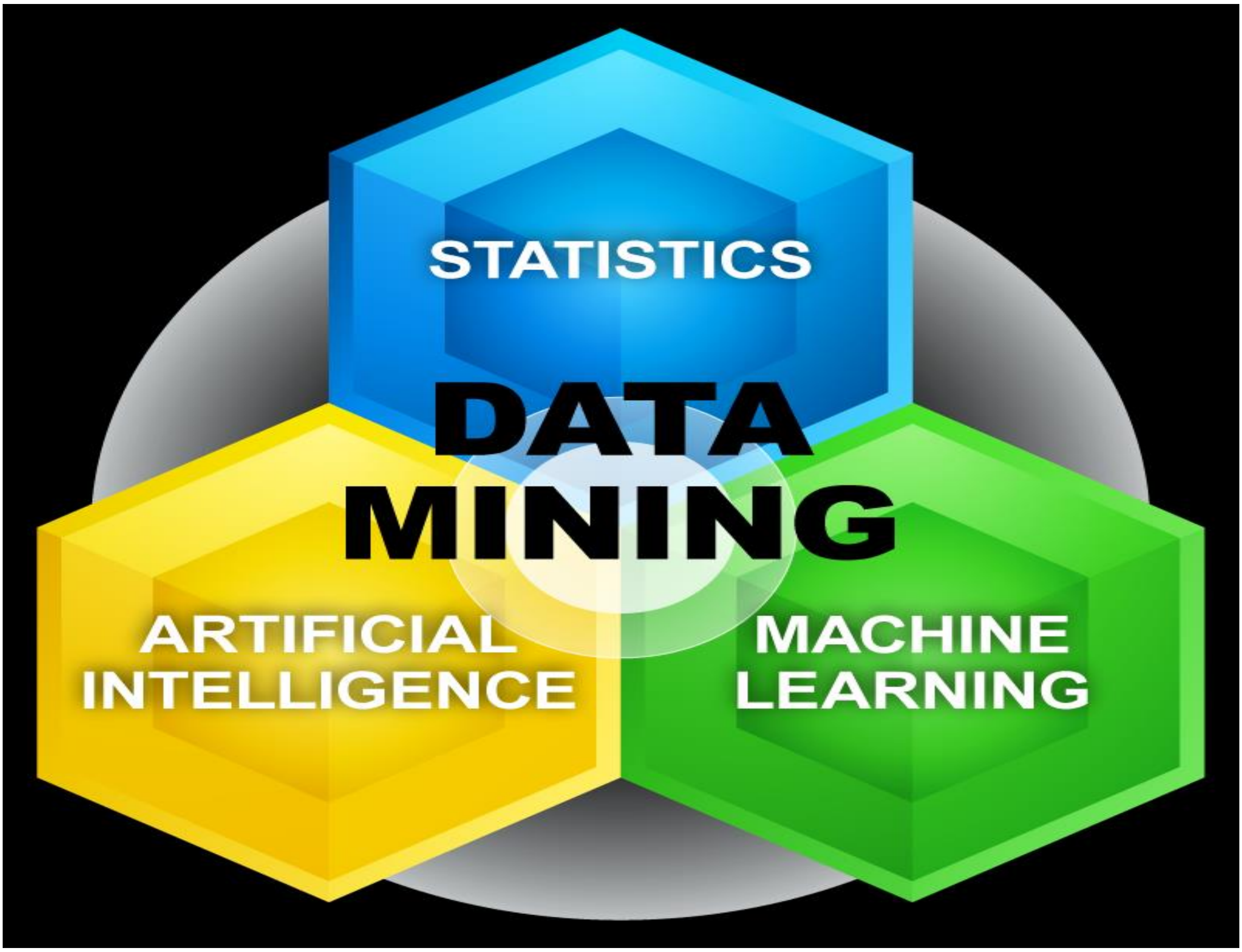


DATA MINING PROJECT

M.K. SUGANTHE RAMYA

CASE STUDY 1:

CUSTOMER SEGMENTATION USING CLUSTERING METHODS

CUSTOMER SEGMENTATION USING CLUSTERING METHODS BASED ON CREDITCARD USAGE

Table of contents

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)

Project Objective:

The Objective of the report is to explore the ": bank_marketing_part1_Data.csv" dataset in Python (JUPYTER NOTEBOOK) and generate insights about the dataset. This exploration report will consist of the following:

- Importing the dataset in jupyter notebook.
- Understanding the structure of dataset.
- Exploratory Data analysis
- Graphical exploration
- Clustering techniques
- Insights from the dataset

Assumptions:

Several business enterprises have come to realize the significance of Customer Relationship Management (CRM) and the application of technical expertise to achieve competitive advantage. This study explores the importance of Customer Segmentation as a core function of CRM as well as the various models for segmenting customers using clustering techniques. The available clustering models for customer segmentation are K-Means and Hierarchical Clustering.

Exploratory Data Analysis:

Bank marketing dataset

	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	6.55
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	5.89	3.694	2.068	5.837

Information on Fever dataset:

```
<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
```

Inference

- There are 210 rows and 7 columns
- There are no null values

Summary of the dataset:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	210	210	210	210	210	210	210
mean	14.84752	14.55929	0.870999	5.628533	3.258605	3.700201	5.408071
std	2.909699	1.305959	0.023629	0.443063	0.377714	1.503557	0.49148
min	10.59	12.41	0.8081	4.899	2.63	0.7651	4.519
25%	12.27	13.45	0.8569	5.26225	2.944	2.5615	5.045
50%	14.355	14.32	0.87345	5.5235	3.237	3.599	5.223
75%	17.305	15.715	0.887775	5.97975	3.56175	4.76875	5.877
max	21.18	17.25	0.9183	6.675	4.033	8.456	6.55

Inference

SPENDING:

- Minimum amounts spend - 10590
- Maximum amounts spend - 21180
- Average amounts spend – 14847

PROBABILITY OF FULL PAYMENT:

On average 87% of customers made a full payment.

ADVANCE_PAYMENTS:

Customers pay 1725 as maximum payment.

MAX SPENT IN SINGLE SHOPPING:

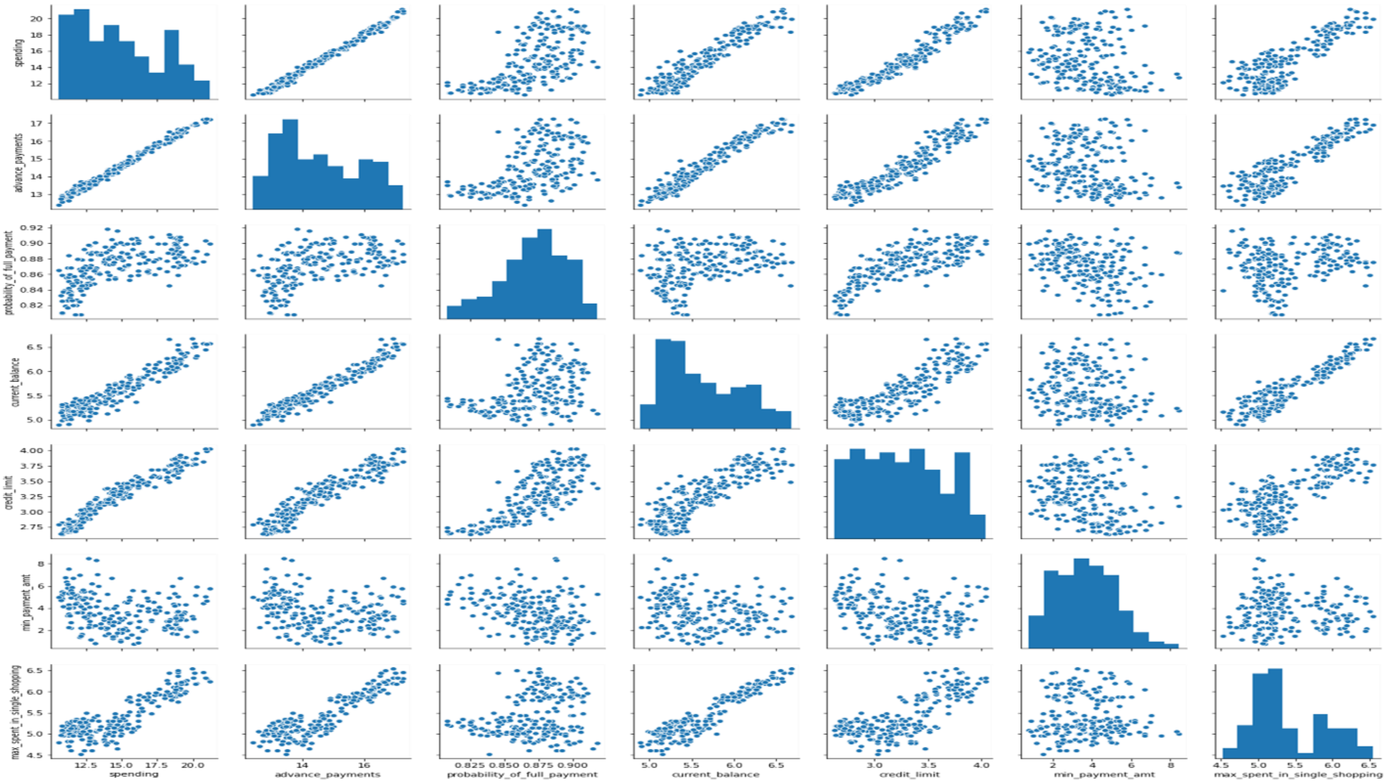
On average customers spend 5408 on single shopping.

MIN PAYMENT AMOUNT:

This variable may have outliers.

Bivariate Analysis:

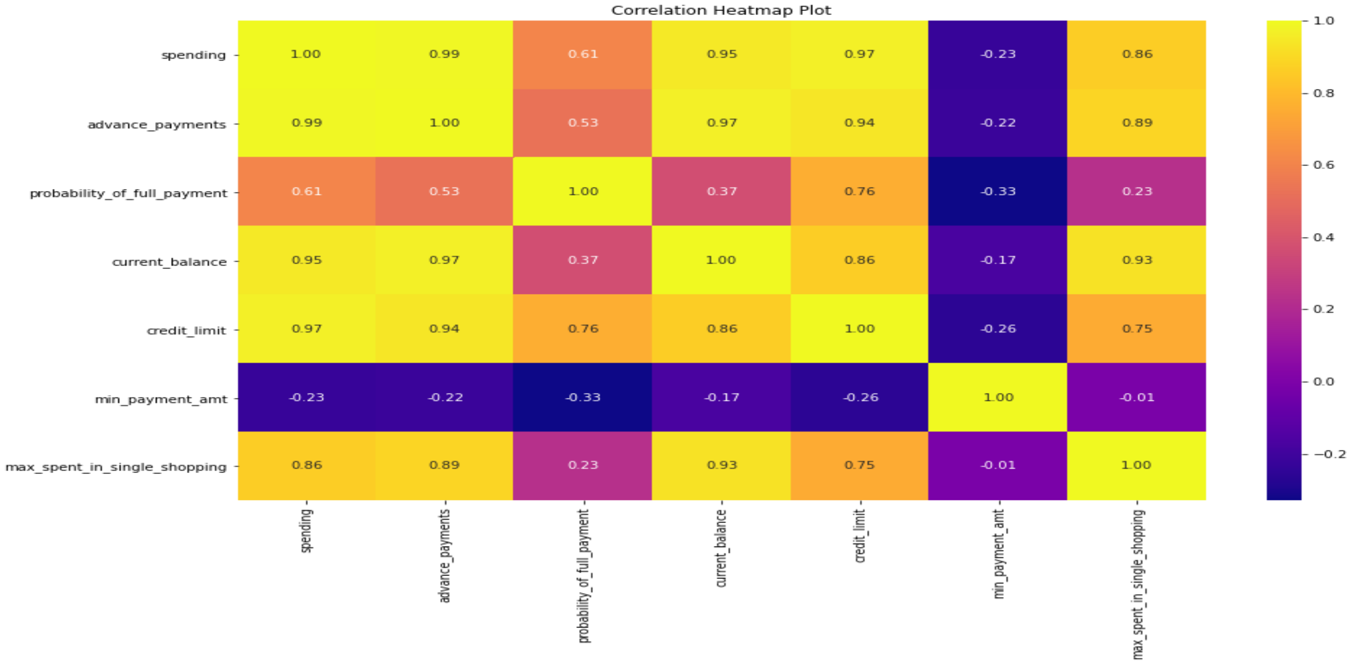
Pair plot between variables:



Inference:

There is multi-collinearity among the variables.

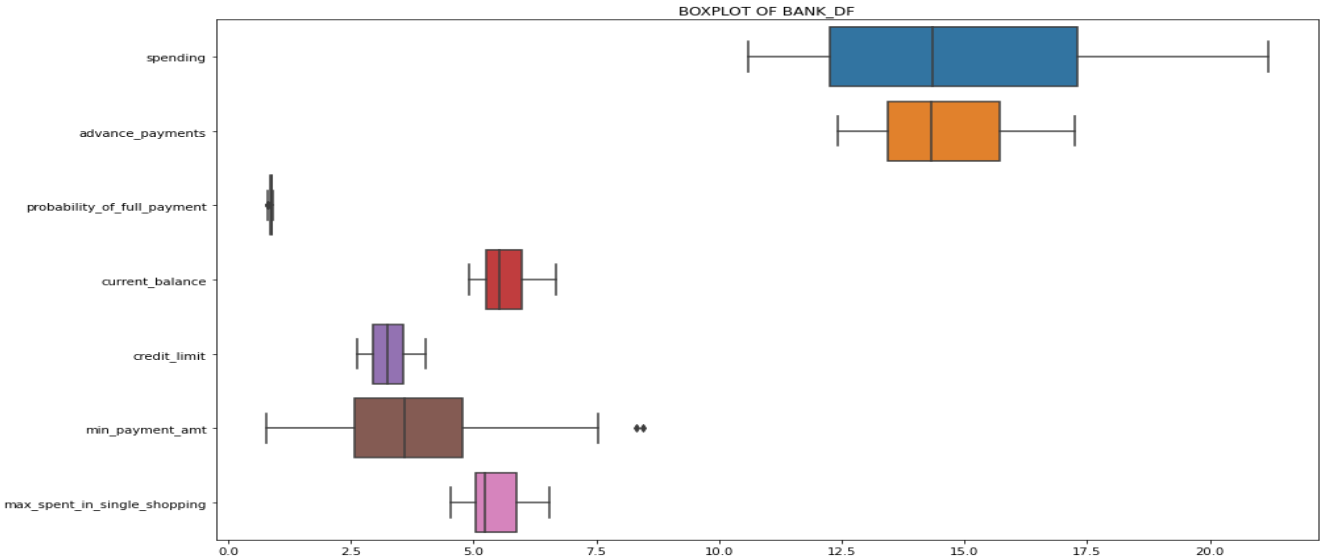
Correlation Heatmap:



Inference

- Spending and advance payments variables are highly correlated.
- Max spent in single shopping and min payment amount are slightly correlated

BOXPLOT OF BANK_DF:



Inference

Most of the variable has no outliers, only min_payment_amt have outlier.

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

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754355	1.811968	0.17823	2.367533	1.338579	-0.29881	2.328998
1	0.393582	0.25384	1.501773	-0.60074	0.858236	-0.24281	-0.53858
2	1.4133	1.428192	0.504874	1.401485	1.317348	-0.22147	1.509107
3	-1.38403	-1.22753	-2.59188	-0.79305	-1.63902	0.987884	-0.45496
4	1.082581	0.998364	1.19634	0.591544	1.155464	-1.08815	0.874813

Inference

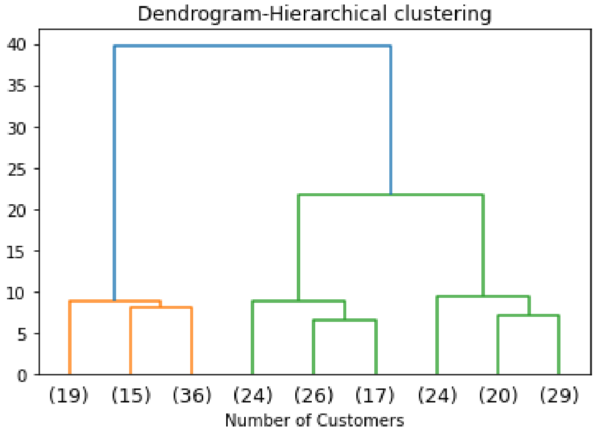
- To make sure that calculations will not be biased either to the extremely high or to the very low values. In other words, to make sure that all your data are at the same level.

- In this dataset, variables like Credit limit, spending, current balance have higher values whereas max payment amount have low values. There is probability of full payment .so most of the variables have different level of values.so, it is wise to scale the data.

- Scaling will make dataset are not biases and all data are at same level. StandardScaler used in this dataset. So, the values stay between 0 and 1

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

Hierarchical clustering:



Inference

- A dendrogram is a diagram that shows the hierarchical relationship between objects.
- The dendrogram above shows the hierarchical clustering of 9 observations
- The height of the dendrogram indicates the order in which the clusters were joined.
- The dendrogram shows us that the big difference between clusters is between the 2 clusters
- This dendrogram shows 2 clusters, but two clusters do not make much difference in the business impact. So, further process done with 3 clusters

Method 1: 'maxclust' criterion

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

Method 2: 'distance' criterion

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

Inference:

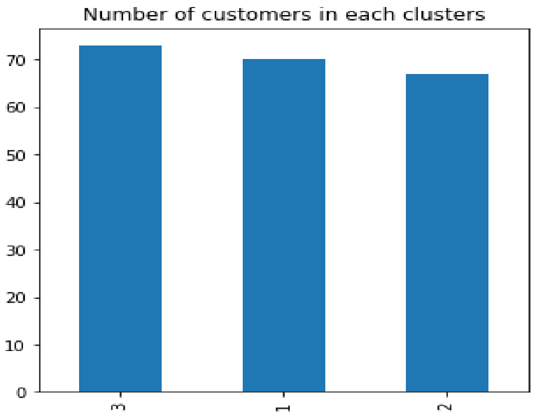
Number of clusters generated by "maxclust" and "distance " criterion is same. Here Maxclust criterion method in this dataset.

Dataset after Hierarchical clustering:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	H_clusters
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.55	1
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	3
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	2
4	17.99	15.86	0.8992	5.89	3.694	2.068	5.837	1

Number of customers in each clustering:

Clusters	No of Customers
1	70
2	67
3	73



Inference

In hierarchical clustering, 3 clusters were generated using Maxclust criterion

- Cluster 1 have 70 customers
- cluster 2 have 67 customers
- cluster 3 have 73 customers

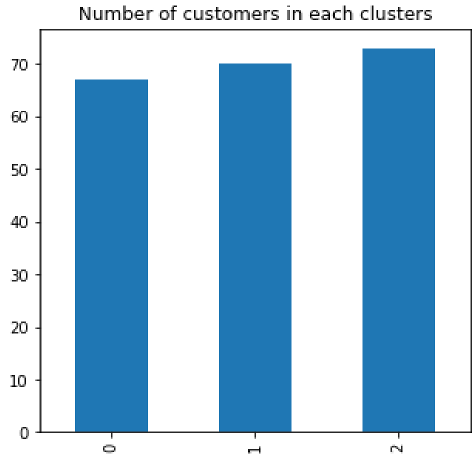
Cluster Profiling of hierarchical clusters:

H_clusters	1	2	3
spending	18.37143	11.87239	14.19904
advance_payments	16.14543	13.25702	14.23356
probability_of_full_payment	0.8844	0.848072	0.87919
current_balance	6.158171	5.23894	5.478233
credit_limit	3.684629	2.848537	3.226452
min_payment_amt	3.639157	4.949433	2.612181
max_spent_in_single_shopping	6.017371	5.122209	5.086178
freq	70	67	73

Inference

1. Cluster1 consists of high-profile customers with more spending and high advance payment with more credit limit than others. Probability of full payment is 88.4%
2. Cluster2 consists of low-profile customers with very less spending and less advance payments customers with incredibly low credit limit. Probability of full payment is 84.8%
3. Cluster3 consists of medium level customer with moderate spending and moderate level of advance payment customers with low minimum payment amount, Probability of full payment is 87.9%

Agglomerative Hierarchical Clustering



Inference

In Agglomerative Cluster, 3 clusters were generated

- Cluster 0 have 67 customers
- cluster 1 have 70 customers
- cluster 2 have 73 customers

Inference:

Cluster Profiling of Agglomerative clusters:

1. Cluster0 consists of low spending and low advance payments customers with low credit limit than other cluster customers. Probability of full payment is 85.0%
2. Cluster1 consists of high spending and high advance payments customers with good credit limit. Probability of full payment is 88.4%
3. Cluster2 consists of medium level spending and medium level advance payment customers with incredibly low minimum payment amount than other cluster customers. Probability of full payment is 87.6%

Based on comparing hierarchical and Agglomerative clusters, both type of clusters yields same results

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

Creating Clusters using K-Means

Forming 3 Clusters with K=3

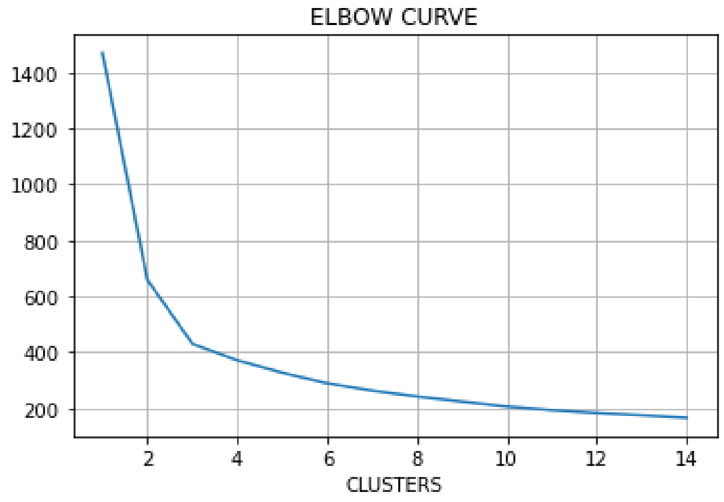
```
array([2, 0, 2, 1, 2, 1, 1, 0, 2, 1, 2, 0, 1, 2, 0, 1, 0, 1, 1, 1, 1, 1,
       2, 1, 0, 2, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 2, 2, 0, 2, 2,
       1, 1, 0, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 1, 2, 0, 1, 1, 0, 0, 2,
       2, 0, 2, 1, 0, 1, 2, 2, 1, 2, 0, 1, 2, 0, 0, 0, 0, 2, 1, 0, 2, 0,
       2, 1, 0, 2, 0, 1, 1, 2, 2, 2, 1, 2, 0, 2, 0, 2, 0, 2, 2, 1, 1, 2,
       0, 0, 2, 1, 1, 2, 0, 0, 1, 2, 0, 1, 1, 1, 0, 0, 2, 1, 0, 0, 1, 0,
       0, 2, 1, 2, 2, 1, 2, 0, 0, 0, 1, 1, 0, 1, 2, 1, 0, 1, 0, 1, 0, 0,
       1, 0, 0, 1, 0, 2, 2, 1, 2, 2, 2, 1, 0, 0, 0, 1, 0, 1, 0, 2, 2, 2,
       0, 1, 0, 1, 0, 0, 0, 0, 2, 2, 1, 0, 0, 1, 1, 0, 1, 2, 0, 2, 2, 1,
       2, 1, 0, 2, 0, 1, 2, 0, 2, 0, 0, 0])
```

- Within cluster sum of squares for 3 clusters = 430.658

Calculating WSS for other values of K - Elbow Method

```
[1469.9999999999998,
 659.171754487041,
 430.6589731513006,
 371.38509060801096,
 327.21278165661346,
 289.31599538959495,
 262.98186570162267,
 241.81894656086033,
 223.91254221002725,
 206.39612184786694,
 193.2835133180646,
 182.97995389115258,
 175.11842017053073,
 166.02965682631788]
```

Elbow curve



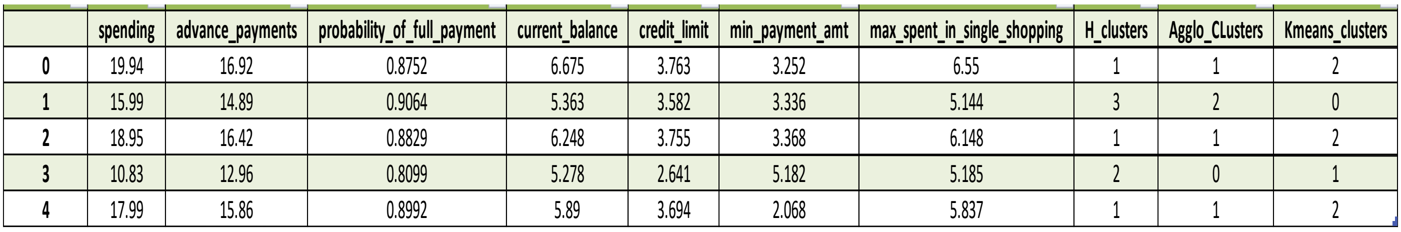
Inference

The optimum number of clusters from the WSS plot and Elbow curve is 3

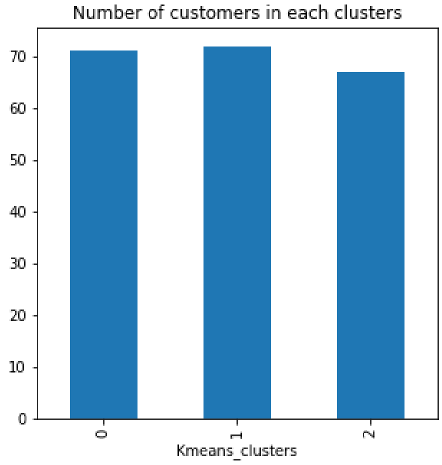
Average silhouette score and silhouette width of the clusters

- The silhouette score for 2 clusters: `0.46577247686580914`
- The silhouette width for 2 clusters: `-0.006171238927461077`
- The silhouette score for 3 clusters: `0.4007270552751299`
- The silhouette width for 3 clusters: `0.002713089347678533`

Data Frame with K-means clusters



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	H_clusters	Agglo_CLusters	Kmeans_clusters
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.55	1	1	2
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	3	2	0
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1	1	2
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	2	0	1
4	17.99	15.86	0.8992	5.89	3.694	2.068	5.837	1	1	2



Inference:

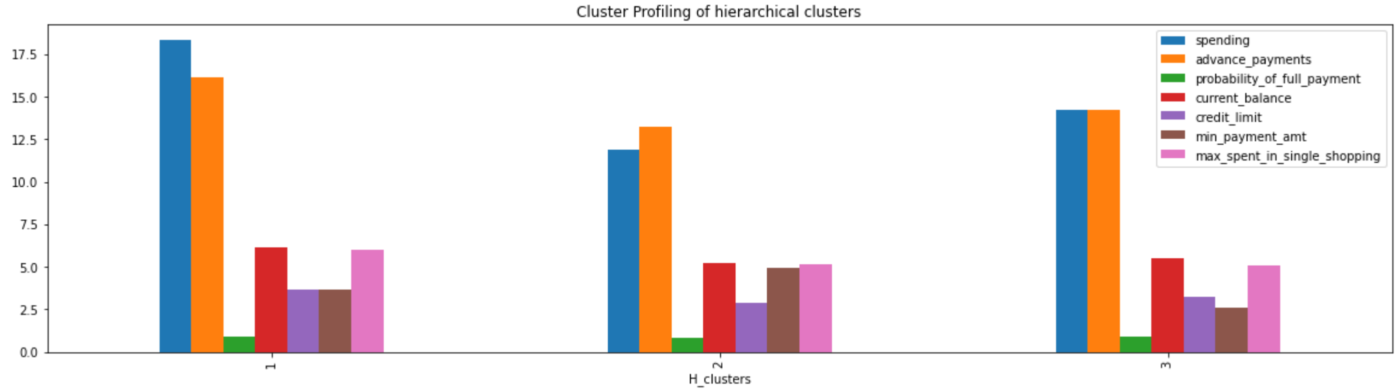
- Cluster0 = 71 customers
- Cluster1 = 72 customers
- Cluster2 = 67 customers
- After comparing WSS for other values of K - Elbow Method, optimal value of cluster is 3.
- Silhouette score of 3 clusters = 0.400727
- silhouette width of 3 clusters = 0.002713

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

- **Cluster Profiling**

- **Hierarchical cluster profiling**

H_clusters	1	2	3
spending	18.371429	11.872388	14.199041
advance_payments	16.145429	13.257015	14.233562
probability_of_full_payment	0.8844	0.848072	0.87919
current_balance	6.158171	5.23894	5.478233
credit_limit	3.684629	2.848537	3.226452
min_payment_amt	3.639157	4.949433	2.612181
max_spent_in_single_shopping	6.017371	5.122209	5.086178
freq	70	67	73



Inference:

Cluster Profiling of hierarchical clusters:

Cluster1:

- There are 70 customers in this cluster
- Customers spending on average of 18,371.4 which is more than other cluster customers.
- Advance payment by cash is high. On average they pay 1,614.54 in cash
- Probability of full payments is 88.4%
- These customers maintain high current balance than other two cluster customers
- These customers spend more in one single purchase which is on average of 6,017.37

Cluster2:

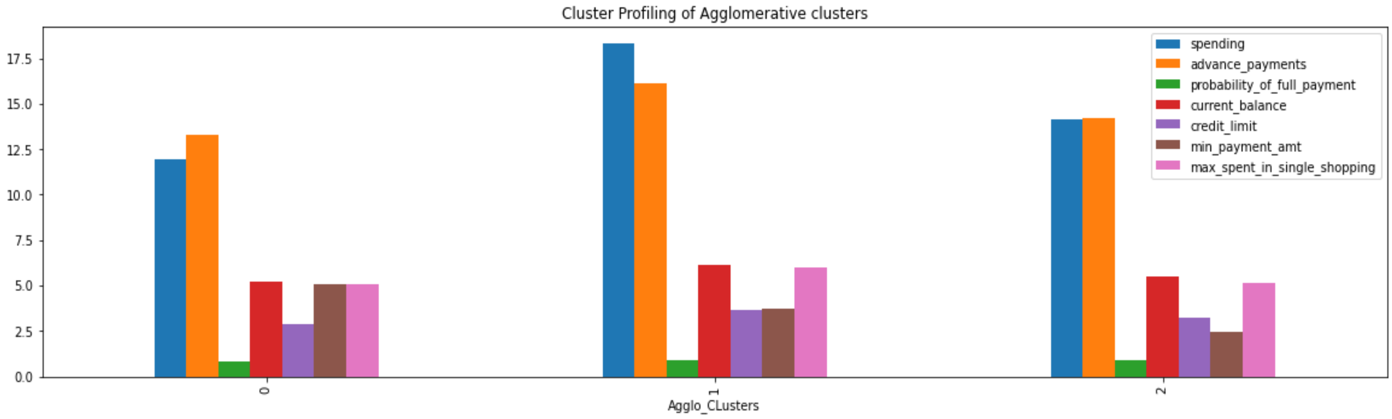
- There are 67 customers in this cluster
- These customers spend relatively less and advance payment by cash is very low than the other cluster customers
- Monthly these customers pay on average of 494.9 as a minimum payment for purchases which is higher than other cluster customers
- These customers spend more in single purchase than the customers in cluster3
- Probability of full payments is 84.8%

Cluster3:

- There are 73 customers in this cluster
- These customers spend moderately with average of 14,199 and advance payment is also in moderate level
- Here customers maintain current balance with average of 5,478
- Maximum spending on one purchase is very less

Agglomerative Cluster Profiling:

Agglo_CLusters	0	1	2
spending	11.921045	18.349714	14.175205
advance_payments	13.260448	16.138429	14.237123
probability_of_full_payment	0.85087	0.884027	0.876979
current_balance	5.22891	6.155243	5.490247
credit_limit	2.866597	3.680143	3.214178
min_payment_amt	5.040701	3.7234	2.447633
max_spent_in_single_shopping	5.097881	6.009857	5.115712
Freq_agglo	67	70	73



Inference:

Cluster Profiling of Agglomerative clusters:

- Cluster0 consists of low spending and low advance payments customers with low credit limit than other cluster customers. Probability of full payment is 85.0%

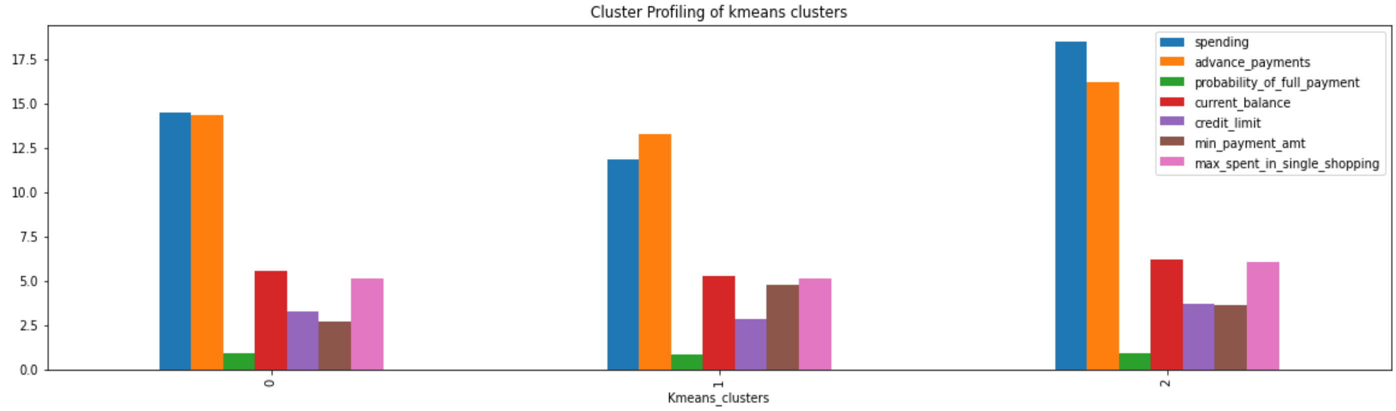
- Cluster1 consists of high spending and high advance payments customers with good credit limit. Probability of full payment is 88.4%

- Cluster2 consists of medium level spending and medium level advance payment customers with incredibly low minimum payment amount than other cluster customers. Probability of full payment is 87.6%

- **Based on comparing hierarchical and Agglomerative clusters, both type of clusters yields same results**

K-Means cluster Profiling:

```
Kmeans_clusters                        0          1          2
spending                       14.437887  11.856944  18.495373
advance_payments               14.337746  13.247778  16.203433
probability_of_full_payment     0.881597   0.848253   0.884210
current_balance                 5.514577   5.231750   6.175687
credit_limit                    3.259225   2.849542   3.697537
min_payment_amt                 2.707341   4.742389   3.632373
max_spent_in_single_shopping    5.120803   5.101722   6.041701
freq1                          71.000000  72.000000  67.000000
```



Inference:

Cluster Profiling of K-Means clusters:

K_Cluster0:

- There are 71 customers in this cluster.
- These customers moderately spend on average of 14,437 is in between K_cluster1 and k_cluster2.
- Amount paid in advance is 1,433.7 on average as cash which is slightly higher than K_cluster1.
- The current balance maintained with the average of 5514 to make purchases.
- Probability of full payment is 88.1.

K_Cluster1:

- There are 72 customers in this cluster.
- These customers spend very less per month compared to K_cluster0 and K_cluster2.
- Minimum amount paid by the customers are high than the other cluster customers.
- They maintain incredibly low credit limit.
- Probability of full payment is 84.8 which is very lowest of all clusters.

K_Cluster2:

- There are 67 customers in this cluster.
- These customers spend on average of 18,495 and amount paid is relatively higher than other two Cluster customers.
- They maintain high current balance.
- They have high credit limit despite spending more.
- On average, they spend 6,041 on one purchase.

Recommendations:

Based on cluster analysis, mainly three group of clusters are formed. They are low, medium, and high levels of spending customers.

- **Promotional strategies for low level spending customers:**

  - Make shopping more rewarding.
  - E-voucher for dining, movie, online shopping.
  - points for groceries shopping, supermarkets, etc.

- **Promotional strategies for medium level spending customers:**

  - E- voucher for Festive offers, dining, movie, online shopping.
  - Discounts in health and fitness membership, apparel shopping.
  - Pay with points.

- **Promotional strategies for medium level spending customers:**

  - Upgrade the credit card like platinum or titanium card.
  - Accelerated reward points on birthday shopping, International purchases.
  - Redemption options.
  - Quick cash using credit card.

CASE STUDY 2:

PREDICTION OF INSURANCE CLAIM STATUS USING DECISION TREE, RANDOM FOREST, AND ANN MODELS

CASE STUDY 2:

PREDICTION OF INSURANCE CLAIM STATUS USING DECISION TREE, RANDOM FOREST, AND ANN MODELS

Table of contents

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)
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)

Project Objective:

The Objective of the report is to explore the ": bank_marketing_part1_Data.csv" dataset in Python (JUPYTER NOTEBOOK) and generate insights about the dataset. This exploration report will consist of the following:

- Importing the dataset in jupyter notebook.
- Understanding the structure of dataset.
- Exploratory Data analysis
- Graphical exploration
- Predictions using Random Forest, Decision tree
- Scaling the data
- Predictions using ANN
- Insights from the dataset

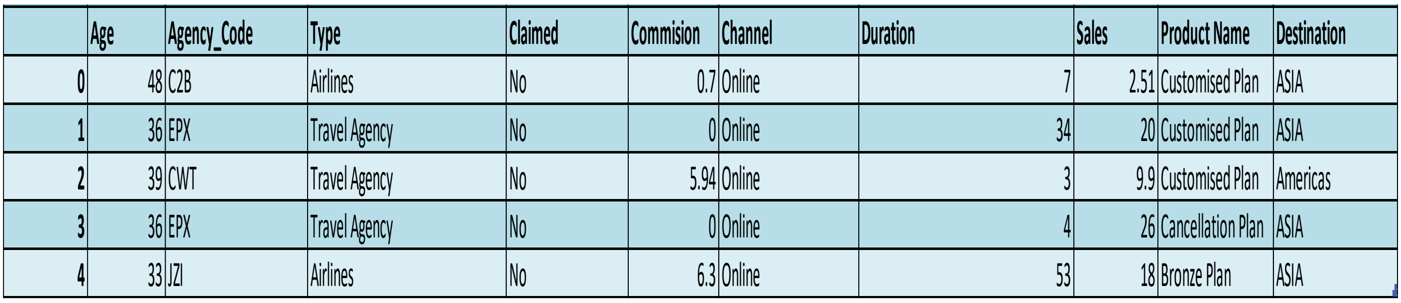
Assumptions:

The insurance companies are interested in the prediction of the company's future. Insurers are using machine learning to progress operational competence, from claims registration to claim settlement. Machine Learning and predictive models can also help insurers with a better understanding of claim cost. Accurate prediction gives probability to decrease financial loss for the company. This helps the insurance company to be one step ahead of its competitor

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

Exploratory data analysis

Insurance data set



	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	C2B	Airlines	No	0.7	Online	7	2.51	Customised Plan	ASIA
1	36	EPX	Travel Agency	No	0	Online	34	20	Customised Plan	ASIA
2	39	CWT	Travel Agency	No	5.94	Online	3	9.9	Customised Plan	Americas
3	36	EPX	Travel Agency	No	0	Online	4	26	Cancellation Plan	ASIA
4	33	JZI	Airlines	No	6.3	Online	53	18	Bronze Plan	ASIA

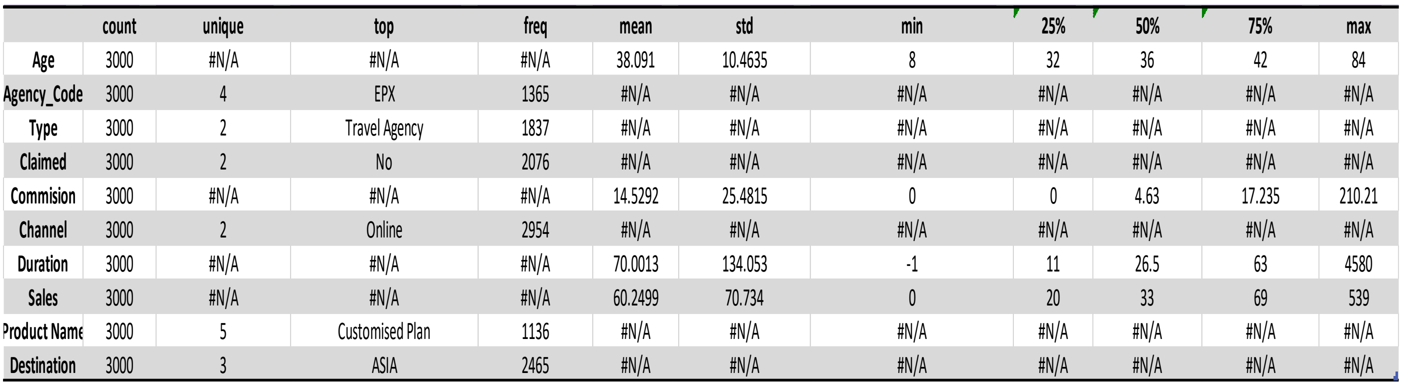
Information of the dataset:

```
<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
```

Inference:

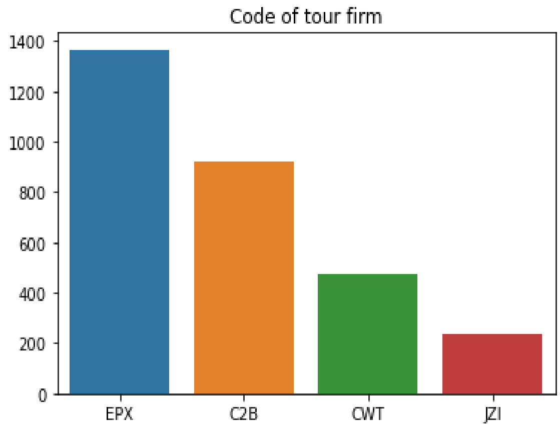
- There are no null values
- There are 3000 entries and 10 columns
- There are 6 object variables, 2 float and 2 int variables

Summary of the dataset:



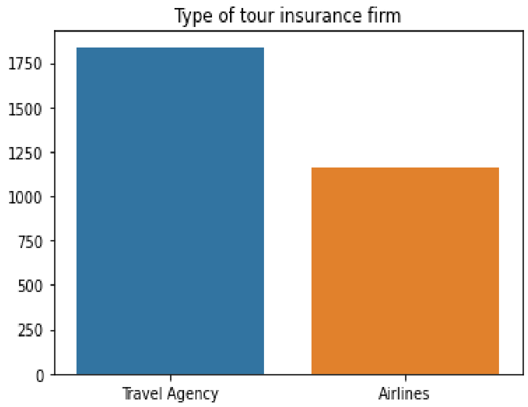
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	3000	#N/A	#N/A	#N/A	38.091	10.4635	8	32	36	42	84
Agency_Code	3000	4	EPX	1365	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
Type	3000	2	Travel Agency	1837	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
Claimed	3000	2	No	2076	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
Commision	3000	#N/A	#N/A	#N/A	14.5292	25.4815	0	0	4.63	17.235	210.21
Channel	3000	2	Online	2954	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
Duration	3000	#N/A	#N/A	#N/A	70.0013	134.053	-1	11	26.5	63	4580
Sales	3000	#N/A	#N/A	#N/A	60.2499	70.734	0	20	33	69	539
Product Name	3000	5	Customised Plan	1136	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A
Destination	3000	3	ASIA	2465	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A	#N/A

Code of tour firm



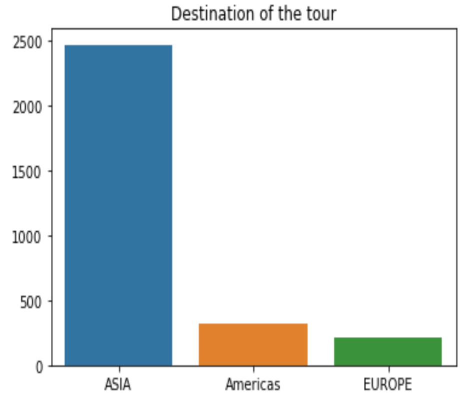
```
EPX    0.455000
C2B    0.308000
CWT    0.157333
JZI    0.079667
```

Type of tour insurance firm



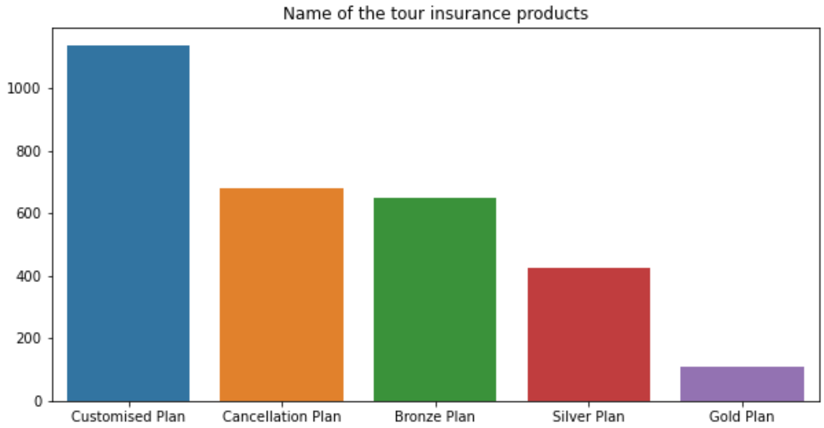
```
Travel Agency    0.612333
Airlines         0.387667
```

Destination of the tour



```
ASIA        0.821667
Americas    0.106667
EUROPE      0.071667
```

Name of the tour insurance products

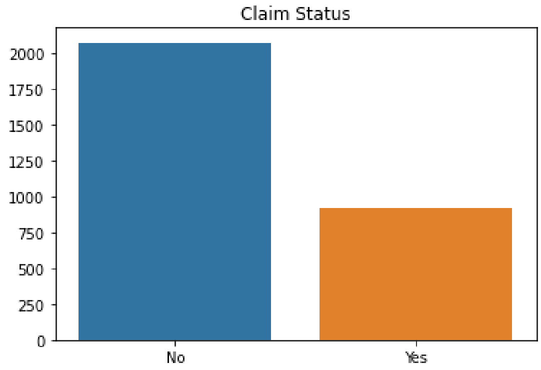


```
Customised Plan      0.378667
Cancellation Plan    0.226000
Bronze Plan          0.216667
Silver Plan          0.142333
Gold Plan            0.036333
```

Inference:

- There are outliers in most of the variables
- Most travellers used **travel agency** as type of insurance firms
- Mostly **online** distribution channel is used by travellers
- **Customized plan** is mostly preferred by the travellers
- Most travelled destination is **ASIA**

Target variable: Claim Status



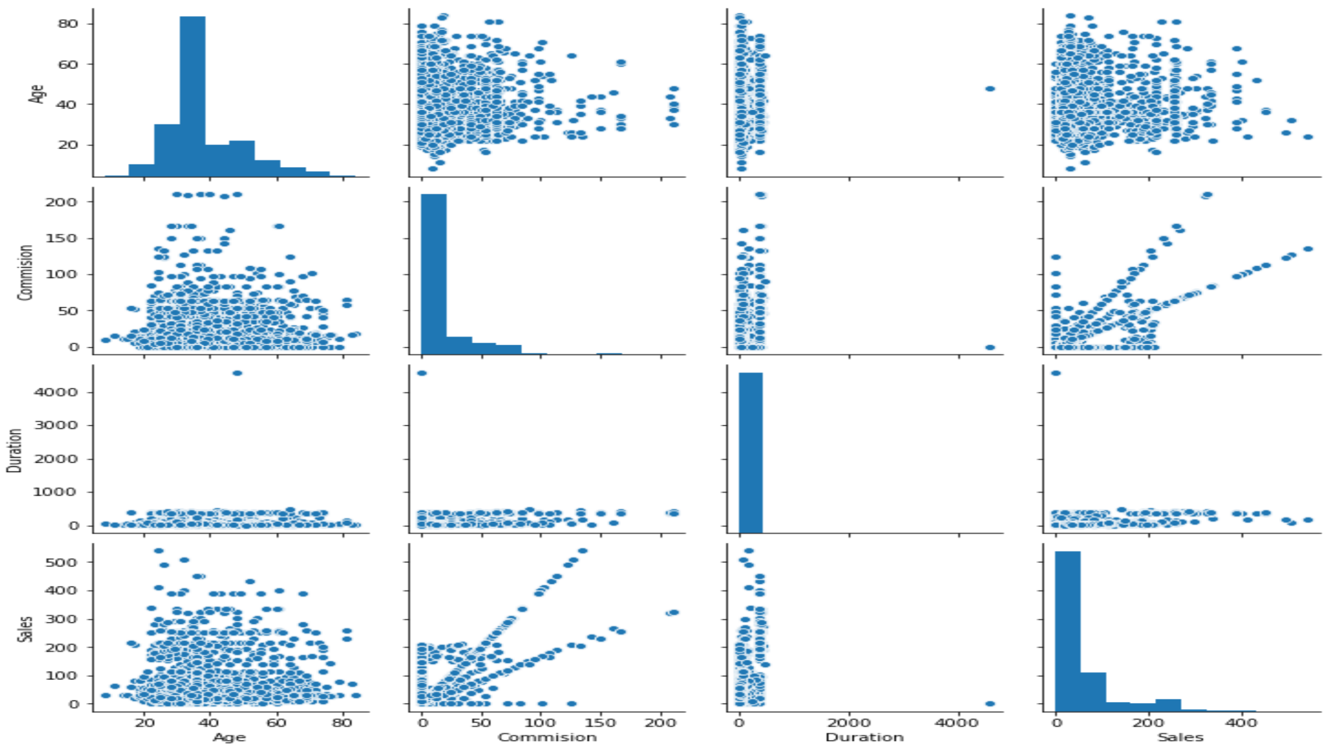
```
No      0.692
Yes     0.308
```

Inference:

- Customers with Yes claim status: 31%
- Customers with No claim status: 69%

Bivariate Analysis:

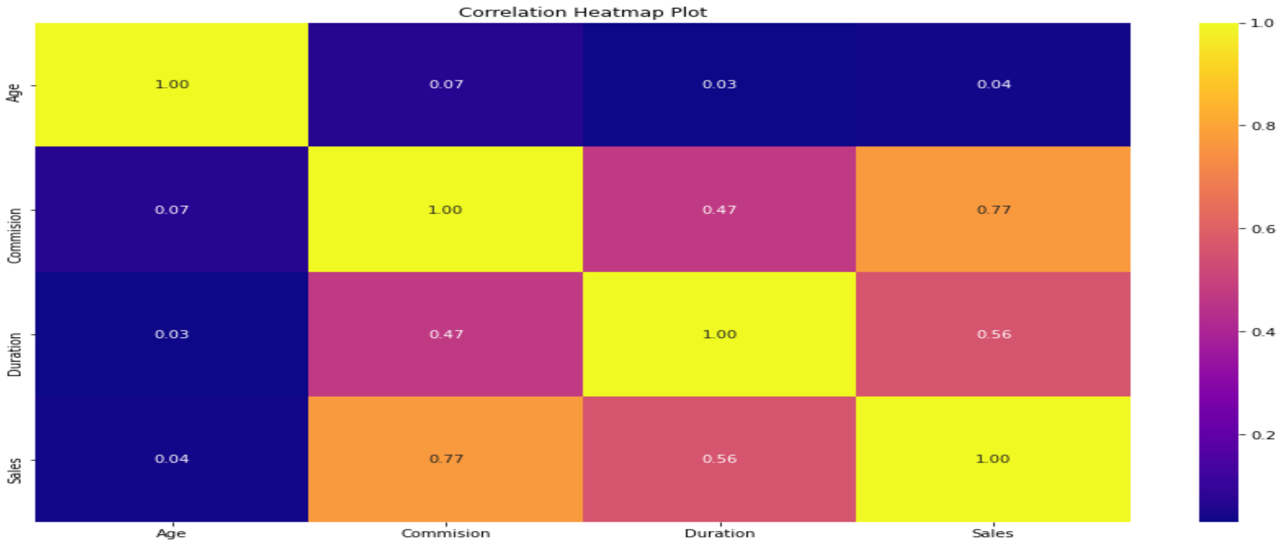
Pair plot between variables:



Inference:

Most of variables is left skewed.

Correlation Heatmap:

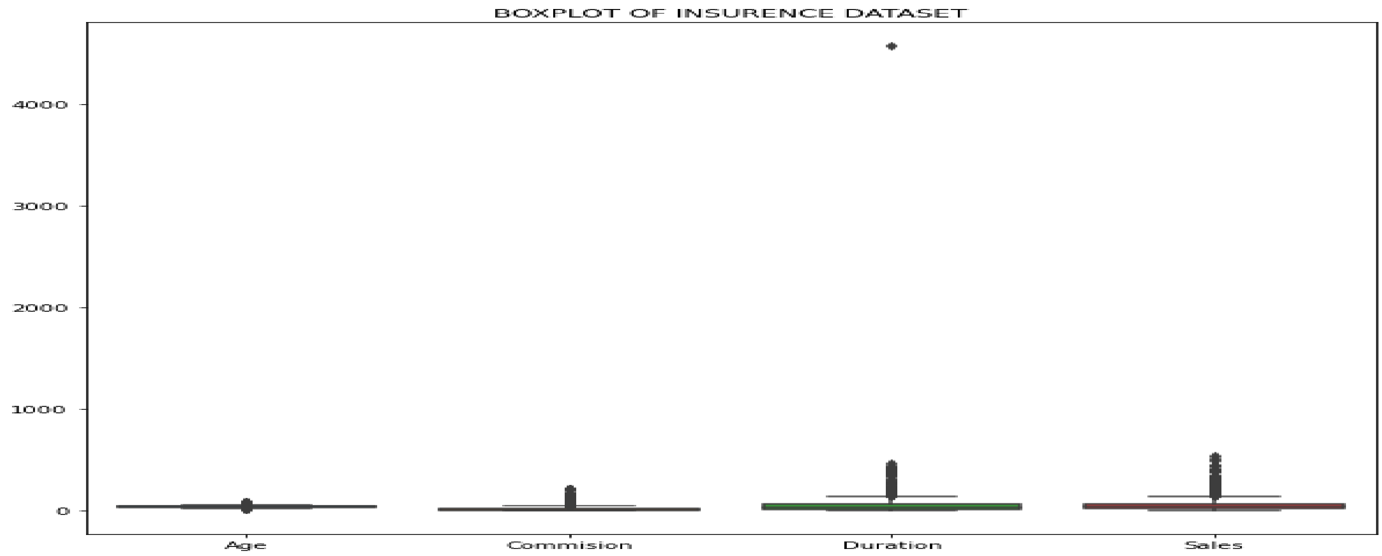


	Age	Commision	Duration	Sales
Age	1.00	0.07	0.03	0.04
Commision	0.07	1.00	0.47	0.77
Duration	0.03	0.47	1.00	0.56
Sales	0.04	0.77	0.56	1.00

Inference:

- Commision and sales are positively correlated.
- Sales and Duration are moderately correlated.

BOXPLOT OF INSURENCE DATASET:



Inference:

- There are outliers in this dataset. But outlier treatment is not done because we may lose more data information.
- There are 139 duplicates which are dropped from dataset.

Converting Object data type into Categorical:

```
feature: Agency_Code
[C2B, EPX, CWT, JZI]
Categories (4, object): [C2B, CWT, EPX, JZI]
[0 2 1 3]

feature: Type
[Airlines, Travel Agency]
Categories (2, object): [Airlines, Travel Agency]
[0 1]

feature: Claimed
[No, Yes]
Categories (2, object): [No, Yes]
[0 1]

feature: Channel
[Online, Offline]
Categories (2, object): [Offline, Online]
[1 0]

feature: Product Name
[Customised Plan, Cancellation Plan, Bronze Plan, Silver Plan, Gold Plan]
Categories (5, object): [Bronze Plan, Cancellation Plan, Customised Plan, Gold Plan
, Silver Plan]
[2 1 0 4 3]

feature: Destination
[ASIA, Americas, EUROPE]
Categories (3, object): [ASIA, Americas, EUROPE]
[0 1 2]
```

Inference:

Object and float variables are converted into integer variables for prediction models

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

- Splitting the dataset into train and test set to build a model. Here 70:30 ratio is used .70% Train data and 30% test data.
- train_test_split is done using sklearn.model_selection

Dimensions on the train and test data:

```
x_train:  (2002, 9)
x_test:  (859, 9)
y_train:  (2002,)
y_test:  (859,)
```

Decision tree – CART Model:

A decision tree is a flowchart-like tree structure where an internal node represents feature (or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition based on the attribute value. It partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps you in decision making. It is visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.

- DecisionTreeClassifier is used for Decision tree model from sklearn.tree
- Using GridsearchCV, we input various parameters like max_depth, max_features ,min_samples, min_sample_split which will helps us to find best grid for prediction of the better model

Variable Importance:

```
                IMP
Agency_Code   0.361695
Sales         0.217361
Duration      0.139983
Commision     0.121831
Age           0.105200
Product Name  0.038207
Destination   0.015385
Channel       0.000337
Type          0.000000
```

Inference:

DecisionTreeClassifier

- best grid:DecisionTreeClassifier(max_depth=10, max_features=5, min_samples_leaf=5,min_samples_split=4, random_state=1)
- Best grid shows which variable is important. In this dataset, Agency_Code is more important feature.
- Next important features are Sales and Duration.

CART- Predicting on Training and Test dataset

Inference:

- Test score of train data: 0.828
- Test score of test data: 0.745

Random Forest model

Random forests are a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a good indicator of the feature importance.

- RandomForestClassifier is used for Random forests models from sklearn.ensemble
- Using GridsearchCV, we input various parameters like max_depth, max_features, min_samples, min_sample_split which will helps us to find best grid for prediction of the better model

Variable Importance:

```
                   IMP
Sales          0.206312
Agency_Code    0.179092
Product Name   0.152201
Duration       0.149934
Commision      0.141171
Age            0.110455
Type           0.035787
Destination    0.020322
Channel        0.004727
```

Inference:

- best grid: RandomForestClassifier(max_depth=10, min_samples_leaf=4, min_samples_split=4,n_estimators=200, random_state=1)
- Best grid shows which variable is important. In this dataset, Sales is more important feature
- Next important features are Agency_Code, Product.

Random Forest- Predicting on Training and Test dataset:

- Test score of train data: 0.83
- Test score of test data: 0.79

Artificial neural networks (ANN):

Multi-layer Perceptron classifier (MLPClassifier)

Class **MLPClassifier** implements a multi-layer perceptron (MLP) algorithm that trains using Backpropagation. MLP trains on two arrays: array X of size (n_samples, n_features), which holds the training samples represented as floating point feature vectors; and array y of size (n_samples,), which holds the target values (class labels) for the training samples

- MLPClassifier is used for ANN model from sklearn.neural_network.
- This model optimizes the log-loss function using LBFGS or stochastic gradient descent.
- Using GridsearchCV, we input various parameters like 'hidden_layer_sizes', 'max_iter', 'solver','tol 'which will help us to find best grid for prediction of the better model.

Inference:

- best grid: MLPClassifier(hidden_layer_sizes=300, max_iter=1000, random_state=1, tol=0.001)
- Test score of train data: 0.77
- Test score of test data: 0.77

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

Performance of Predictions on Train and Test sets

Decision Tree- CART Prediction

CART- Predicting on Training and Test dataset

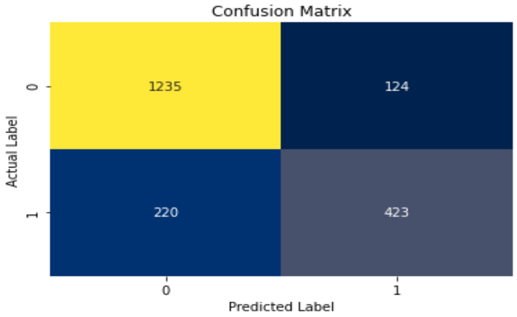
Model Evaluation

Trained data

Train Data Accuracy:

- Accuracy score of cart_train_data: 0.828

Confusion Matrix for the training data:



Inference:

- **True Positive:** Actual and predict claim status is yes - 423 customers
- **True Negative:** Actual and predict claim status is no - 1235 customers
- **False Positive**: Actual claim status is no, but predict claim status is yes - 124 customers
- **False Negative**: Actual claim status is yes, but predict claim status is no- 220 customers

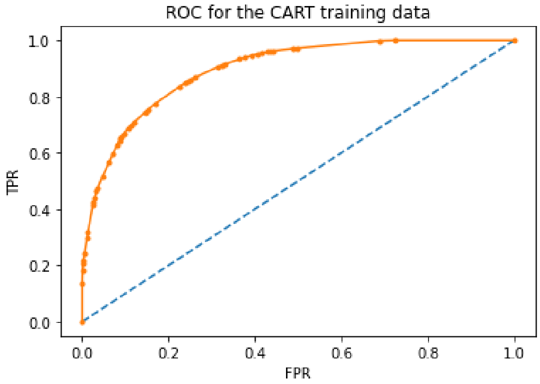
Classification Report of CART Train data:

```
              precision    recall  f1-score   support
           0       0.85      0.91      0.88      1359
           1       0.77      0.66      0.71       643

    accuracy                           0.83      2002
   macro avg       0.81      0.78      0.79      2002
weighted avg       0.82      0.83      0.82      2002
```

AUC and ROC for the CART training data:

```
Cart_train_AUC: 0.896
```

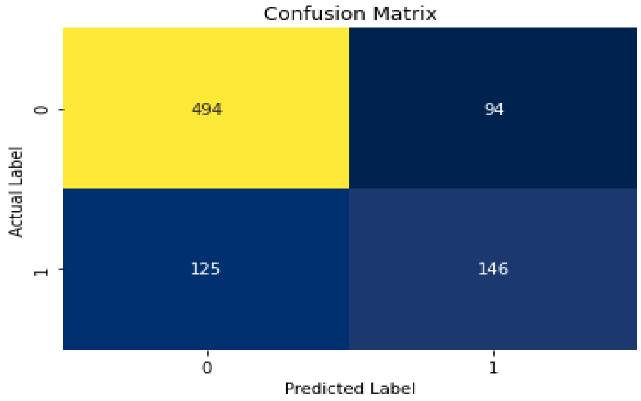


Test data

Test Data Accuracy:

Accuracy score of cart_test_data: 0.745

Confusion Matrix for test data:



Inference:

- True Positive: Actual and predict claim status is yes -146 customers
- True Negative: Actual and predict claim status is no - 494 customers
- False Positive: Actual claim status is no, but predict claim status is yes - 94 customers
- False Negative: Actual claim status is yes, but predict claim status is no- 125 customers

Classification Report of Test data:

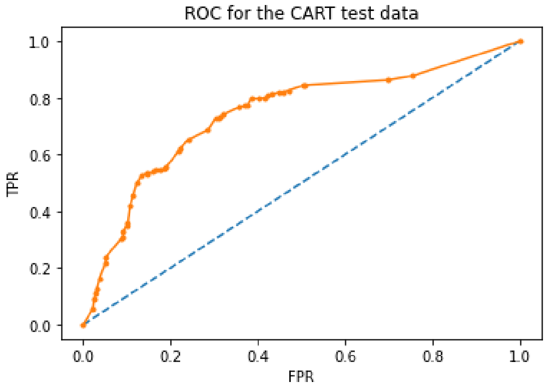
```
              precision    recall  f1-score   support

           0       0.80      0.84      0.82       588
           1       0.61      0.54      0.57       271

    accuracy                           0.75       859
   macro avg       0.70      0.69      0.69       859
weighted avg       0.74      0.75      0.74       859
```

AUC and ROC for the CART test data:

Cart_test_AUC: 0.742



CART MODEL CONCLUSION

Train Data:

AUC: 90%

Accuracy: 83%

Precision: 77%

Recall: 66%

f1-Score: 71%

Test Data:

AUC: 74%

Accuracy: 75%

Precision: 61%

Recall: 54%

f1-Score: 57%

Training and Test set results are not similar, this proves, there may be overfitting

Agency_Code is the most important variable for predicting claim status

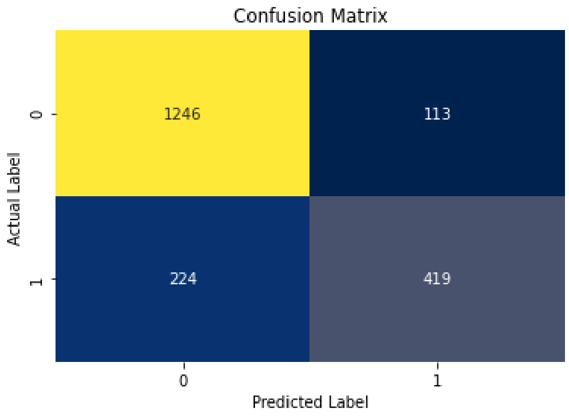
Random Forest Prediction

Predicting on Training and Test dataset

Train Data Accuracy:

Accuracy score of Random_Forest_train_data: 0.83

Confusion Matrix for the training data:



Inference:

Random Forest Trained data

- ***True Positive****: Actual and predict claim status is yes - 419 customers*
- ***True Negative****: Actual and predict claim status is no - 1246 customers*
- ***False Positive****: Actual claim status is no, but predict claim status is yes - 113 customers*
- ***False Negative****: Actual claim status is yes, but predict claim status is no- 224 customers*

Classification Report of Training data:

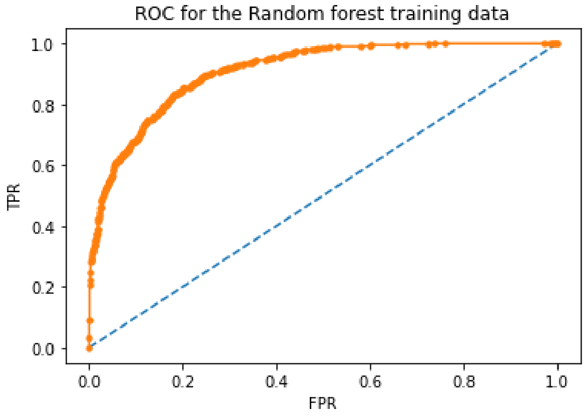
```
              precision    recall  f1-score   support

           0       0.85      0.92      0.88      1359
           1       0.79      0.65      0.71       643

    accuracy                           0.83      2002
   macro avg       0.82      0.78      0.80      2002
weighted avg       0.83      0.83      0.83      2002
```

AUC and ROC for the Random Forest training data:

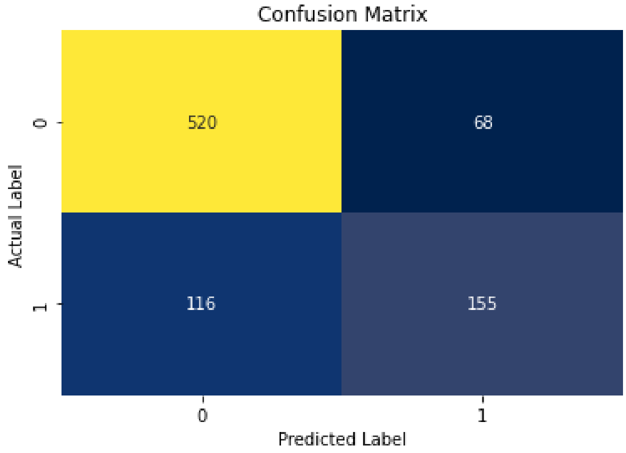
```
rf_train_AUC: 0.910
```



Test Data Accuracy:

```
Accuracy score of Random_Forest_test_data: 0.785
```

Confusion Matrix for the test data:



Inference:

Random Forest Test data:

- True Positive: Actual and predict claim status is yes -155 customers
- True Negative: Actual and predict claim status is no - 520 customers
- False Positive: Actual claim status is no, but predict claim status is yes - 68 customers
- False Negative: Actual claim status is yes, but predict claim status is no- 116 customers

Classification Report of Test data:

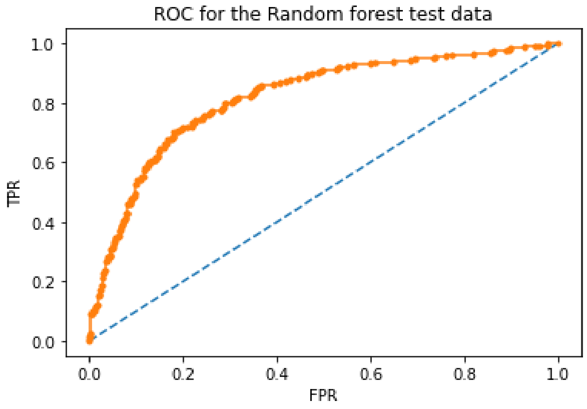
```
              precision    recall  f1-score   support

           0       0.82      0.88      0.85       588
           1       0.70      0.57      0.63       271

    accuracy                           0.79       859
   macro avg       0.76      0.73      0.74       859
weighted avg       0.78      0.79      0.78       859
```

AUC and ROC for the Random Forest test data:

```
rf_test_AUC: 0.819
```



Inference:

RANDOM FOREST MODEL CONCLUSION:

Train Data:

AUC: 91%

Accuracy: 83%

Recall: 65%

Precision: 79%

f1-Score: 71%

Test Data:

AUC: 82%

Accuracy: 79%

Recall: 57%

Precision: 70%

f1-Score: 63%

Training and Test set results have different outcomes, this proves, there may have overfitting

Here also, Agency_Code is the most important variable for predicting claim status

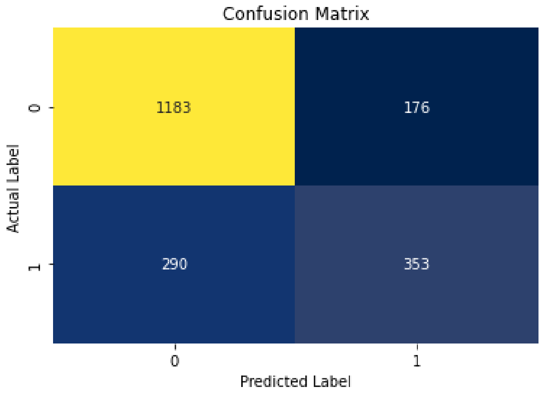
Artificial Neural Network (ANN) model Prediction

Predicting on Training and Test dataset

ANN Train Data Accuracy:

```
Accuracy score of Ann_train_data: 0.767
```

Confusion Matrix for the training data:



Inference:

- True Positive: Actual and predict claim status is yes -353 customers
- True Negative: Actual and predict claim status is no - 1183 customers
- False Positive: Actual claim status is no, but predict claim status is yes - 176 customers
- False Negative:Actual claim status is yes, but predict claim status is no- 290 customers

Classification Report of ANN Training data:

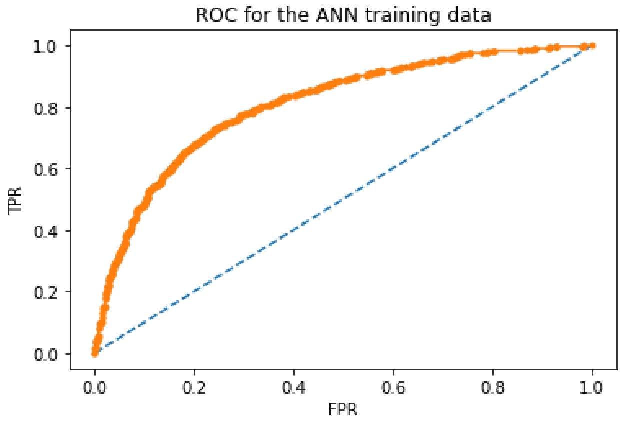
```
              precision    recall  f1-score   support

           0       0.80      0.87      0.84      1359
           1       0.67      0.55      0.60       643

    accuracy                           0.77      2002
   macro avg       0.74      0.71      0.72      2002
weighted avg       0.76      0.77      0.76      2002
```

AUC and ROC for the ANN training data:

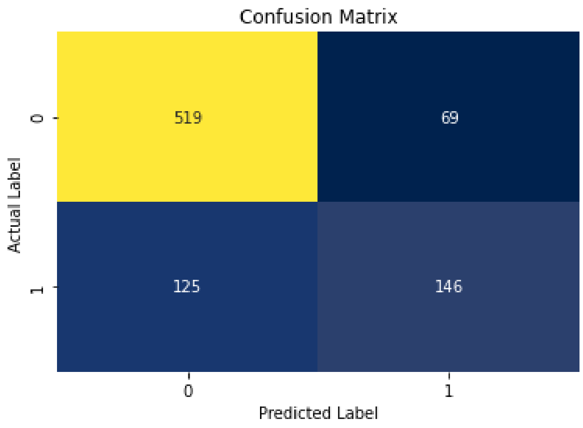
```
ann_train_AUC: 0.809
```



ANN Test Data Accuracy

```
Accuracy score of Ann_test_data: 0.774
```

Classification Report of ANN Test data



Inference:

- True Positive: Actual and predict claim status is yes -146 customers
- True Negative: Actual and predict claim status is no - 519 customers
- False Positive: Actual claim status is no, but predict claim status is yes - 69 customers
- False Negative: Actual claim status is yes, but predict claim status is no- 125 customers

Classification Report of Test data

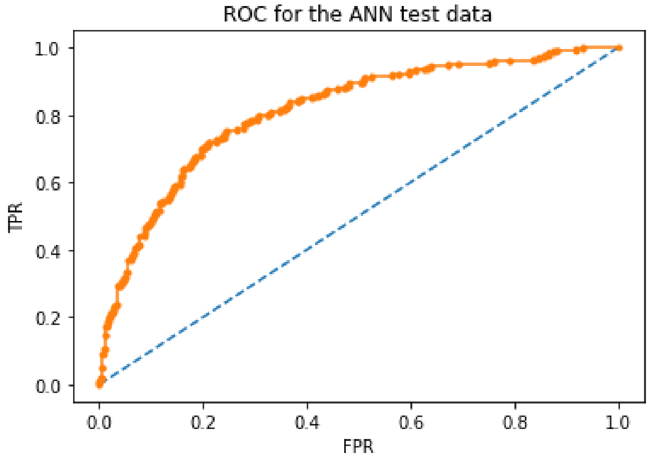
```
              precision    recall  f1-score   support

           0       0.81      0.88      0.84       588
           1       0.68      0.54      0.60       271

    accuracy                           0.77       859
   macro avg       0.74      0.71      0.72       859
weighted avg       0.77      0.77      0.77       859
```

AUC and ROC for the ANN test data

```
ann_test_AUC: 0.813
```



Inference

Artificial Neural Network

Train Data:

AUC: 81%

Accuracy: 77%

Recall: 55%

Precision: 67%

f1-Score: 60%

Test Data:

AUC: 80%

Accuracy: 77%

Precision: 69%

f1-Score: 57%

Training and Test set results are almost similar, this proves no overfitting or underfitting

The Precision and Recall metrics also almost similar between training and test set

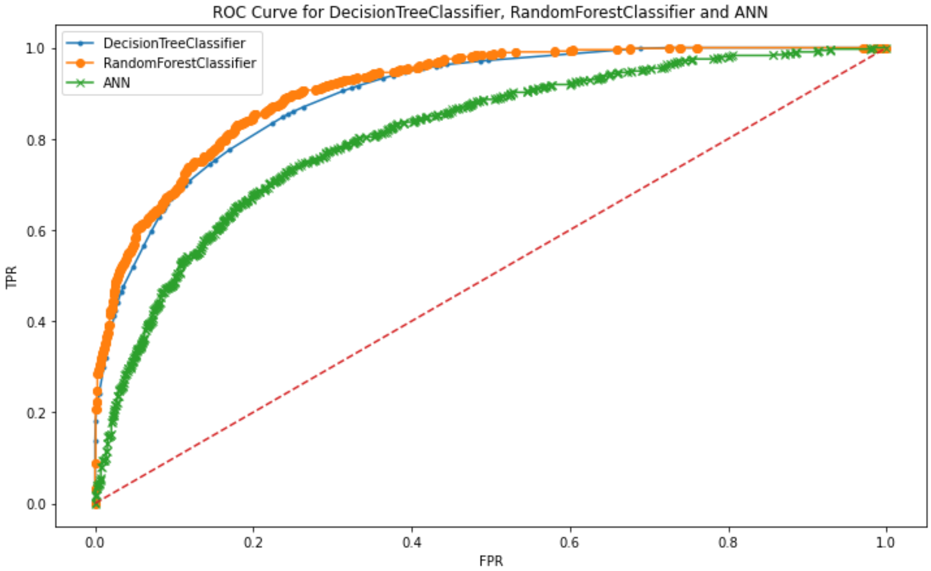
2.4 Final Model: Compare all the model and write an inference which model is best/optimized?

Comparison of the performance metrics from the 3 models

	CART Train	CART Test	Random Forest Train	Random Forest Test	ural Network Tr	Neural Network Test
Accuracy	0.83	0.75	0.83	0.79	0.77	0.77
AUC	0.9	0.74	0.91	0.82	0.81	0.81
Recall	0.66	0.54	0.65	0.57	0.55	0.54
Precision	0.77	0.61	0.79	0.7	0.67	0.68
F1 Score	0.71	0.57	0.71	0.63	0.6	0.6

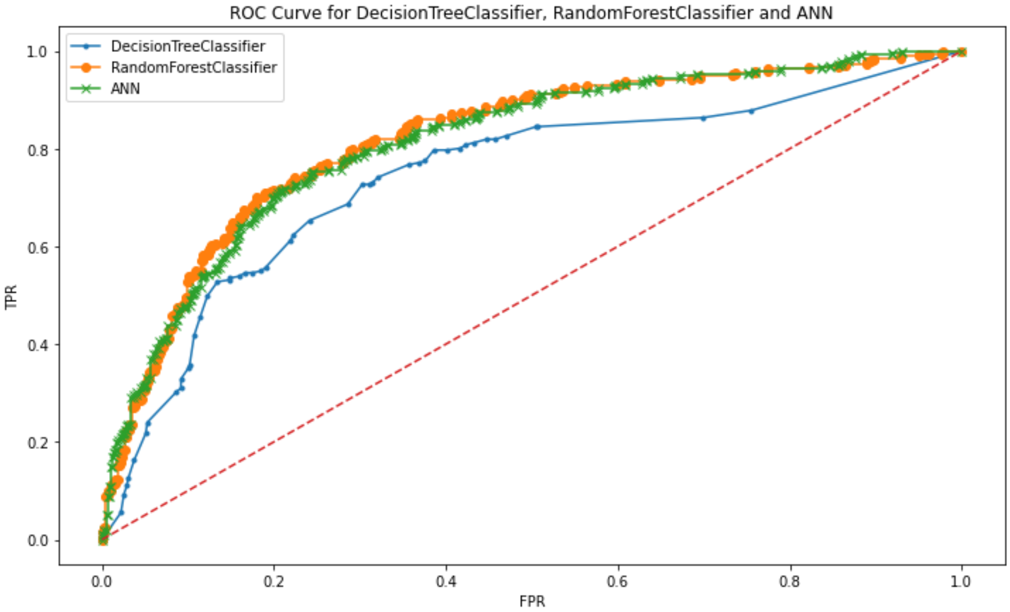
ROC Curve for the 3 models on the Training data

```
Area under the curve for Decision Tree Classification Model is 0.896
Area under the curve for Random Forest Classification Model is 0.910
Area under the curve for Artificial Neural Network Model is 0.808
```



ROC Curve for the 3 models on the Test data

```
Area under the curve for Decision Tree Classification Model is 0.742
Area under the curve for Random Forest Classification Model is 0.818
Area under the curve for Artificial Neural Network Model is 0.812
```



Conclusion:

- Artificial Neural Network (ANN) performs better than Decision Tree and Random Forest
- Train and test score of ANN is 77%
- AUC of Train and test in ANN is 81%
- f1 score of Train and test in ANN is 60%
- Accuracy, AUC, Precision, Recall for test data are almost in line with training data in Artificial Neural Network (ANN). This indicates no overfitting or underfitting in the model
- Although ANN models are moderate enough for prediction. But still, the model is more useful only in class 0 than class 1. This is because the dataset is unbalanced and we may have a class imbalance problem
- We need to collect more data to build a good model.
- Both Random Forest model and Decision Tree (CART) have overfitting data. Because Random Forest model and Decision Tree (CART) modelled the training data too well and therefore the model is weak in generalising and predicting any new data.

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

Recommendations:

- Prediction Models needs to be trained with huge volumes of documents/transactions to cover all possible scenarios.
- In machine learning, Right data source and quality of data used to train predictive models is equally important as the quantity
- Accurate prediction gives a probability to decrease financial loss for the company
- In our dataset, there are not enough to train the data, so it created class imbalance problem during the prediction
- we recommend the client to collect more data in quality and quantity wise
- Also, Insurance company should concentrate more in the variables like Agency_code, Sales, Duration data. These variables will helpful in good future prediction.

Reference

- *https://towardsdatascience.com/for-real-auto-insurance-fraud-claim-detection-with-machine-learning-efcf957b38f3*
- *https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/*
- *https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d*
- *https://www.youtube.com/watch?v=355u2bDqB7c&ab_channel=KrishNaik*
- *https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/*
- *https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/*