DATA MINING PROJECT REPORT

By Karthik Sreeram R

OCTOBER 24, 2020

UNIVERSITY OF TEXAS AT AUSTIN AND GREAT LAKES

Purpose
This document is the business report for my final project in the subject “Data Mining This document gives us a detailed explanation of various approaches used, their insight and inferences. Tools used analysis: Python and Jupiter notebook. Packages used: NumPy, pandas, seaborn, os, matplotlib, SciPy, stats model , sklearn and sweetviz

Contents

Problem 1 : Clustering

Business scenario

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

1.1 Read the data and do exploratory data analysis. Describe the data briefly.

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.

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

1.1) Read the data and do exploratory data analysis. Describe the data briefly.

a.) Dataset Head

	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

Inference :

Dataset has 7 columns .

b.) Type of the variables in 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 :

Dataset has No null values and all variables are of float data type

c.) Summary of the dataset:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	210.000000	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
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

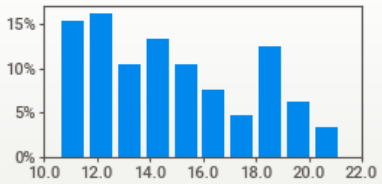
Inference :

While comparing the maximum value , Minimum value , average and percentiles , data appears not to have any major outliers .
But the data value range differs from variable to variable. For example spending column has data value range between 10 and 21. Whereas min_paymnet_amt column has value range from 0.76 to 8.45 .

d) Data visualization

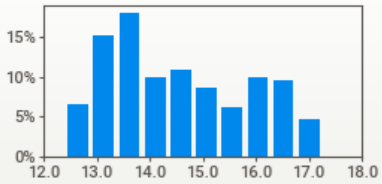
spending

VALUES:	210 (100%)	MAX	21.2	RANGE	10.6
MISSING:	---	95%	19.5	IQR	5.04
		Q3	17.3	STD	2.91
DISTINCT:	193 (92%)	AVG	14.8	VAR	8.47
		MEDIAN	14.4		
		Q1	12.3	KURT.	-1.08
		5%	11.2	SKEW	0.400
		MIN	10.6	SUM	3,118



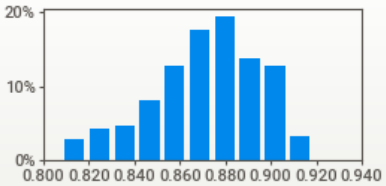
advance_payments

VALUES:	210 (100%)	MAX	17.25	RANGE	4.84
MISSING:	---	95%	16.73	IQR	2.27
		Q3	15.71	STD	1.31
DISTINCT:	170 (81%)	AVG	14.56	VAR	1.71
		MEDIAN	14.32		
		Q1	13.45	KURT.	-1.11
		5%	12.86	SKEW	0.387
		MIN	12.41	SUM	3,057



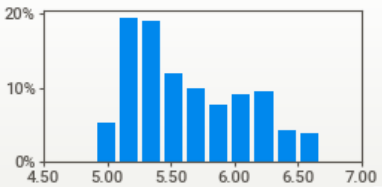
probability_of_full_payment

VALUES:	210 (100%)	MAX	0.918	RANGE	0.110
MISSING:	---	95%	0.905	IQR	0.031
		Q3	0.888	STD	0.024
DISTINCT:	186 (89%)	MEDIAN	0.873	VAR	0.00
		AVG	0.871		
		Q1	0.857	KURT.	-0.140
		5%	0.826	SKEW	-0.538
		MIN	0.808	SUM	183



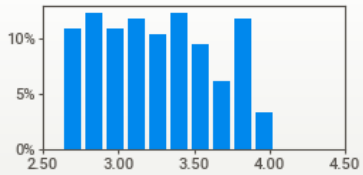
current_balance

VALUES:	210 (100%)	MAX	6.67	RANGE	1.78
MISSING:	---	95%	6.45	IQR	0.718
		Q3	5.98	STD	0.443
DISTINCT:	188 (90%)	AVG	5.63	VAR	0.196
		MEDIAN	5.52		
		Q1	5.26	KURT.	-0.786
		5%	5.08	SKEW	0.525
		MIN	4.90	SUM	1,182



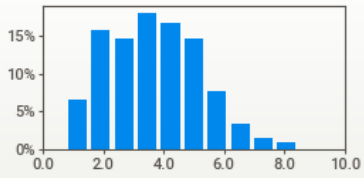
credit_limit

VALUES:	210 (100%)	MAX	4.03	RANGE	1.40
MISSING:	---	95%	3.86	IQR	0.618
		Q3	3.56	STD	0.378
DISTINCT:	184 (88%)	AVG	3.26	VAR	0.143
		MEDIAN	3.24		
		Q1	2.94	KURT.	-1.10
		5%	2.69	SKEW	0.134
		MIN	2.63	SUM	684



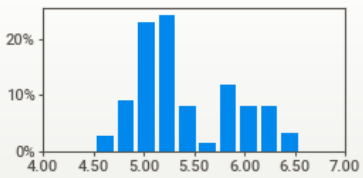
min_payment_amt

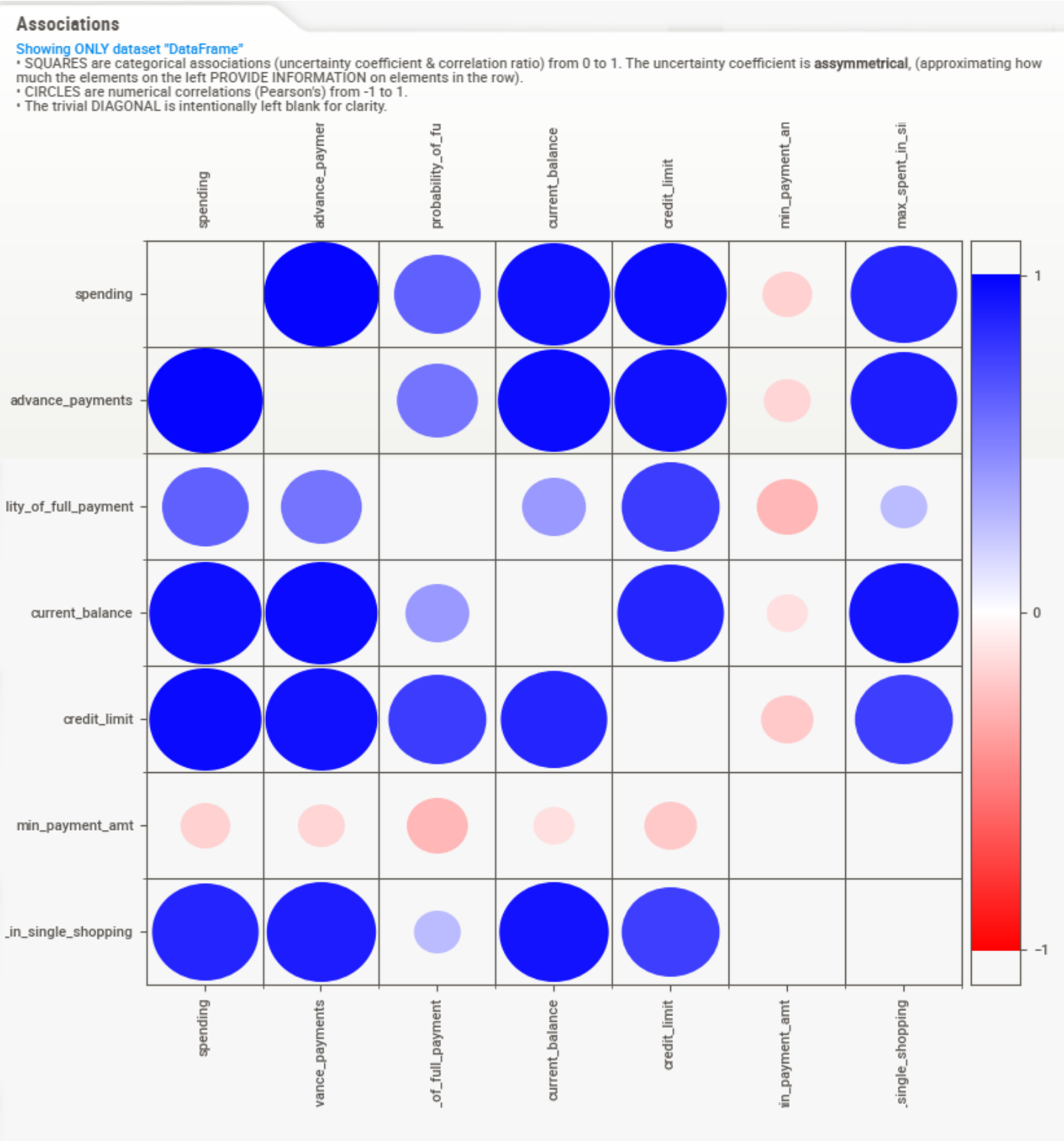
VALUES:	210 (100%)	MAX	8.46	RANGE	7.69
MISSING:	---	95%	6.18	IQR	2.21
		Q3	4.77	STD	1.50
DISTINCT:	207 (99%)	AVG	3.70	VAR	2.26
		MEDIAN	3.60		
		Q1	2.56	KURT.	-0.067
		5%	1.47	SKEW	0.402
		MIN	0.77	SUM	777



max_spent_in_single_shopping

VALUES:	210 (100%)	MAX	6.55	RANGE	2.03
MISSING:	---	95%	6.27	IQR	0.832
		Q3	5.88	STD	0.491
DISTINCT:	148 (70%)	AVG	5.41	VAR	0.242
		MEDIAN	5.22		
		Q1	5.04	KURT.	-0.841
		5%	4.78	SKEW	0.562
		MIN	4.52	SUM	1,136





Inference :

There is a huge correlation with credit limit and other variables. There is correlation between most of the variables.

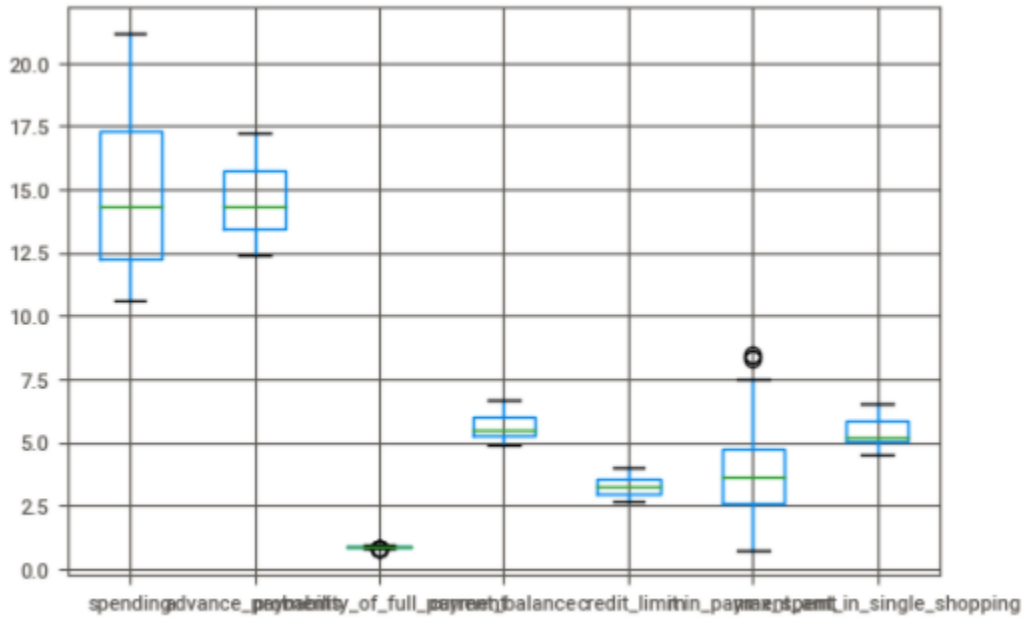
e.) Duplicated data

```
Number of Duplicates  0
```

Inference :

No duplicated data in the dataset

f.) Box plot to check the outliers



Inference :

The Dataset does not appear to have any outliers

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

Justification

Yes. Scaling is required.

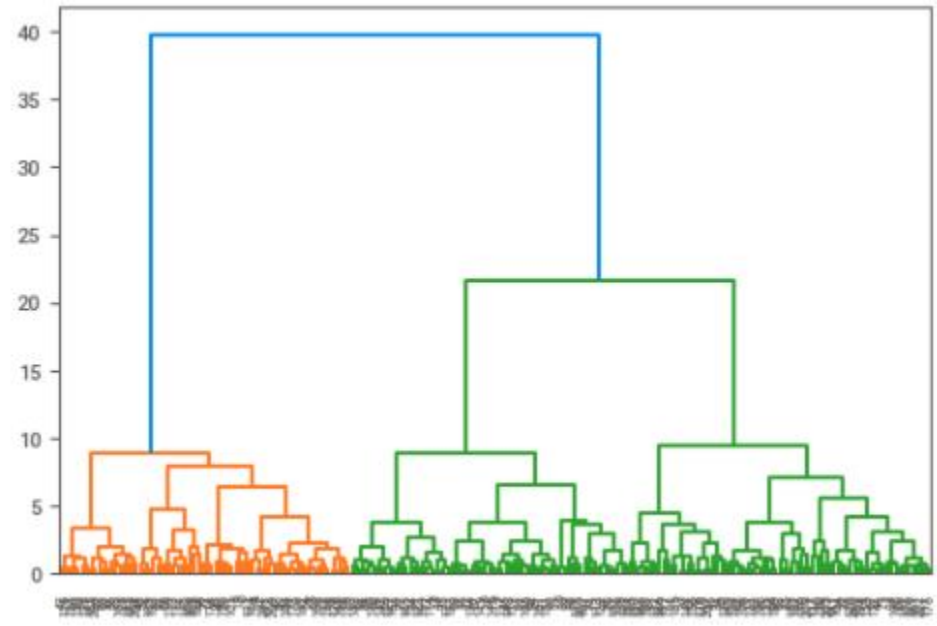
The data set contains different range of values. Clustering uses some sort of distance measure (ex: Euclidean distance) to determine if data belong to particular class. So, if there is a difference in range of values of data between variables It will affect the clustering determination as Higher weightage variable may get more preference. So scaling is required in clustering. In this data also we need to do clustering cos there are difference in range of values between columns. For example, spending mean is 14.8 whereas probability of full payment mean is 0.8709

Summary of data after scaling

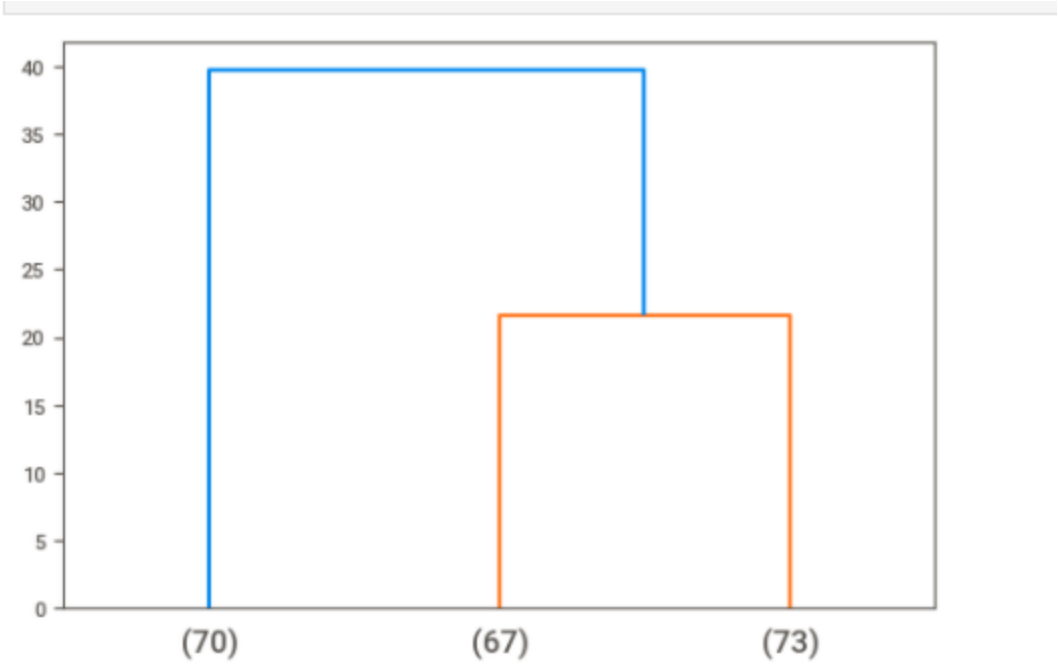
	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
count	2.100000e+02	2.100000e+02	2.100000e+02	2.100000e+02	2.100000e+02	2.100000e+02	2.100000e+02
mean	9.148766e-16	1.097006e-16	1.260896e-15	-1.358702e-16	-2.790757e-16	5.418946e-16	-1.935489e-15
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

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

Default Dendrogram without any optimization



Dendrogram with optimum cluster (Only by using dendrogram distance)



Insights

The optimum clusters are chosen based on the Maximum distance between the vertical segments of the dendrogram). Two clusters are formed

Head of dataset with cluster.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Clus_kmeans	cluster
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	2	1
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	0	2
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	2	1
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	1	2
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837	2	1
5	12.70	13.41	0.8874	5.183	3.091	8.456	5.000	1	2
6	12.02	13.33	0.8503	5.350	2.810	4.271	5.308	1	2
7	13.74	14.05	0.8744	5.482	3.114	2.932	4.825	0	2
8	18.17	16.26	0.8637	6.271	3.512	2.853	6.273	2	1
9	11.23	12.88	0.8511	5.140	2.795	4.325	5.003	1	2

Summary of cluster grouped dataset (Describe)

cluster		1	2
spending	count	70.000000	140.000000
	mean	18.371429	13.085571
	std	1.381233	1.550003
	min	15.380000	10.590000
	25%	17.330000	11.817500
	50%	18.720000	12.770000
	75%	19.137500	14.347500
	max	21.180000	16.630000
advance_payments	count	70.000000	140.000000
	mean	16.145429	13.766214
	std	0.599277	0.696916
	min	14.860000	12.410000
	25%	15.737500	13.207500
	50%	16.210000	13.665000
	75%	16.557500	14.305000
	max	17.250000	15.460000
probability_of_full_payment	count	70.000000	140.000000
	mean	0.884400	0.864298
	std	0.014767	0.024405
	min	0.845200	0.808100
	25%	0.874700	0.848075
	50%	0.883950	0.865800
	75%	0.898225	0.882075
	max	0.910800	0.918300
current_balance	count	70.000000	140.000000
	mean	6.158171	5.363714
	std	0.245926	0.230740
	min	5.709000	4.899000
	25%	5.979250	5.179000
	50%	6.148500	5.351000
	75%	6.312000	5.521750
	max	6.675000	6.053000
credit_limit	count	70.000000	140.000000
	mean	3.684629	3.045593
	std	0.174909	0.249454
	min	3.268000	2.630000
	25%	3.554250	2.835250
	50%	3.693500	3.037000
	75%	3.804750	3.234500
	max	4.033000	3.582000
min_payment_amt	count	70.000000	140.000000
	mean	3.639157	3.730723
	std	1.208271	1.634514
	min	1.472000	0.765100
	25%	2.845500	2.461750
	50%	3.629000	3.597500
	75%	4.459250	4.879250
	max	6.682000	8.456000
max_spent_in_single_shopping	count	70.000000	140.000000
	mean	6.017371	5.103421
	std	0.251132	0.226834
	min	5.443000	4.519000
	25%	5.877000	5.000000
	50%	5.981500	5.091500
	75%	6.187750	5.222500
	max	6.550000	5.879000

Insights(describe):

The average spending of cluster 1 is 18300, cluster 2 is 13000
The average amount of advance payment of cluster 1 is 1610, cluster 2 is 1370
The average probability of full payment of cluster 1 is 88% cluster 2 is 86 %
The average current balance of cluster 1 is 6100, cluster 2 is 5300
The average credit limit of cluster 1 is 36000, cluster 2 is 30000

The average amount of minimum payment amount of cluster 1 is 3600 and cluster 2 is 3700
The average amount of maximum spent in single shopping is 6000, cluster 2 is 5100

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

INERTIA (N_Clusters =2)

```
659.171754487041
```

Inertia (N_Clusters = 3)

```
430.6589731513006
```

Inertia (N_Clusters = 4)

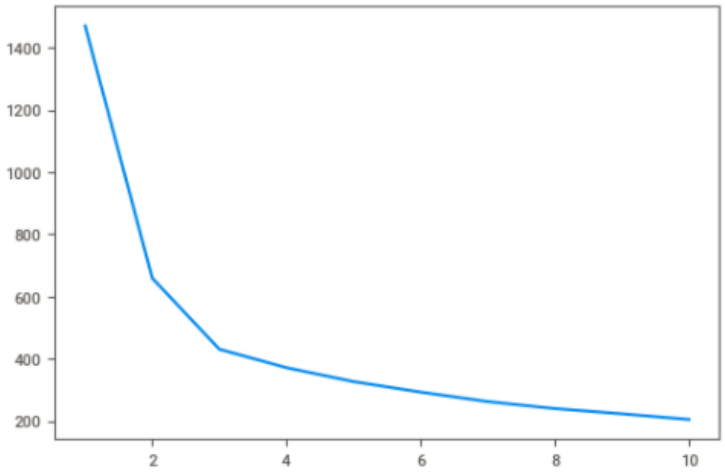
```
371.65314399951626
```

Inertia (**N_Clusters** = 5)

```
326.22891682972653
```

So the optimum cluster is 3

Elbow curve (n cluster range between 1 to 11)



Head of data set with Kmeans cluster (Number of clusters = 3)

[117]:

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

silhouette score

: 0.4007270552751299

Head of dataset with silhouette samples and kmeans cluster

]:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Clus_kmeans	sil_width
0	19.94	16.92	0.8752	6.675	3.763	3.252	6.550	1	0.573699
1	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	2	0.366386
2	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	1	0.637784
3	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	0	0.512458
4	17.99	15.86	0.8992	5.890	3.694	2.068	5.837	1	0.362276

Insights

Using the Elbow curve we conclude that the Optimal number of cluster using Kmeans clustering is 3 .

If we choose more than three clusters there is no vast changes in within-cluster sum-of-squares or inertia . Ie The feature difference between the clusters will less and so the model accuracy will be affected .

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

Cluster profile :

Clus_kmeans		0	1	2
spending	count	72.000000	67.000000	71.000000
	mean	11.856944	18.495373	14.437887
	std	0.714801	1.277122	1.056513
	min	10.590000	15.560000	12.080000
	25%	11.255000	17.590000	13.820000
	50%	11.825000	18.750000	14.430000
	75%	12.395000	19.145000	15.260000
	max	13.340000	21.180000	16.440000
advance_payments	count	72.000000	67.000000	71.000000
	mean	13.247778	16.203433	14.337746
	std	0.355208	0.546439	0.525706
	min	12.410000	14.890000	13.150000
	25%	12.992500	15.855000	14.030000
	50%	13.250000	16.230000	14.390000
	75%	13.482500	16.580000	14.760000
	max	13.950000	17.250000	15.270000
probability_of_full_payment	count	72.000000	67.000000	71.000000
	mean	0.848253	0.884210	0.881597
	std	0.019953	0.014917	0.015502
	min	0.808100	0.845200	0.852700
	25%	0.835000	0.874650	0.871300
	50%	0.848600	0.882900	0.881900
	75%	0.861475	0.898050	0.893350
	max	0.888300	0.910800	0.918300
current_balance	count	72.000000	67.000000	71.000000
	mean	5.231750	6.175687	5.514577
	std	0.141795	0.237807	0.225266
	min	4.899000	5.718000	4.984000
	25%	5.139250	6.011500	5.380000
	50%	5.225000	6.153000	5.541000
	75%	5.337250	6.328000	5.689500
	max	5.541000	6.675000	5.920000
credit_limit	count	72.000000	67.000000	71.000000
	mean	2.849542	3.697537	3.259225
	std	0.138689	0.166014	0.154766
	min	2.630000	3.387000	2.936000
	25%	2.738500	3.564500	3.155000
	50%	2.836500	3.719000	3.258000
	75%	2.967000	3.808000	3.378000
	max	3.232000	4.033000	3.582000
min_payment_amt	count	72.000000	67.000000	71.000000
	mean	4.742389	3.632373	2.707341
	std	1.354711	1.211052	1.176440
	min	1.502000	1.472000	0.765100
	25%	4.032250	2.848000	1.951000
	50%	4.799000	3.619000	2.640000
	75%	5.463750	4.421000	3.332000
	max	8.456000	6.682000	6.685000
max_spent_in_single_shopping	count	72.000000	67.000000	71.000000
	mean	5.101722	6.041701	5.120803
	std	0.184012	0.229566	0.269558
	min	4.519000	5.484000	4.605000
	25%	5.001000	5.879000	4.958500
	50%	5.089000	6.009000	5.132000
	75%	5.223500	6.192500	5.263500
	max	5.491000	6.550000	5.879000

Cluster profile Insights (average of cluster profile):

The average spending of cluster 0 is 11000, cluster 1 is 18000 and cluster 2 is 14000
The average amount of advance payment of cluster 0 is 1300, cluster 1 is 1600 and cluster 2 is 1400
The average probability of full payment of cluster 0 is 84%, cluster 1 is 88 % and cluster 2 is 88 %
The average current balance of cluster 0 is 5200, cluster 1 is 6100 and cluster 2 is 5500
The average credit limit of cluster 0 is 28000, cluster 1 is 36000 and cluster 2 is 32000

The average amount of minimum payment amount of cluster 0 is 4700, cluster 1 is 3600 and cluster 2 is 2700
The average amount of maximum spent in single shopping for cluster 0 is 5100, cluster 1 is 6000, cluster 2 is 5100

Recommendations

For cluster 0 customers: Their credit limit, advance payment and their probability full payment is less compared to other clusters So to improve their probability of full payment as well to increase the advance payment , I will recommend to give gift vouchers for the cluster 0 if they are willing to make more advance payment . This will reduce the risk of not paying (i.e. increase the probability of full payment) .

For cluster 1: They have the maximum credit limits, good full payment record (average) as well maximum spending power compared to other clusters . So, if we provide rewards point multiplier or more discounts in partner sites such as ecommerce or retailers, it will motivate the cluster 1 customers to spend more i.e. in turn it will increase the business for the lending bank.

For cluster 2: They have good full payment history (average) , median credit limit with median values for most of the other variables . Recommendation is to give discounts if they pay more advance amount as well as to provide reward points multiplier / discounts in partner sites to increase their spending amount .

Problem 2: CART-RF-ANN

Business scenario

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

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it.
2.2 Data Split: Split the data into test and train, build classification model CART, Random Forest, Artificial Neural Network
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
2.4 Final Model: Compare all the model and write an inference which model is best/optimized.
2.5 Inference: Basis on these predictions, what are the business insights and recommendations

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

a.) Dataset Head

	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

b.) Summary of the dataset:

]:

	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

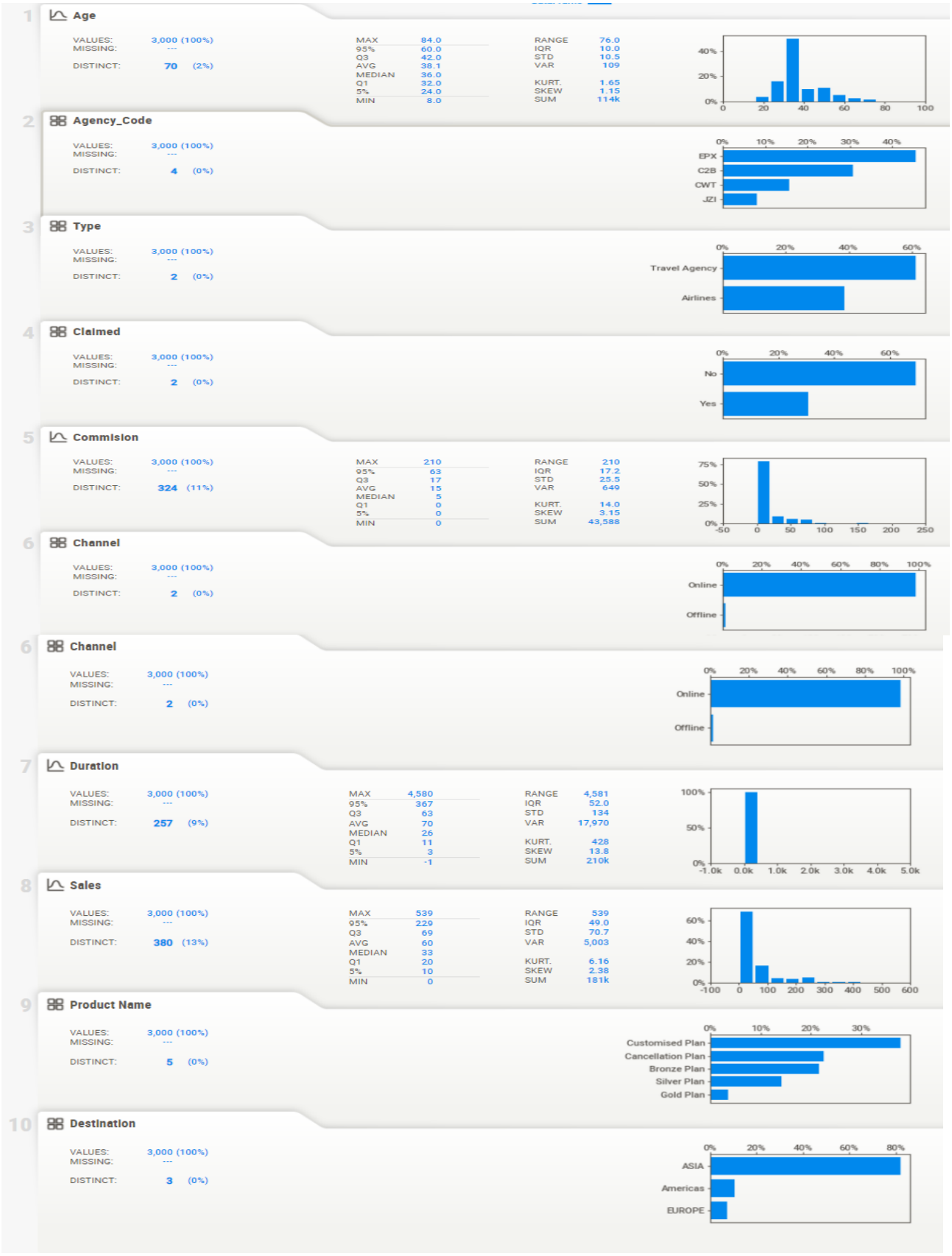
c.) Type of the variables in dataset

```
<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
```

d.) Dataset has any null values.

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

e.) EDA using sweet viz to visualize the summary for each variable as well to underrated the data



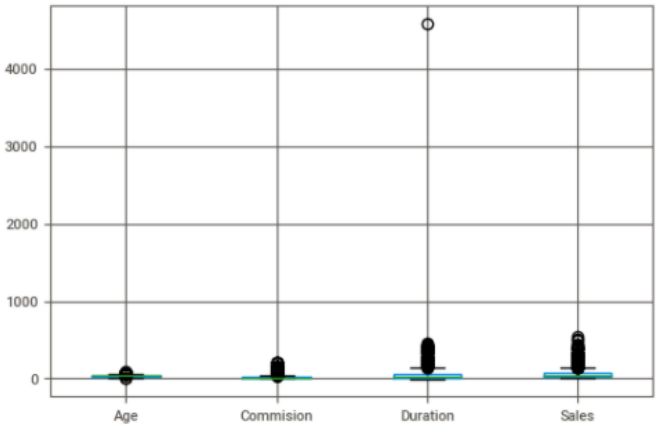
f.) Check for duplicates.

```
Number of Duplicates  139
```

Data after removing duplicates (Kept the last updated duplicate value)

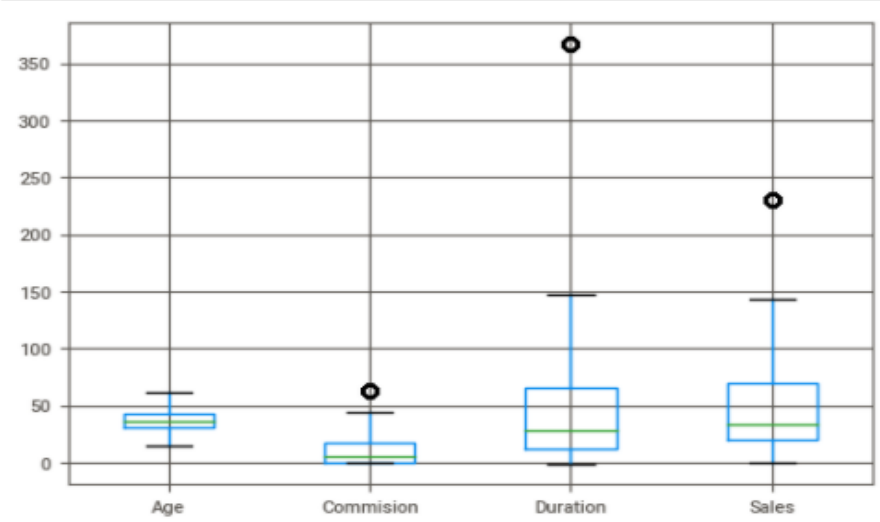
```
Removing duplicates
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2861 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Age           2861 non-null   int64
 1   Agency_Code   2861 non-null   object
 2   Type          2861 non-null   object
 3   Claimed       2861 non-null   object
 4   Commision     2861 non-null   float64
 5   Channel       2861 non-null   object
 6   Duration      2861 non-null   int64
 7   Sales         2861 non-null   float64
 8   Product Name  2861 non-null   object
 9   Destination   2861 non-null   object
dtypes: float64(2), int64(2), object(6)
memory usage: 245.9+ KB
```

g) Check for outliers




```
the number of outliers are  290
```

h) Boxplot of the dataset after handling Outliers



Inference:

Insurance Data set had duplicates and outliers. Have removed the duplicates and handled the outliers.
Dataset has no null values.
There is difference in scaling of data between the variables. So, scaling maybe required for some models.
There is a high categorical correlation between Agency code, Type, commission and product Name variables
There is a high numerical correlation between Commission and Sales variables.

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

Data split

Head of dataset after converting the object variables into categorical codes .

]:

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48.0	0	0	0	0.70	1	7.0	2.51	2	0
2	39.0	1	1	0	5.94	1	3.0	9.90	2	1
3	36.0	2	1	0	0.00	1	4.0	26.00	1	0
4	33.0	3	0	0	6.30	1	53.0	18.00	0	0
5	45.0	3	0	1	15.75	1	8.0	45.00	0	0

Head Independent variables (Extracted from the above dataset):

:

	Age	Agency_Code	Type	Commision	Channel	Duration	Sales	Product Name	Destination
0	48.0	0	0	0.70	1	7.0	2.51	2	0
1	36.0	2	1	0.00	1	34.0	20.00	2	0
2	39.0	1	1	5.94	1	3.0	9.90	2	1
3	36.0	2	1	0.00	1	4.0	26.00	1	0
4	33.0	3	0	6.30	1	53.0	18.00	0	0

Head of Dependent variables(Claimed column)

3]:

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

Head of Train data after splitting. (Independent variables)

	Age	Agency_Code	Type	Commision	Channel	Duration	Sales	Product Name	Destination
1045	36.0	2	1	0.00	1	30.0	20.000	2	0
2717	36.0	2	1	0.00	1	139.0	42.000	2	1
2835	28.0	0	0	63.21	1	367.0	228.565	4	0
2913	28.0	0	0	12.13	1	29.0	48.500	4	0
959	48.0	1	1	18.62	1	53.0	49.000	3	0

Head of Test data after splitting. (Independent variables)

	Age	Agency_Code	Type	Commision	Channel	Duration	Sales	Product Name	Destination
1957	22.0	1	1	28.50	1	28.0	75.0	0	2
2087	55.0	0	0	6.63	1	24.0	26.5	0	0
1394	29.0	0	0	4.00	1	33.0	16.0	0	0
1520	27.0	0	0	15.88	1	40.0	63.5	4	0
1098	36.0	2	1	0.00	1	35.0	27.0	1	0

Head of Train Labels (dependent variable Claimed column)

```
1045    0
2717    0
2835    1
2913    1
959     0
Name: Claimed, dtype: int8
```

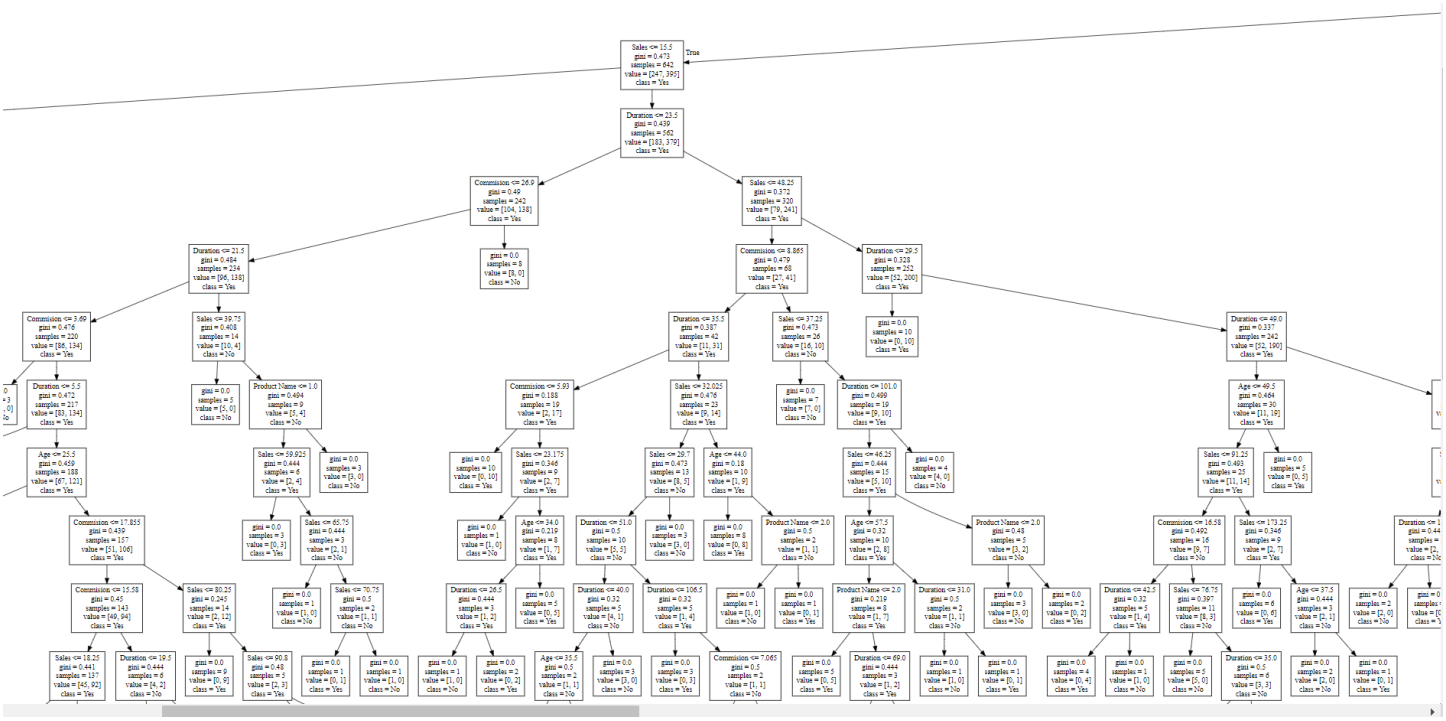
Head of Test Labels (dependent variable Claimed column)

```
1957    0
2087    1
1394    1
1520    1
1098    0
Name: Claimed, dtype: int8
```

The insurance dataset is split in the ratio of 70:30. i.e. 70 percent of data for training and 30 percent of data for testing. We used random state 1 to make sure the split data value remains the same even if we run the code many times as long there is no change the in original csv data.

a) CART Decision Tree

Decision tree without any optimization



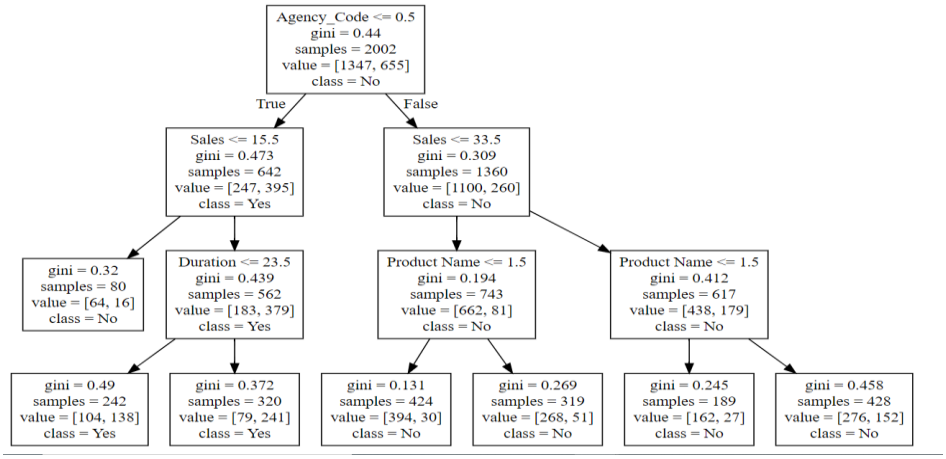
Optimization metrics :

```
GridSearchCV(cv=4, estimator=DecisionTreeClassifier(),
             param_grid={'max_depth': [2, 3, 4, 5],
                         'min_samples_leaf': [20, 25, 35],
                         'min_samples_split': [80, 100, 120]})
```

Decision Tree with best Optimized parameters :

```
DecisionTreeClassifier(max_depth=3, min_samples_leaf=20, min_samples_split=100)
```

Optimized Decision Tree :



b.) Random Forest

Different Optimization metrics :

```
GridSearchCV(estimator=RandomForestClassifier(),
             param_grid={'max_depth': [2, 5, 6, 7],
                         'min_samples_leaf': [20, 25, 30],
                         'min_samples_split': [80, 100, 120],
                         'n_estimators': [101, 201]})
```

Random forest classification is built with best Optimization:

```
]: RandomForestClassifier(max_depth=5, min_samples_leaf=25, min_samples_split=80,
                          n_estimators=101)
```

c.) MLP Classifier (Artificial Neural Network)

Different optimization metrics:

```
GridSearchCV(cv=3, estimator=MLPClassifier(),
             param_grid={'activation': ['logistic', 'relu'],
                         'hidden_layer_sizes': [(100, 100, 100),
                                                (200, 200, 200),
                                                (300, 300, 300)],
                         'max_iter': [10000, 5000], 'solver': ['sgd', 'adam'],
                         'tol': [0.1, 0.01]})
```

Best Optimized Parameters:

```
{'activation': 'relu',
 'hidden_layer_sizes': (200, 200, 200),
 'max_iter': 10000,
 'solver': 'adam',
 'tol': 0.1}
```

Artificial Neural Network with best optimization:

```
MLPClassifier(hidden_layer_sizes=(200, 200, 200), max_iter=10000, tol=0.1)
```

Insights:

Data is split into train and test and three different models namely Decision tree , Random forest and Artificial Neural Network are built .

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

a) Performance Metrics for CART Decision Tree

Accuracy score for Training data

```
0.7707292707292708
```

Accuracy score for Testing data

```
0.7671711292200233
```

Confusion Matrix For Training set

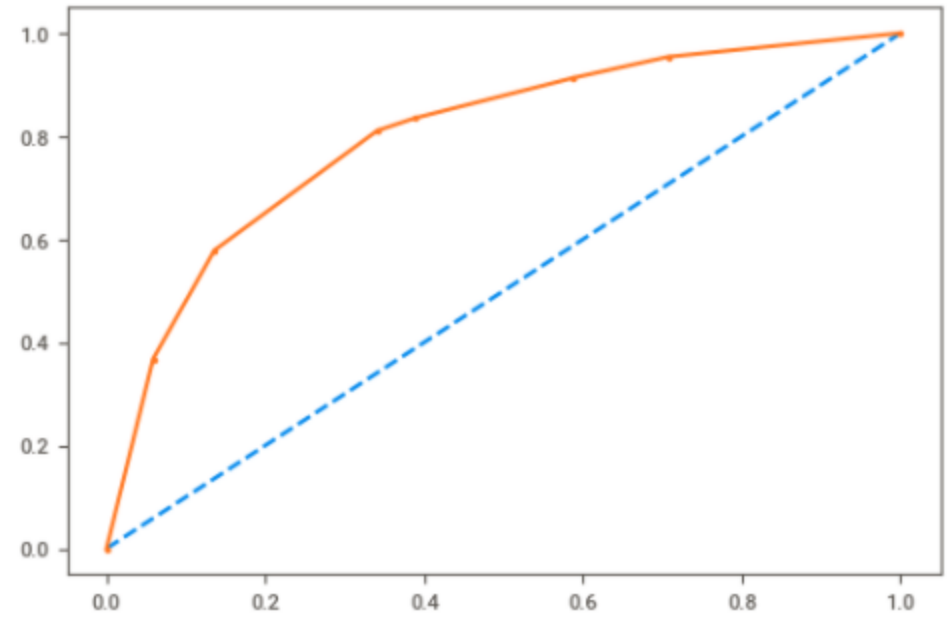
```
array([[1164,  183],
       [ 276,  379]], dtype=int64)
```

Confusion Matrix For Testing set

```
array([[503,  97],
       [103, 156]], dtype=int64)
```

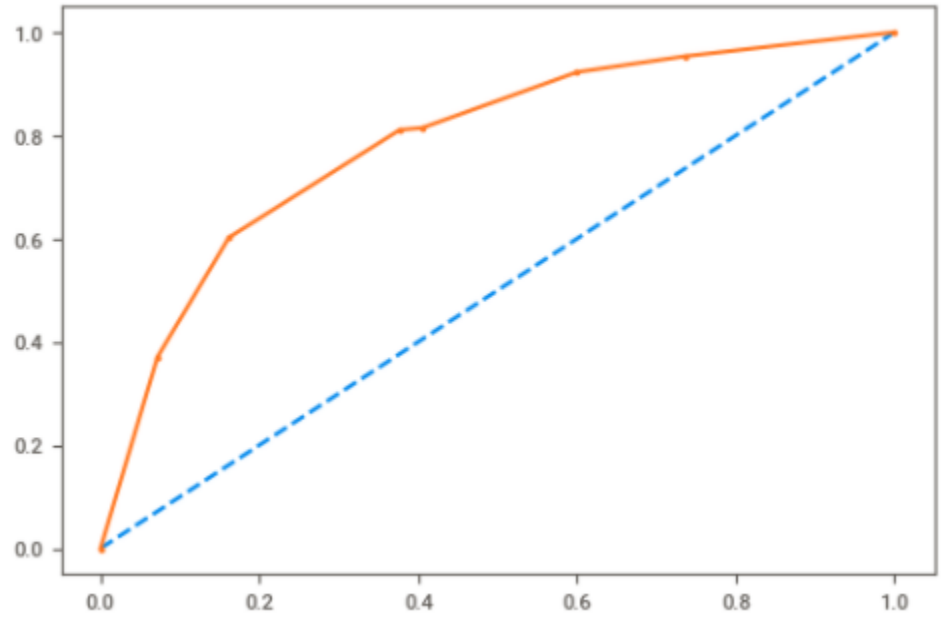
ROC curve and ROC_AUC score Training set

```
AUC: 0.8007.
```



ROC curve and ROC_AUC score Testing set

```
AUC: 0.7869
```



b) Performance Metrics for Random Forest

Accuracy score for Training data

```
0.7867132867132867
```

Accuracy score for Testing data

```
0.7753201396973225
```

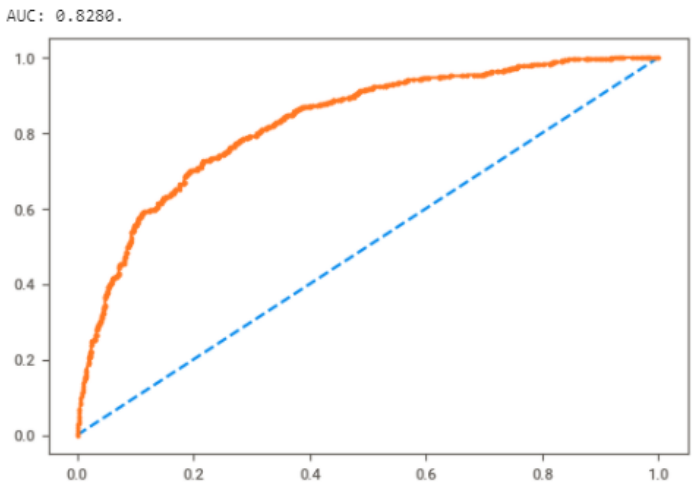
Confusion Matrix For Training set

```
array([[1187,  160],
       [ 267,  388]], dtype=int64)
```

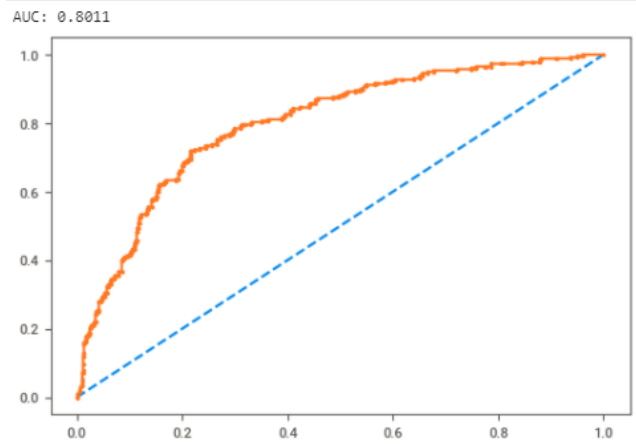
Confusion Matrix For Testing set

```
array([[505,  95],
       [ 98, 161]], dtype=int64)
```

ROC curve and ROC_AUC score Training set



ROC curve and ROC_AUC score Testing set



c) Performance Metrics for Artificial Neural Network

Accuracy score for Training data

```
0.7697302697302697
```

Accuracy score for Testing data

```
: 0.7718277066356228
```

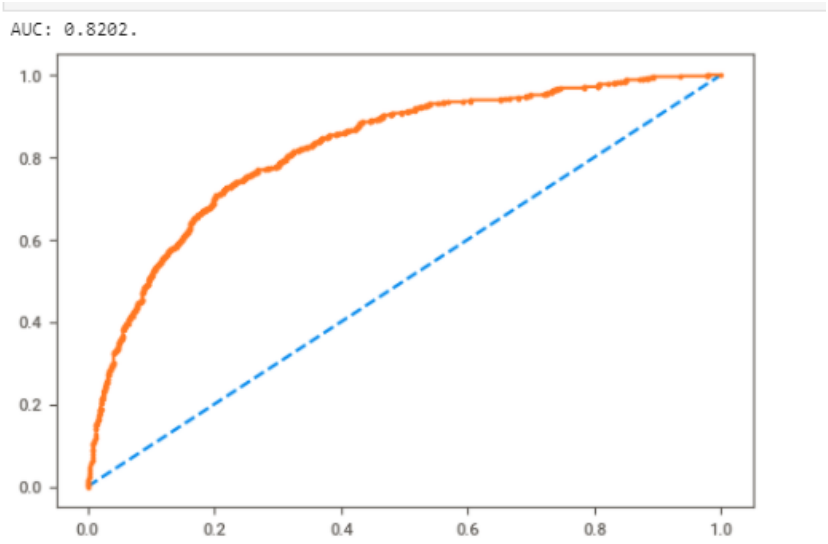
Confusion Matrix For Training set

```
array([[1144,  203],
       [ 258,  397]], dtype=int64)
```

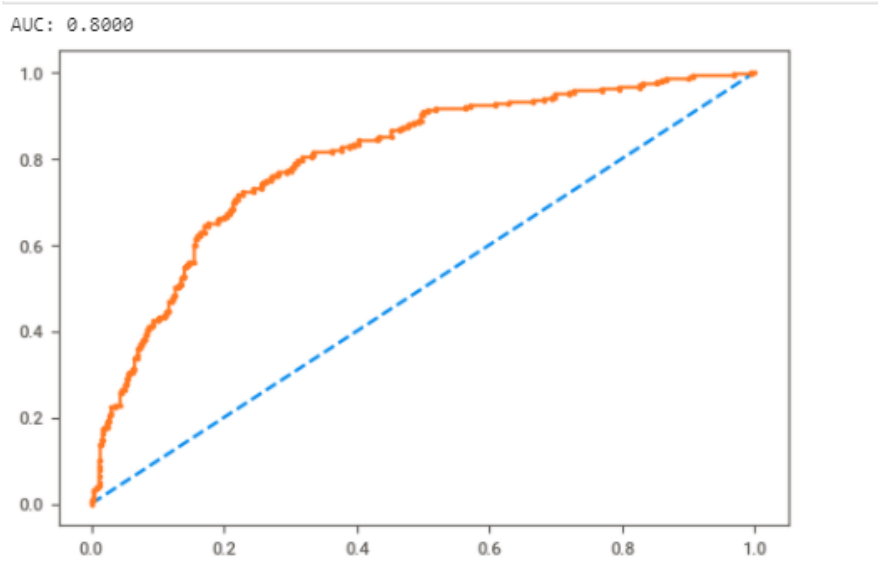
Confusion Matrix For Testing set

```
array([[498, 102],
       [ 94, 165]], dtype=int64)
```

ROC curve and ROC_AUC score Training set



ROC curve and ROC_AUC score Testing set



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

a.)CART decision Tree Performance metrics

Training Set

	precision	recall	f1-score	support
0	0.81	0.86	0.84	1347
1	0.67	0.58	0.62	655
accuracy			0.77	2002
macro avg	0.74	0.72	0.73	2002
weighted avg	0.76	0.77	0.77	2002

Testing Set

	precision	recall	f1-score	support
0	0.83	0.84	0.83	600
1	0.62	0.60	0.61	259
accuracy			0.77	859
macro avg	0.72	0.72	0.72	859
weighted avg	0.77	0.77	0.77	859

b.) Random Forest Classifier Performance Metrics

Training Set

	precision	recall	f1-score	support
0	0.82	0.88	0.85	1347
1	0.71	0.59	0.65	655
accuracy			0.79	2002
macro avg	0.76	0.74	0.75	2002
weighted avg	0.78	0.79	0.78	2002

Testing Set

	precision	recall	f1-score	support
0	0.84	0.84	0.84	600
1	0.63	0.62	0.63	259
accuracy			0.78	859
macro avg	0.73	0.73	0.73	859
weighted avg	0.77	0.78	0.77	859

c.) Artificial Neural Network performance metrics

Training Set

	precision	recall	f1-score	support
0	0.82	0.85	0.83	1347
1	0.66	0.61	0.63	655
accuracy			0.77	2002
macro avg	0.74	0.73	0.73	2002
weighted avg	0.77	0.77	0.77	2002

Testing Set

	precision	recall	f1-score	support
0	0.84	0.83	0.84	600
1	0.62	0.64	0.63	259
accuracy			0.77	859
macro avg	0.73	0.73	0.73	859
weighted avg	0.77	0.77	0.77	859

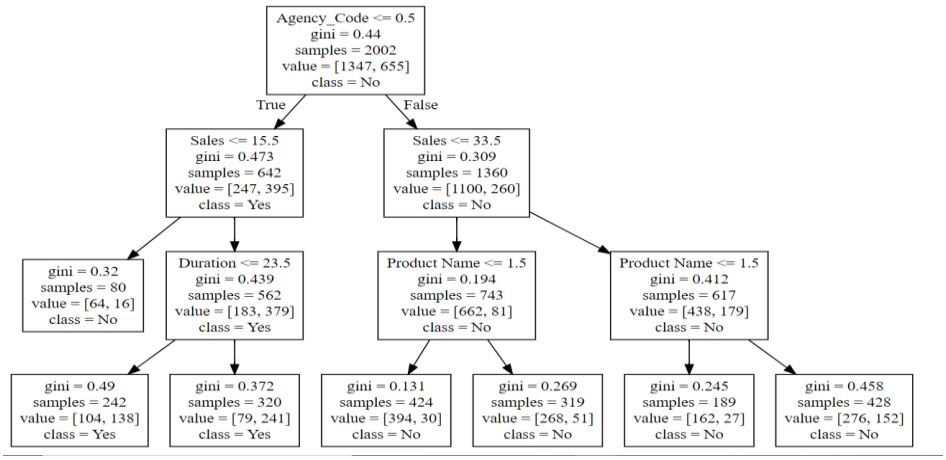
Inference:

The Artificial Neural Network is the best optimized model.
Even though the accuracy score is little less compared to the random forest, Artificial Neural network has the lease difference between the training and testing sets in almost all performance metrics such as accuracy, precision, recall and f1 score.

2.5 Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective.

I will recommend the insurance firm to use Artificial Neural network model to decide whether approve the claim or not. The model will be able to predict the claim status with 77 percent accuracy for both known data fields as well for unknown data.

To do it manually or to increase the accuracy the firm may use the below decision tree along with the ANN model to make the decision .



.