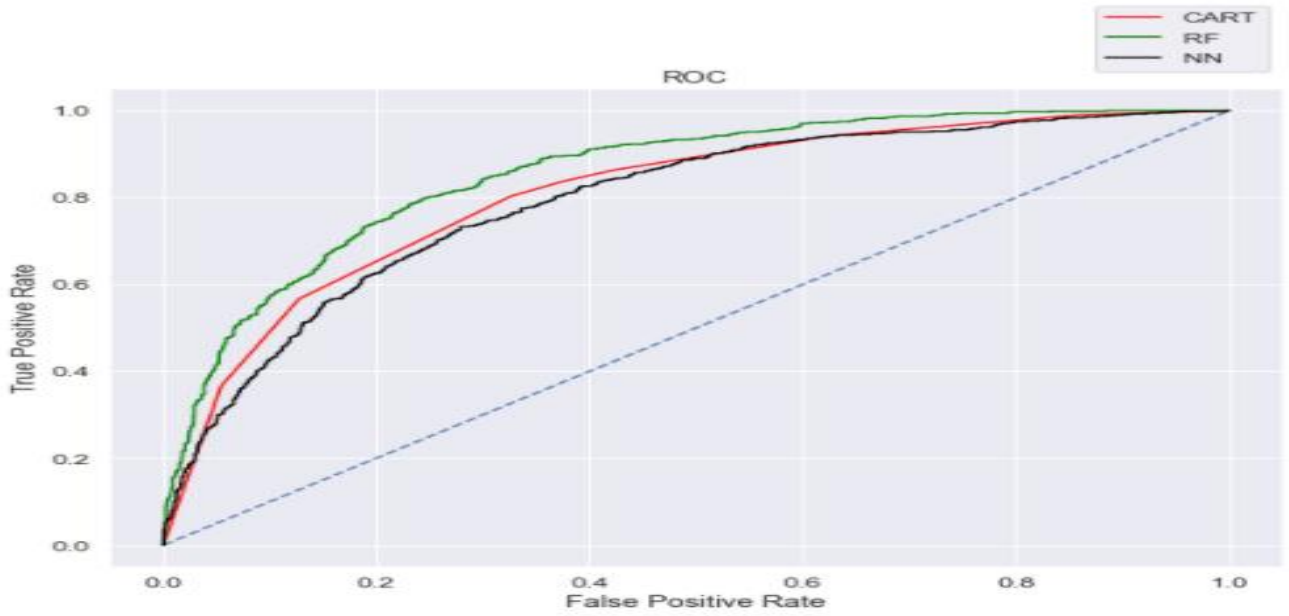
- Training and Test set results are almost similar, and with the overall measures high, the model is a good model.

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

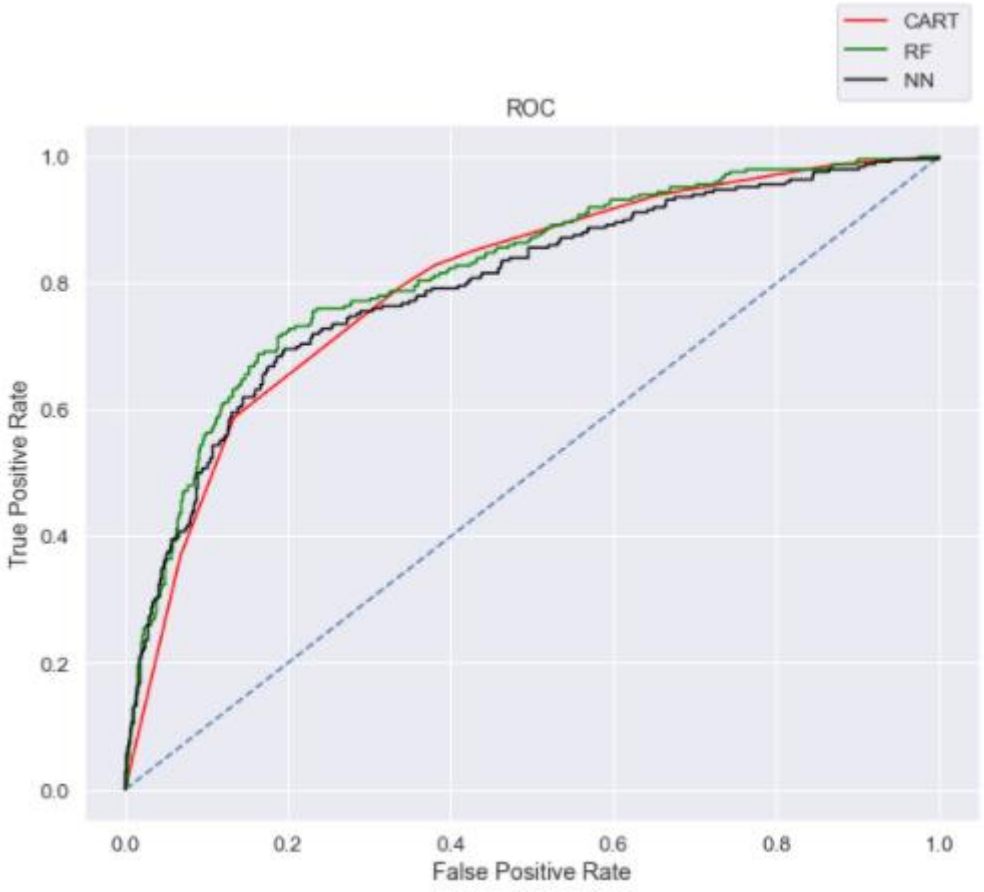
Solution:

	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.77	0.79	0.79	0.80	0.75	0.79
AUC	0.81	0.80	0.85	0.82	0.79	0.80
Recall	0.57	0.59	0.60	0.61	0.51	0.55
Precision	0.68	0.63	0.70	0.65	0.64	0.65
F1 Score	0.62	0.61	0.65	0.63	0.57	0.59

ROC Curve for the 3 models on the Training data



ROC Curve for the 3 models on the Testing data



- Out of the 3 models, Random Forest has slightly better performance than the Cart and Neural network model.
- Overall all the 3 models are reasonably stable enough to be used for making any future predictions.
- From Cart and Random Forest Model, the variable Agency_code is found to be the most useful feature amongst all other features for predicting if a person has claimed or not. If Claimed is yes, then those customers have more chances of getting tour insurance.

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

Solution:

- From the above model's we know that 'Agency_code' feature plays an important role for predictions to be made on the target variable Claimed.
- We know that the best model accuracy received was 80%. We need to increase the samples of proportions of 1's claim frequency in order to increase the model accuracy towards 90% or to a more accurate result.
- With the above 3 models we can certainly decrease the higher claim frequency or the customer who incorrectly claimed tour insurance can be rejected .By doing this we can certainly save enormous amount overflow of cash money for an agency company.