PGPDSBA Online FEB A 2021

greatlearning
Power Ahead

Pavan Kumar R Naik

PGP-DSBA Online

Feb A 2021

27/06/2021

Table of Contents

List of Figures

List of Tables

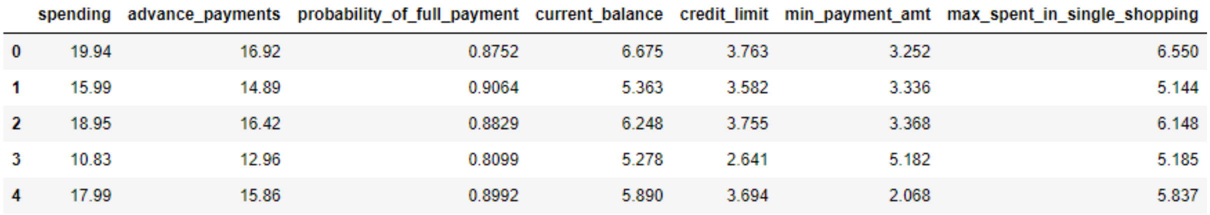
Problem 1: Clustering:

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.

Q1.1. Read the data, do the necessary initial steps, and exploratory data analysis.

Solution:

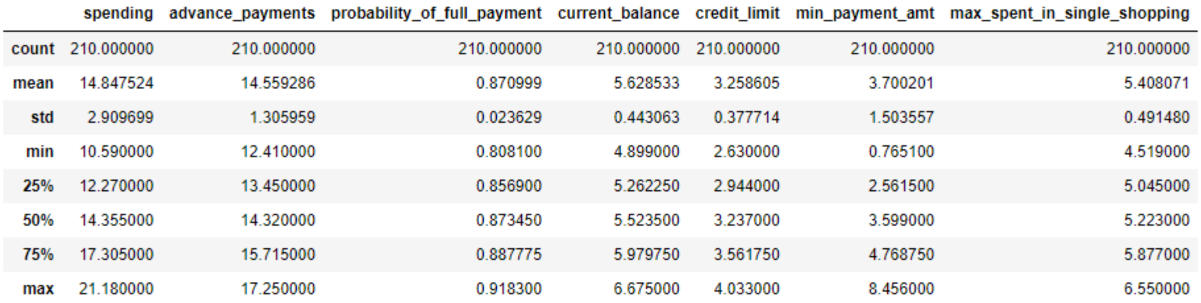
Sample of Dataset:



	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	6.550
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

Table 1. Dataset Sample (Bank Marketing)

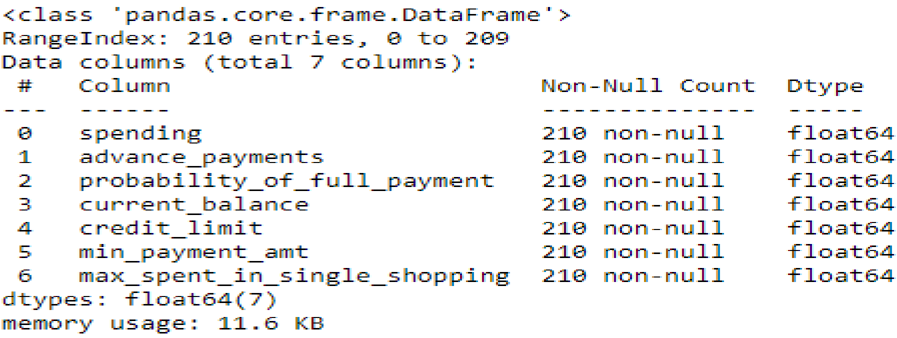
Summary of Dataset:



	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

Table 2. Dataset Summary (Bank Marketing)

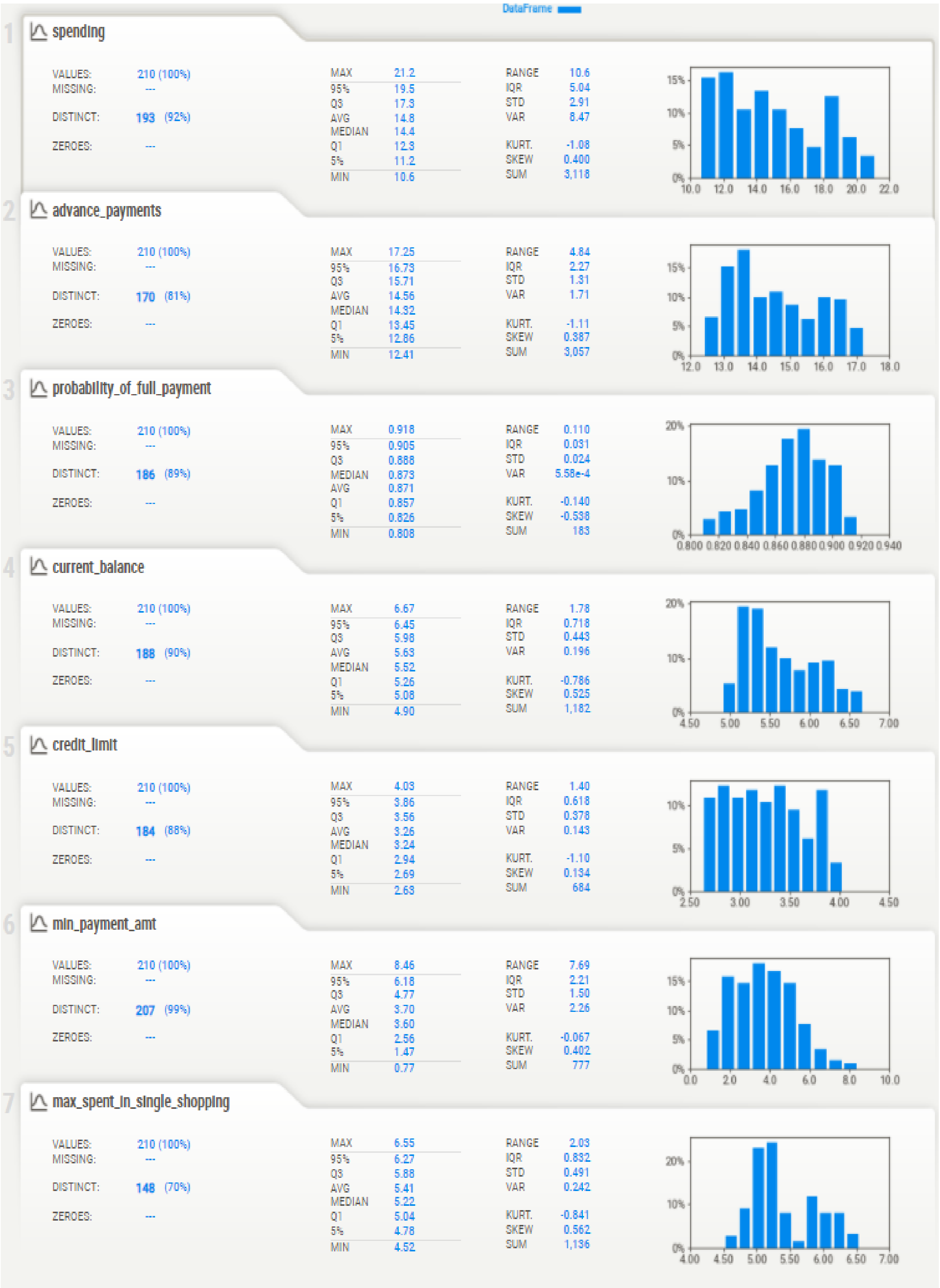
Type of Variables:



```
<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
```

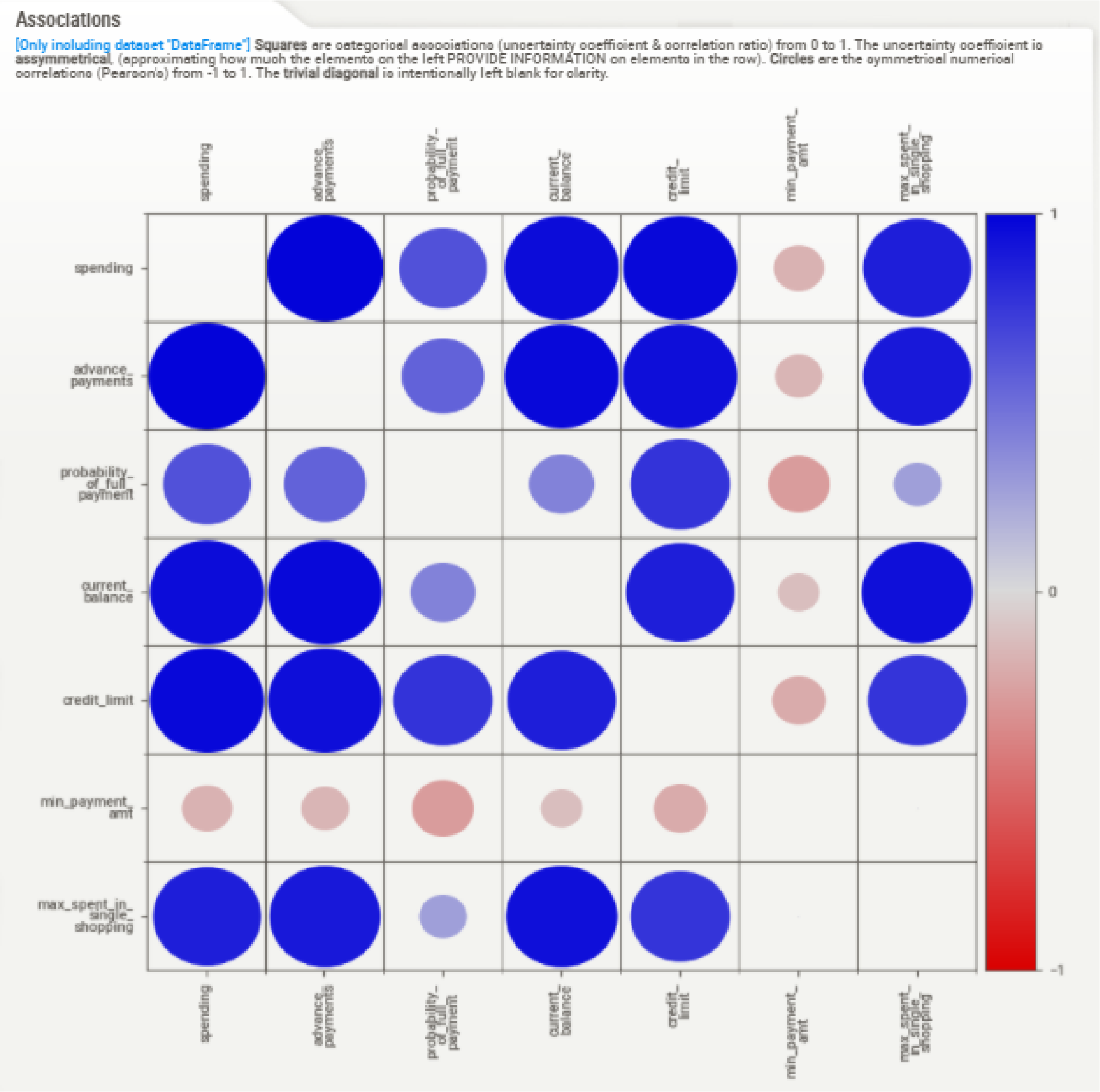
Table 3. Type of Variables (Bank Marketing)

Data Visualization: EDA using sweet viz to visualize the summary for each variable as well to underrated data –



```
1 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
ZEROES:    ---           Q1      12.8    KURT.  -1.08
                         5%      11.2    SKEW   0.400
                         MIN     10.6    SUM    3,118

2 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
ZEROES:    ---           Q1      13.45   KURT.  -1.11
                         5%      12.86   SKEW   0.387
                         MIN     12.41   SUM    3,057

3 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    5.58e-4
                         AVG     0.871
ZEROES:    ---           Q1      0.857   KURT.  -0.140
                         5%      0.826   SKEW   -0.538
                         MIN     0.808   SUM    183

4 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
ZEROES:    ---           Q1      5.26    KURT.  -0.786
                         5%      5.08    SKEW   0.525
                         MIN     4.90    SUM    1,182

5 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
ZEROES:    ---           Q1      2.94    KURT.  -1.10
                         5%      2.69    SKEW   0.134
                         MIN     2.63    SUM    684

6 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
ZEROES:    ---           Q1      2.56    KURT.  -0.067
                         5%      1.47    SKEW   0.402
                         MIN     0.77    SUM    777

7 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
ZEROES:    ---           Q1      5.04    KURT.  -0.841
                         5%      4.78    SKEW   0.562
                         MIN     4.52    SUM    1,136
```

Fig 1.1. Sweet viz Univariate analysis



Associations

[Only including dataset "DataFrame"] **Squares** are categorical associations (uncertainty coefficient & correlation ratio) from 0 to 1. The uncertainty coefficient is **assymmetrical**, (approximating how much the elements on the left PROVIDE INFORMATION on elements in the row). **Circles** are the symmetrical numerical correlations (Pearson's) from -1 to 1. The **trivial diagonal** is intentionally left blank for clarity.

Fig 1.2. Sweet viz Multivariate analysis

Boxplot to check the outliers:

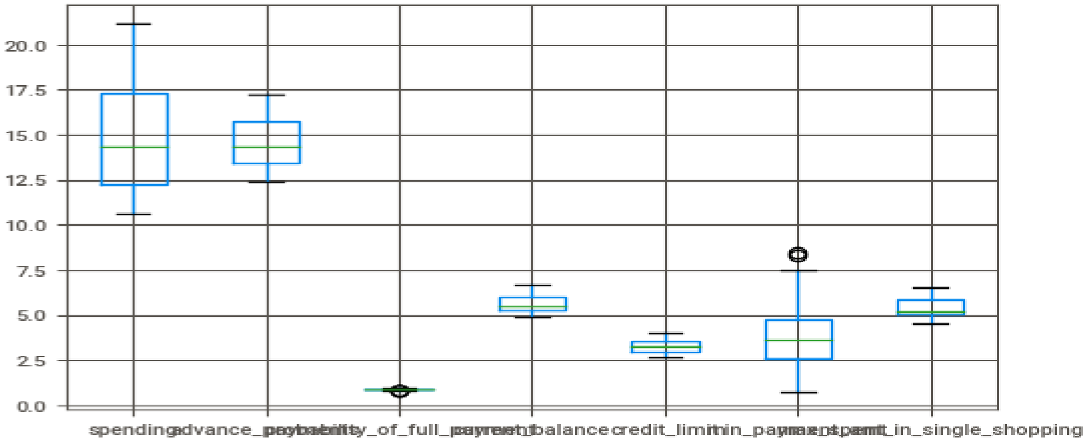


Fig 1.3. Boxplot (Bank Marketing)

Observation (EDA):

- There are 7 variables and 210 records
- No missing record based
- All the variables float data type
- There are no missing values
- There are no duplicate rows
- While comparing the max, min, avg and 5-point summary data appears not to have the outliers
- Data range differs from variable to variable. Ex. Spending column has data value range between 10 and 21 whereas min_payment_amt column has value range between 0.76 and 8.45
- Strong positive correlation between:
    i. spending & advance_payments
    ii. advance_payments & current_balance
    iii. credit_limit & spending
    iv. spending & current_balance
    v. credit_limit & advance_payments
    vi. max_spent_in_single_shopping current_balance

Q1.2. Do you think scaling is necessary for clustering in this case? Justify.

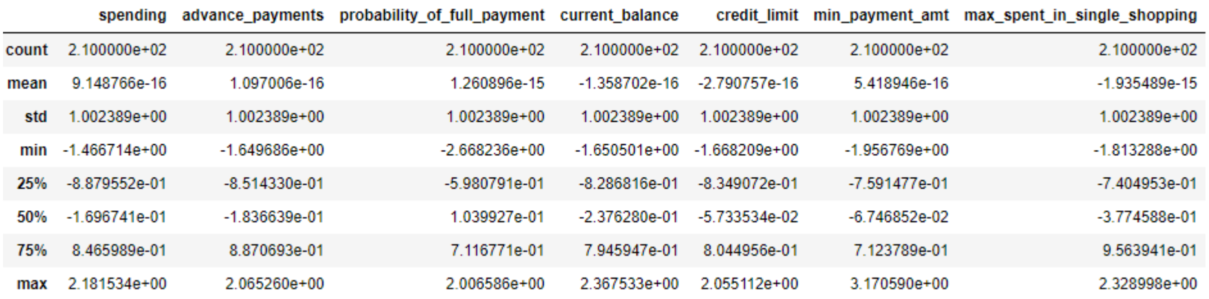
Solution:

Yes, The Scaling is required

The data set contains different range of values. Clustering uses sort of distance measure (ex: Euclidean distance) to determine if the 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. Hence scaling is required in clustering. In this data also we need to do clustering because there is difference in range of values between columns. For ex. spending mean is 14.8 whereas probability of full payment mean is 0.8709. Scaling needs to be done as the values of the variables are different.

I have used z score to standardize the data to relative same scale -3 to +3.

Summary of the Data post doing the 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

Table 4. Scaled Data Summary (Bank Marketing)

Q1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them

Solution:

Default Dendrogram without any optimization and using wardlink:

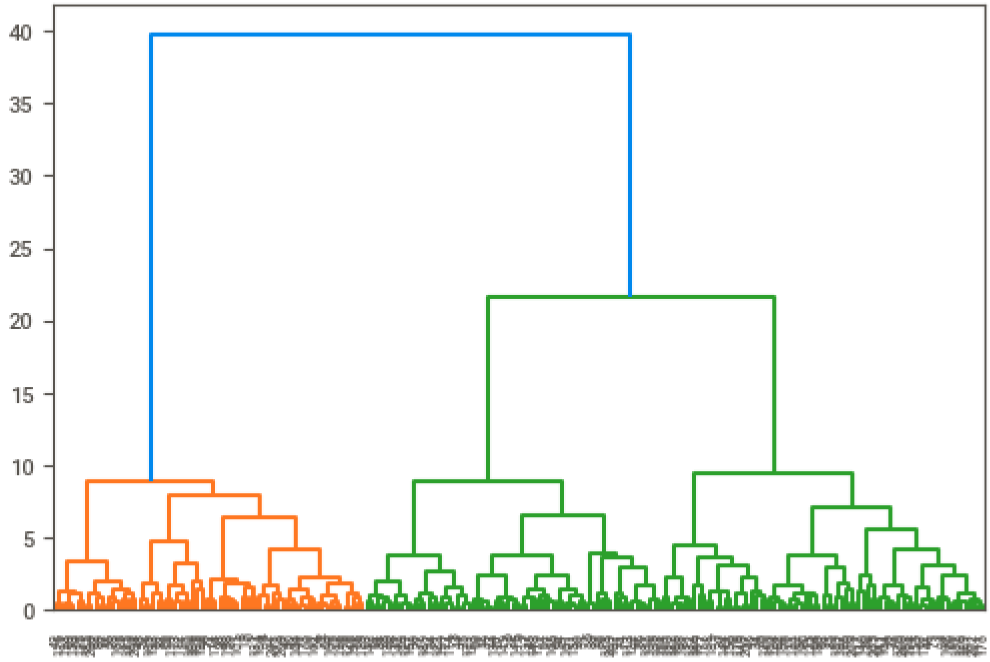


Fig 1.4. Default Dendrogram

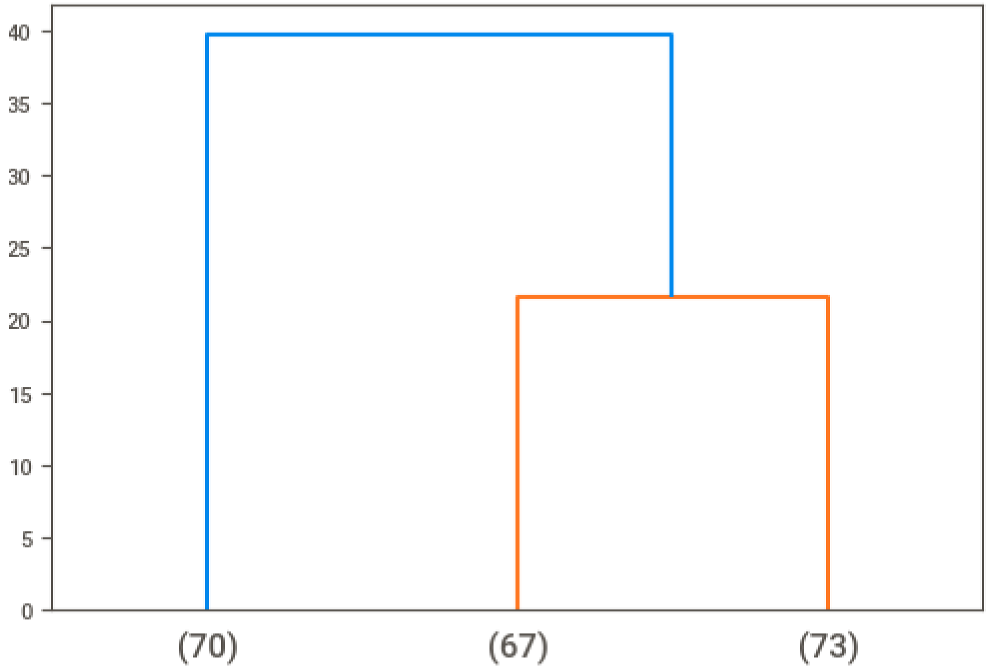
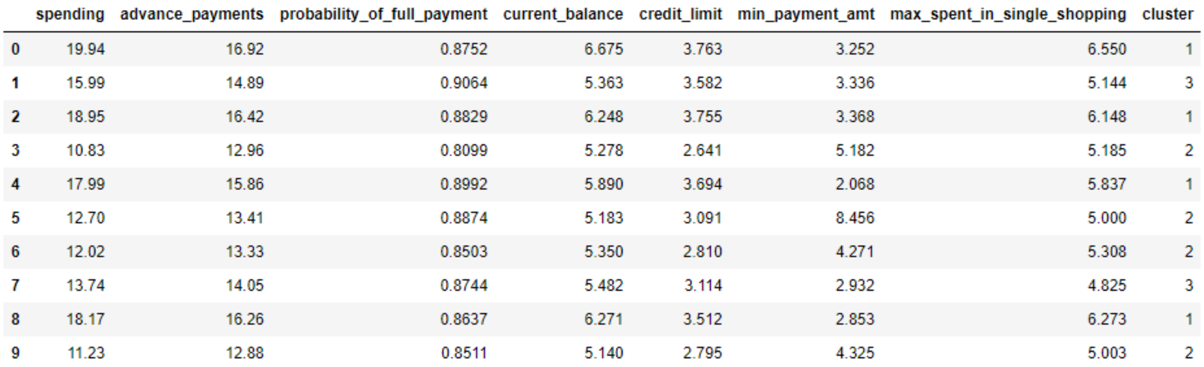


Fig 1.5. Dendrogram using Ward link

The optimum clusters are chosen based on the maximum distance between the vertical segments of the dendrogram.Three clusters are formed.

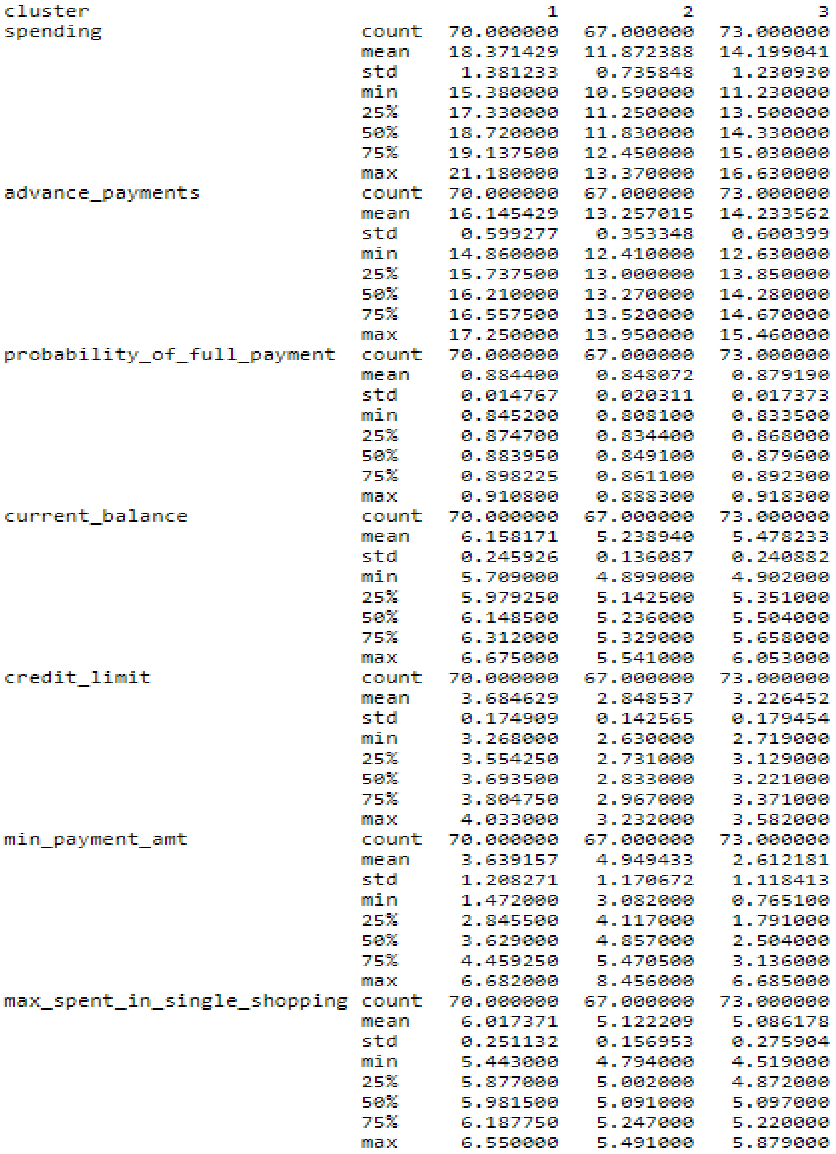
Head of the dataset with cluster:



	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	cluster
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	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	5.890	3.694	2.068	5.837	1
5	12.70	13.41	0.8874	5.183	3.091	8.456	5.000	2
6	12.02	13.33	0.8503	5.350	2.810	4.271	5.308	2
7	13.74	14.05	0.8744	5.482	3.114	2.932	4.825	3
8	18.17	16.26	0.8637	6.271	3.512	2.853	6.273	1
9	11.23	12.88	0.8511	5.140	2.795	4.325	5.003	2

Table 5. Dataset with cluster

Summary of the Cluster grouped Dataset:



```
cluster                                    1          2          3
spending                     count  70.000000  67.000000  73.000000
                             mean   18.371429  11.872388  14.199041
                             std     1.381233   0.735848   1.230930
                             min    15.380000  10.590000  11.230000
                             25%    17.330000  11.250000  13.500000
                             50%    18.720000  11.830000  14.330000
                             75%    19.137500  12.450000  15.030000
                             max    21.180000  13.370000  16.630000
advance_payments             count  70.000000  67.000000  73.000000
                             mean   16.145429  13.257015  14.233562
                             std     0.599277   0.353348   0.600399
                             min    14.860000  12.410000  12.630000
                             25%    15.737500  13.000000  13.850000
                             50%    16.210000  13.270000  14.280000
                             75%    16.557500  13.520000  14.670000
                             max    17.250000  13.950000  15.460000
probability_of_full_payment  count  70.000000  67.000000  73.000000
                             mean    0.884400   0.848072   0.879190
                             std     0.014767   0.020311   0.017373
                             min     0.845200   0.808100   0.833500
                             25%     0.874700   0.834400   0.868000
                             50%     0.883950   0.849100   0.879600
                             75%     0.898225   0.861100   0.892300
                             max     0.910800   0.888300   0.918300
current_balance              count  70.000000  67.000000  73.000000
                             mean    6.158171   5.238940   5.478233
                             std     0.245926   0.136087   0.240882
                             min     5.709000   4.899000   4.902000
                             25%     5.979250   5.142500   5.351000
                             50%     6.148500   5.236000   5.504000
                             75%     6.312000   5.329000   5.658000
                             max     6.675000   5.541000   6.053000
credit_limit                 count  70.000000  67.000000  73.000000
                             mean    3.684629   2.848537   3.226452
                             std     0.174909   0.142565   0.179454
                             min     3.268000   2.630000   2.719000
                             25%     3.554250   2.731000   3.129000
                             50%     3.693500   2.833000   3.221000
                             75%     3.804750   2.967000   3.371000
                             max     4.033000   3.232000   3.582000
min_payment_amt              count  70.000000  67.000000  73.000000
                             mean    3.639157   4.949433   2.612181
                             std     1.208271   1.170672   1.118413
                             min     1.472000   3.082000   0.765100
                             25%     2.845500   4.117000   1.791000
                             50%     3.629000   4.857000   2.504000
                             75%     4.459250   5.470500   3.136000
                             max     6.682000   8.456000   6.685000
max_spent_in_single_shopping count  70.000000  67.000000  73.000000
                             mean    6.017371   5.122209   5.086178
                             std     0.251132   0.156953   0.275904
                             min     5.443000   4.794000   4.519000
                             25%     5.877000   5.002000   4.872000
                             50%     5.981500   5.091000   5.097000
                             75%     6.187750   5.247000   5.220000
                             max     6.550000   5.491000   5.879000
```

Table 6. Summary with cluster

Observation:

- We for cluster grouping based on the dendrogram, 3 or 4 looks good. Did the further analysis, and based on the dataset had gone for 3 group cluster solution based on the hierarchical clustering
- Also in real time, there could have been more variables value captured - tenure, BALANCE_FREQUENCY, balance, purchase, instalments of purchase, others.
- And three group cluster solution gives a pattern based on high/medium/low spending with max_spent_in_single_shopping (high value item) and probability_of_full_payment (payment made)

Q1.4 Apply K-Means clustering on scaled data and determine optimum clusters. Apply elbow curve and silhouette score. Explain the results properly. Interpret and write inferences on the finalized clusters.

Solution:

K-Means inertia for Cluster 2 = 659.171

K-Means inertia for Cluster 3 = 430.658

K-Means inertia for Cluster 4 = 371.581

K-Means inertia for Cluster 5 = 326.306

Elbow curve range between clusters 1 to 10:

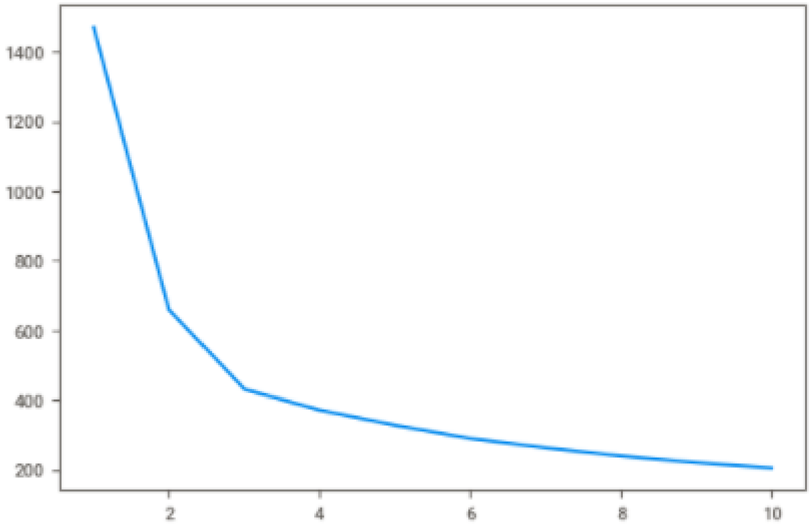
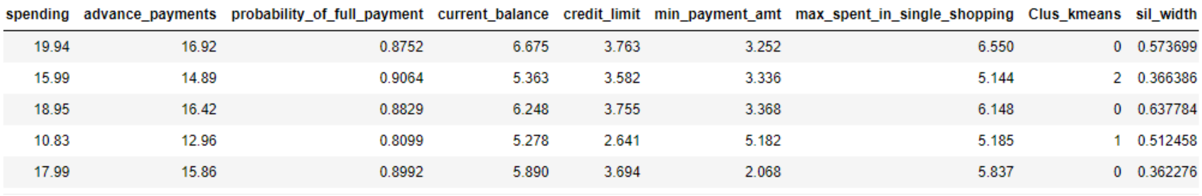


Fig 1.6. Elbow curve

Silhouette score = 0.40072

Head of dataset with Silhouette samples and K-Means cluster:



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

Table 7. Dataset head with Silhouette and K-Means

Insights:

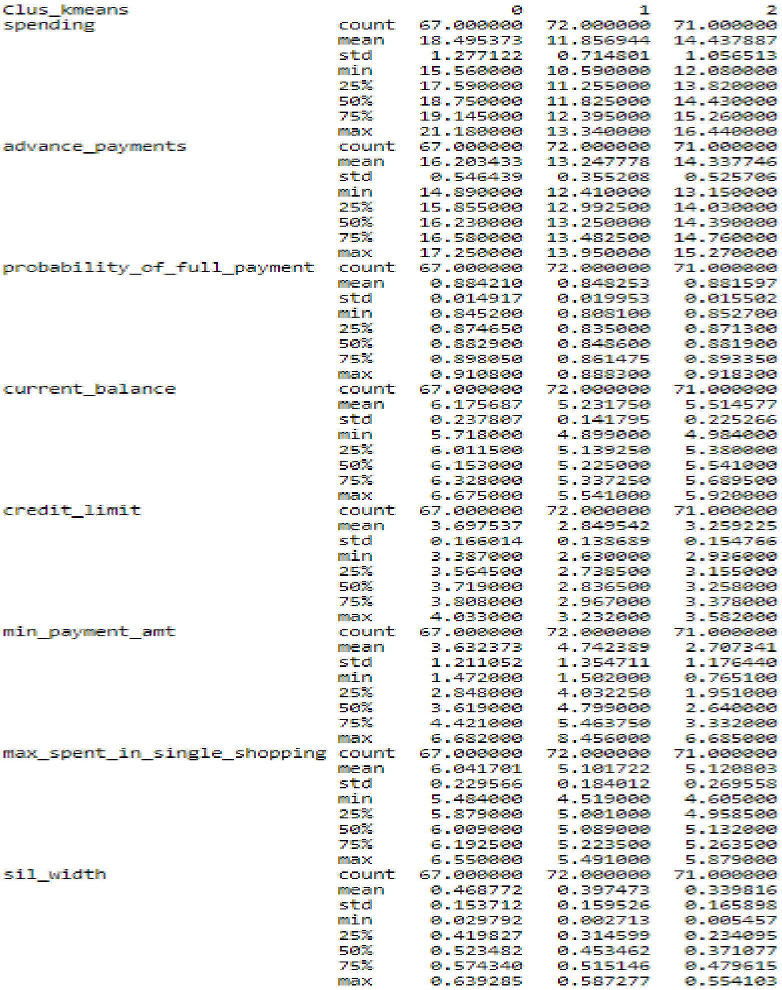
Using the elbow curve, we conclude that optimal number of clusters using K Means clustering is 3.

If we choose more than three clusters there is no vast changes within cluster sum of squares or inertia. i.e., the feature difference between clusters will be less and hence the model accuracy will be affected.

Q1.5 Describe cluster profiles for the clusters defined. Recommend different promotional strategies for different clusters.

Solution:

Cluster profile:



```
Clus_kmeans                                 0          1          2
spending                     count  67.000000  72.000000  71.000000
                             mean   18.495373  11.856944  14.437887
                             std     1.277122   0.714801   1.056513
                             min    15.560000  10.590000  12.080000
                             25%    17.590000  11.255000  13.820000
                             50%    18.750000  11.825000  14.430000
                             75%    19.145000  12.395000  15.260000
                             max    21.180000  13.340000  16.440000
advance_payments             count  67.000000  72.000000  71.000000
                             mean   16.203433  13.247778  14.337746
                             std     0.546439   0.355208   0.525706
                             min    14.890000  12.410000  13.150000
                             25%    15.855000  12.992500  14.030000
                             50%    16.230000  13.250000  14.390000
                             75%    16.580000  13.482500  14.760000
                             max    17.250000  13.950000  15.270000
probability_of_full_payment  count  67.000000  72.000000  71.000000
                             mean    0.884210   0.848253   0.881597
                             std     0.014917   0.019953   0.015502
                             min     0.845200   0.808100   0.852700
                             25%     0.874650   0.835000   0.871300
                             50%     0.882900   0.848600   0.881900
                             75%     0.898050   0.861475   0.893350
                             max     0.910800   0.888300   0.918300
current_balance              count  67.000000  72.000000  71.000000
                             mean    6.175687   5.231750   5.514577
                             std     0.237807   0.141795   0.225266
                             min     5.718000   4.899000   4.984000
                             25%     6.011500   5.139250   5.380000
                             50%     6.153000   5.225000   5.541000
                             75%     6.328000   5.337250   5.689500
                             max     6.675000   5.541000   5.920000
credit_limit                 count  67.000000  72.000000  71.000000
                             mean    3.697537   2.849542   3.259225
                             std     0.166014   0.138689   0.154766
                             min     3.387000   2.630000   2.936000
                             25%     3.564500   2.738500   3.155000
                             50%     3.719000   2.836500   3.258000
                             75%     3.808000   2.967000   3.378000
                             max     4.033000   3.232000   3.582000
min_payment_amt              count  67.000000  72.000000  71.000000
                             mean    3.632373   4.742389   2.707341
                             std     1.211052   1.354711   1.176440
                             min     1.472000   1.502000   0.765100
                             25%     2.848000   4.032250   1.951000
                             50%     3.619000   4.799000   2.640000
                             75%     4.421000   5.463750   3.332000
                             max     6.682000   8.456000   6.685000
max_spent_in_single_shopping count  67.000000  72.000000  71.000000
                             mean    6.041701   5.101722   5.120803
                             std     0.229566   0.184012   0.269558
                             min     5.484000   4.519000   4.605000
                             25%     5.879000   5.001000   4.958500
                             50%     6.009000   5.089000   5.132000
                             75%     6.192500   5.223500   5.263500
                             max     6.550000   5.491000   5.879000
sil_width                    count  67.000000  72.000000  71.000000
                             mean    0.468772   0.397473   0.339816
                             std     0.153712   0.159526   0.165898
                             min     0.029792   0.002713   0.005457
                             25%     0.419827   0.314599   0.234095
                             50%     0.523482   0.453462   0.371077
                             75%     0.574340   0.515146   0.479615
                             max     0.639285   0.587277   0.554103
```

Table 8. Cluster profile (Describe)

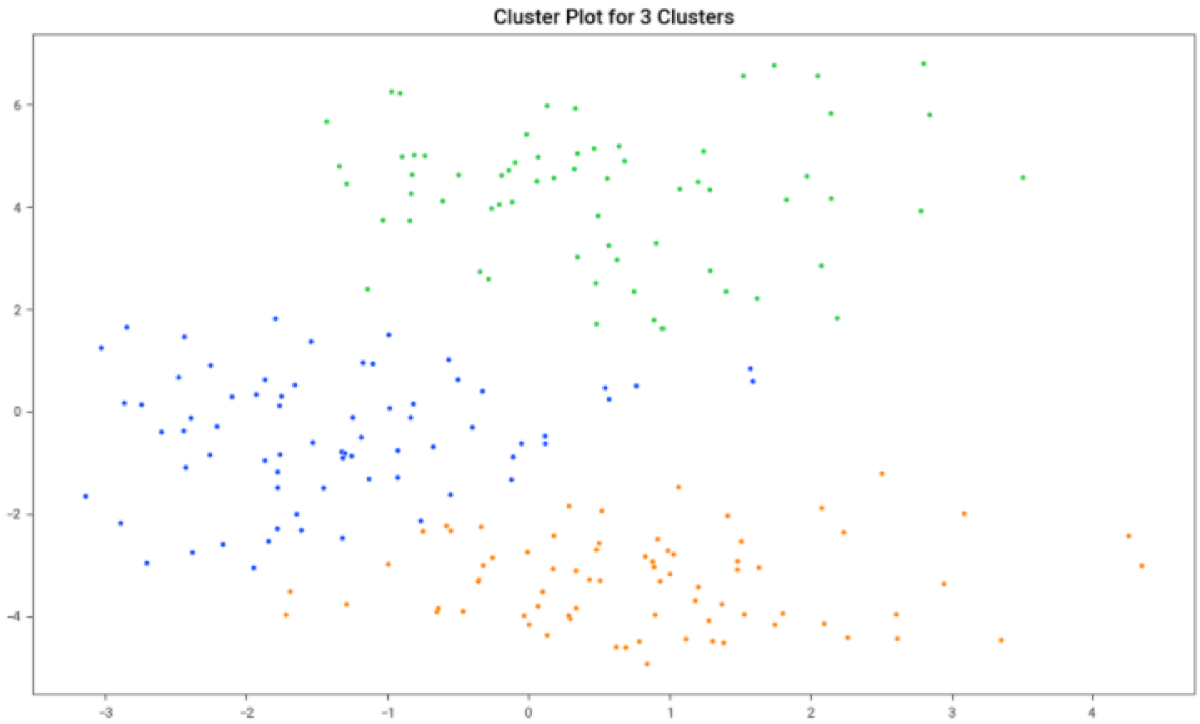


Fig 1.7. Scatter plot for 3 clusters

Insights:

- 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 and cluster 2 is 5100

Promotional strategies for each cluster:

Group 1: Cluster 0 -

The credit limit, advanced payment and the probability of full payment is least compared to other two clusters.

To improve their probability of full payment as well as to increase the advance payments, recommendation is to provide the gift vouchers to make more advance payments which will result in reduction not paying i.e., increase the probability of full payment.

Group 2: Cluster 1-

Maximum credit limits, average full payment record with maximum spending compared to other clusters.

Giving reward points might increase their purchases and maximum max_spent_in_single_shopping is high for this group, so can be offered discount/offer on next transactions upon full payment. Give loan against the credit card, as they are customers with good repayment record. Tie up with luxury brands, which will drive more one time maximum spending.

Group 3: Cluster 2 –

Average full payment history and having medium credit limit with medium values for most of the other variables

They are potential target customers who are paying bills and doing purchases and maintaining comparatively good credit score. So, we can increase credit limit or can lower down interest rate. Promote premium cards/loyalty cars to increase transactions. Increase spending habits by trying with premium ecommerce sites, travel portal, travel airlines/hotel, as this will encourage them to spend more.

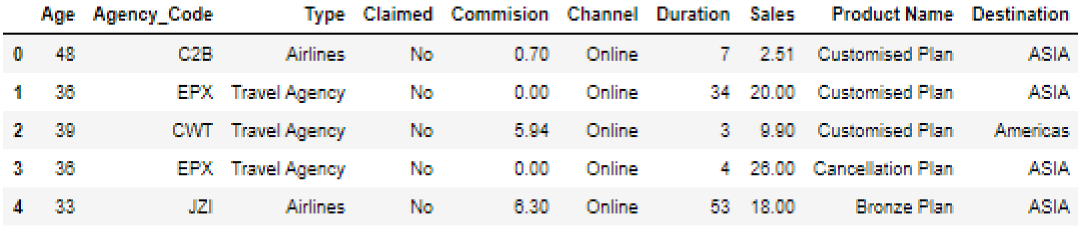
Problem 2: CART-RF-ANN

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.

Q2.1. Read the data, do the necessary initial steps, and exploratory data analysis

Solution:

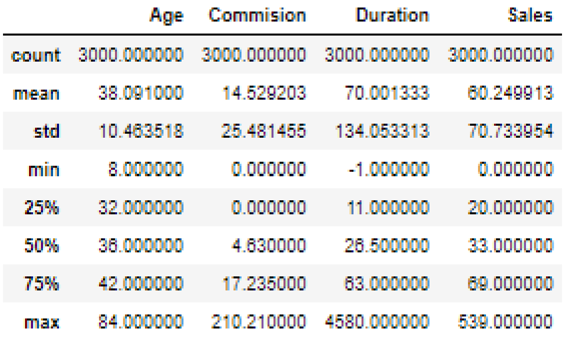
Sample of Dataset:



	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

Table 9. Dataset Sample (Insurance)

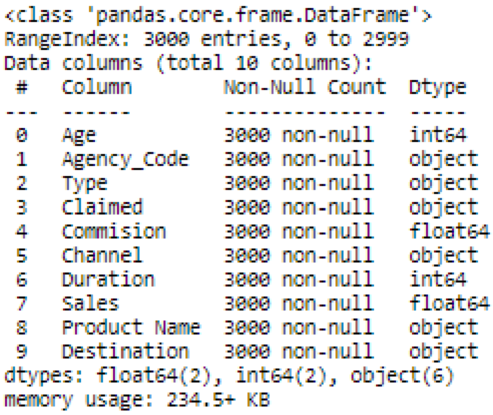
Summary of Dataset:



	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

Table 10. Dataset Summary (Insurance)

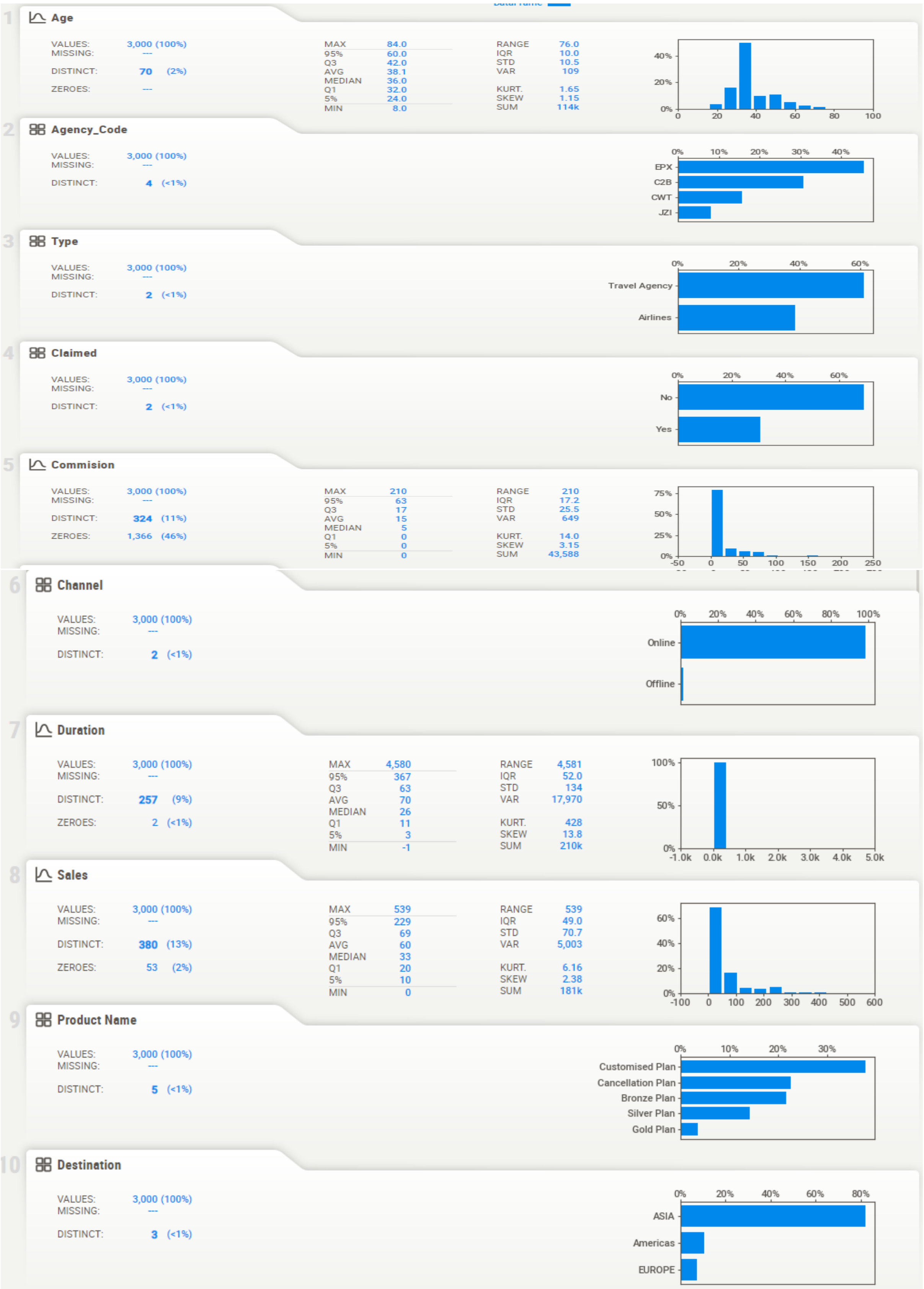
Type of Variables:



```
<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
```

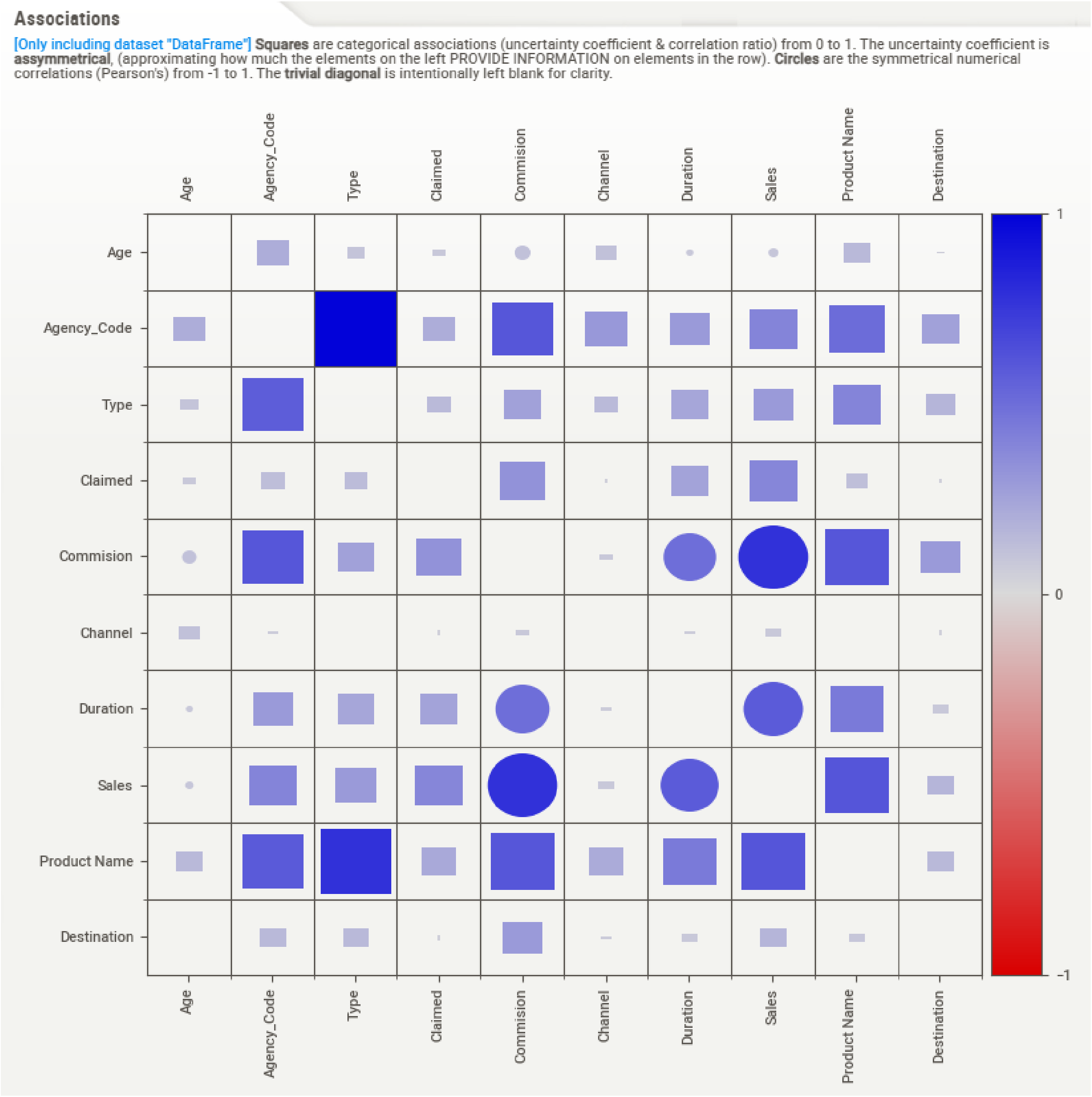
Table 11. Type of Variables (Insurance)

Data Visualization: EDA using sweet viz to visualize the summary for each variable as well to underrated data –



```
1  Age
   VALUES:    3,000 (100%)      MAX     84.0      RANGE  76.0
   MISSING:   ---               95%     60.0      IQR    10.0
                                Q3      42.0      STD    10.5
   DISTINCT:  70    (2%)        AVG     38.1      VAR    109
                                MEDIAN  36.0
   ZEROES:    ---               Q1      32.0      KURT.  1.65
                                5%      24.0      SKEW   1.15
                                MIN     8.0       SUM    114k

2  Agency_Code
   VALUES:    3,000 (100%)
   MISSING:   ---
   DISTINCT:  4  (<1%)

3  Type
   VALUES:    3,000 (100%)
   MISSING:   ---
   DISTINCT:  2  (<1%)

4  Claimed
   VALUES:    3,000 (100%)
   MISSING:   ---
   DISTINCT:  2  (<1%)

5  Commision
   VALUES:    3,000 (100%)      MAX     210       RANGE  210
   MISSING:   ---               95%     63        IQR    17.2
                                Q3      17        STD    25.5
   DISTINCT:  324   (11%)       AVG     15        VAR    649
                                MEDIAN  5
   ZEROES:    1,366 (46%)       Q1      0         KURT.  14.0
                                5%      0         SKEW   3.15
                                MIN     0         SUM    43,588

6  Channel
   VALUES:    3,000 (100%)
   MISSING:   ---
   DISTINCT:  2  (<1%)

7  Duration
   VALUES:    3,000 (100%)      MAX     4,580     RANGE  4,581
   MISSING:   ---               95%     367       IQR    52.0
                                Q3      63        STD    134
   DISTINCT:  257   (9%)        AVG     70        VAR    17,970
                                MEDIAN  26
   ZEROES:    2     (<1%)       Q1      11        KURT.  428
                                5%      3         SKEW   13.8
                                MIN     -1        SUM    210k

8  Sales
   VALUES:    3,000 (100%)      MAX     539       RANGE  539
   MISSING:   ---               95%     229       IQR    49.0
                                Q3      69        STD    70.7
   DISTINCT:  380   (13%)       AVG     60        VAR    5,003
                                MEDIAN  33
   ZEROES:    53    (2%)        Q1      20        KURT.  6.16
                                5%      10        SKEW   2.38
                                MIN     0         SUM    181k

9  Product Name
   VALUES:    3,000 (100%)
   MISSING:   ---
   DISTINCT:  5  (<1%)

10 Destination
   VALUES:    3,000 (100%)
   MISSING:   ---
   DISTINCT:  3  (<1%)
```

Fig 2.1. Sweet viz Univariate analysis

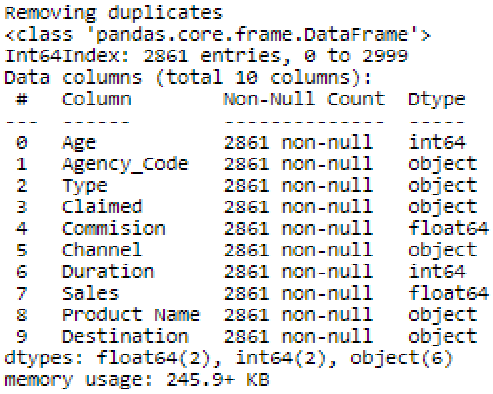


Associations

[Only including dataset "DataFrame"] **Squares** are categorical associations (uncertainty coefficient & correlation ratio) from 0 to 1. The uncertainty coefficient is **assymmetrical**, (approximating how much the elements on the left PROVIDE INFORMATION on elements in the row). **Circles** are the symmetrical numerical correlations (Pearson's) from -1 to 1. The **trivial diagonal** is intentionally left blank for clarity.

Fig 2.2. Sweet viz Multivariate analysis

We have 139 number of duplicates, data post removing the duplicates and by keeping the last updated duplicates:



```
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
```

Table 12. Type of Variables (Insurance post removing duplicates)

Boxplot to check the outliers:

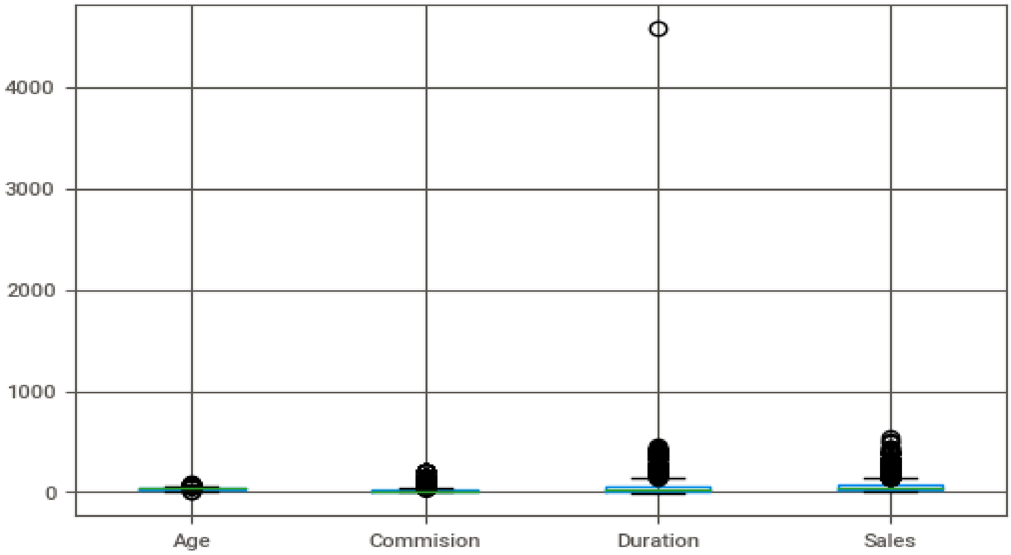


Fig 2.3. Boxplot (Insurance)

Boxplot of the dataset post handling the outliers:

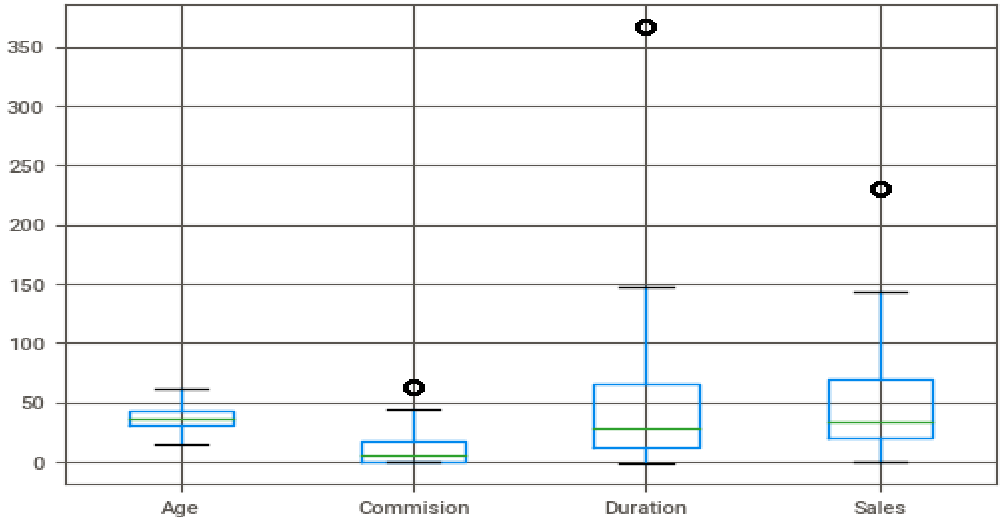


Fig 2.4. Boxplot post handling outliers (Insurance)

Observation (EDA):

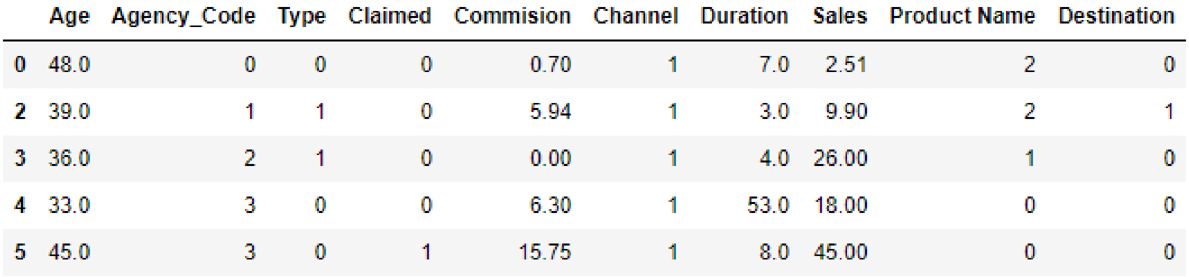
- We have 9 independent variables and 1 target variable with no missing values
- Dataset had 139 duplicates and outliers which have been removed and handled
- Scaling might be required for some models as the difference in scale of data between the variables
- There is high categorical correlation between Agency code, Type, commission and product name variables
- There is high numerical correlation between Commission and Sales Variables

Q2.2 Data Split: Split the data into test and train, build classification model CART, Random Forest, Artificial Neural Network

Solution:

Data split –

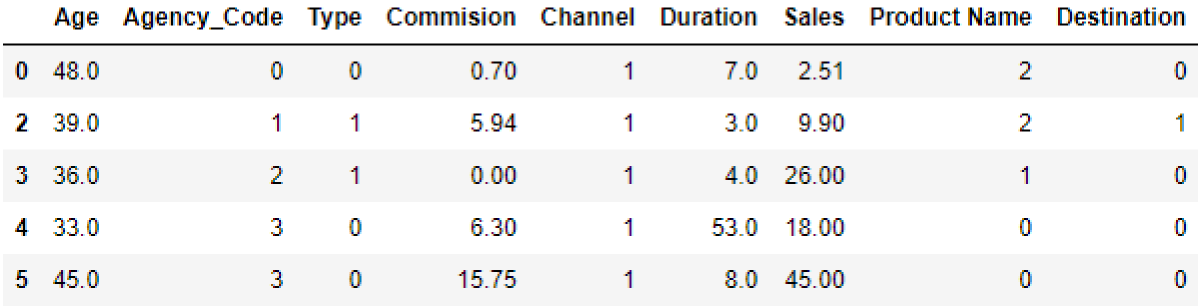
Head of dataset post conversion of object variables into categorical codes:



	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

Table 13. Head of dataset (Insurance post conversion of variables)

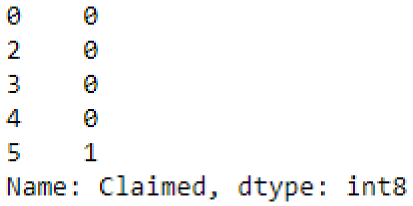
Head of Independent variables (extracted from above dataset):



	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
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
5	45.0	3	0	15.75	1	8.0	45.00	0	0

Table 14. Head of independent variables

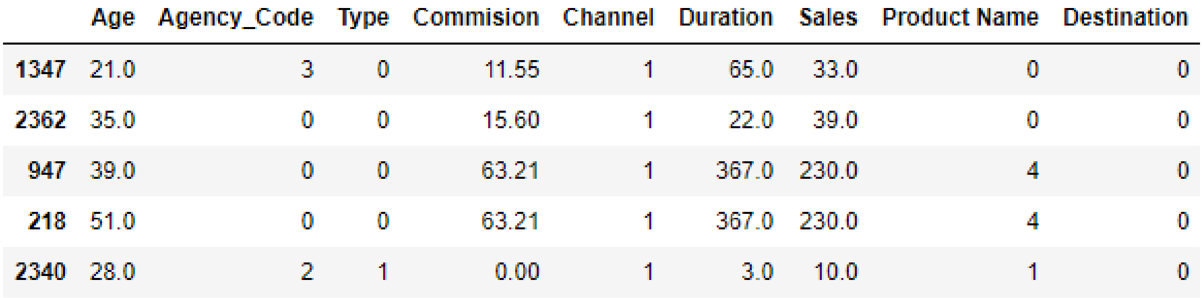
Head of Dependent variable:



```
0    0
2    0
3    0
4    0
5    1
Name: Claimed, dtype: int8
```

Table 15. Head of Dependent variable

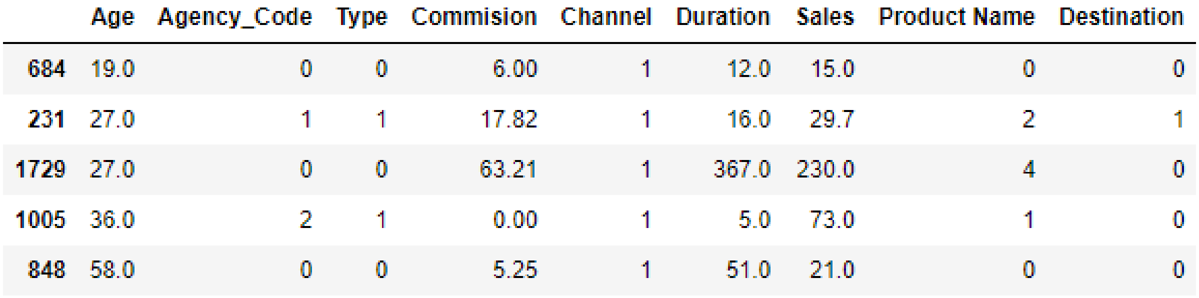
Head of Trained data after splitting (Independent variables):



	Age	Agency_Code	Type	Commision	Channel	Duration	Sales	Product Name	Destination
1347	21.0	3	0	11.55	1	65.0	33.0	0	0
2362	35.0	0	0	15.60	1	22.0	39.0	0	0
947	39.0	0	0	63.21	1	367.0	230.0	4	0
218	51.0	0	0	63.21	1	367.0	230.0	4	0
2340	28.0	2	1	0.00	1	3.0	10.0	1	0

Table 16. Head of Trained data (Independent variables)

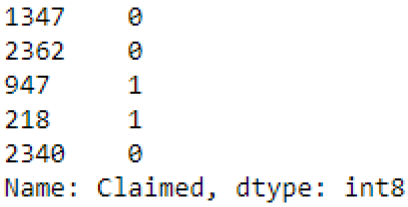
Head of Test data after splitting (Independent variables):



	Age	Agency_Code	Type	Commision	Channel	Duration	Sales	Product Name	Destination
684	19.0	0	0	6.00	1	12.0	15.0	0	0
231	27.0	1	1	17.82	1	16.0	29.7	2	1
1729	27.0	0	0	63.21	1	367.0	230.0	4	0
1005	36.0	2	1	0.00	1	5.0	73.0	1	0
848	58.0	0	0	5.25	1	51.0	21.0	0	0

Table 17. Head of Test data (Independent variables)

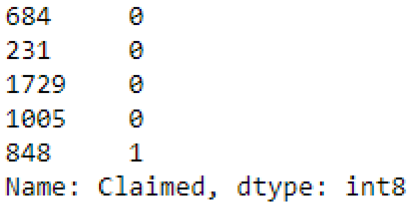
Head of Train labels (dependent variable):



```
1347    0
2362    0
947     1
218     1
2340    0
Name: Claimed, dtype: int8
```

Table 18. Head of Train labels (Dependent variable)

Head of Test labels (dependent variable):



```
684     0
231     0
1729    0
1005    0
848     1
Name: Claimed, dtype: int8
```

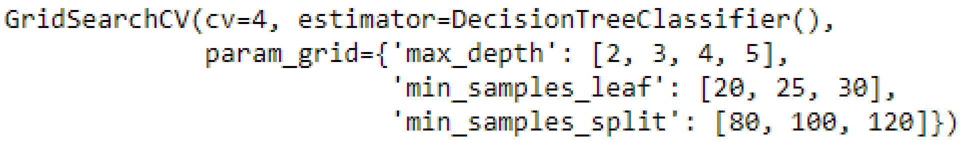
Table 19. Head of Test labels (Dependent variable)

Inference:

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

CART Decision Tree –

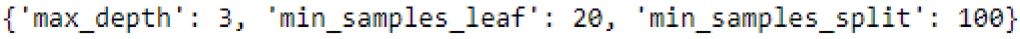
Optimization metrics used:



```
GridSearchCV(cv=4, estimator=DecisionTreeClassifier(),
             param_grid={'max_depth': [2, 3, 4, 5],
                         'min_samples_leaf': [20, 25, 30],
                         'min_samples_split': [80, 100, 120]})
```

Fig 2.5. Optimization metric (CART)

Best optimized parameters:



```
{'max_depth': 3, 'min_samples_leaf': 20, 'min_samples_split': 100}
```

Fig 2.6. Best optimized parameters (CART)

Decision tree with best optimized parameters:

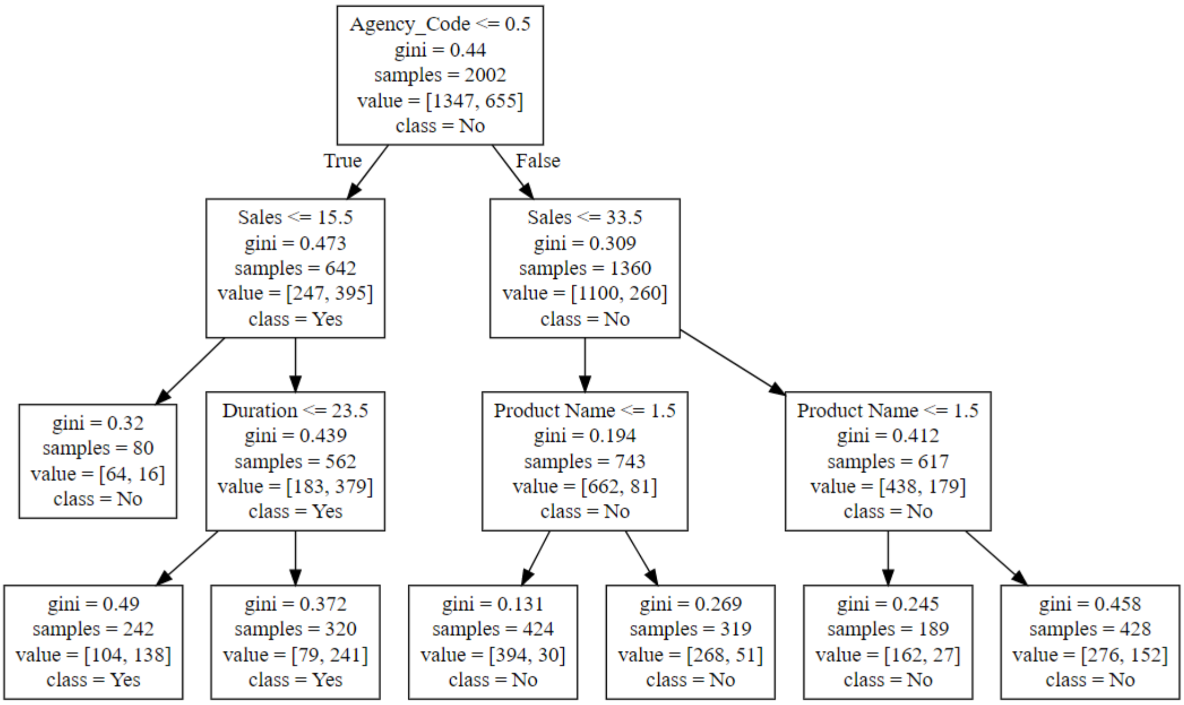
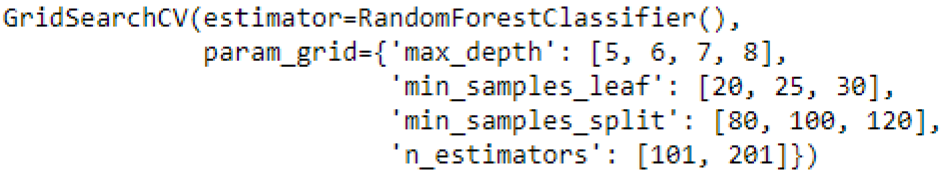


Fig 2.7. Decision tree (Best optimized parameters)

Random Forest –

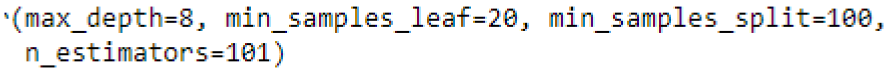
Optimization metrics used:



```
GridSearchCV(estimator=RandomForestClassifier(),
             param_grid={'max_depth': [5, 6, 7, 8],
                         'min_samples_leaf': [20, 25, 30],
                         'min_samples_split': [80, 100, 120],
                         'n_estimators': [101, 201]})
```

Fig 2.8. Optimization metric (RF)

Best optimized parameters:

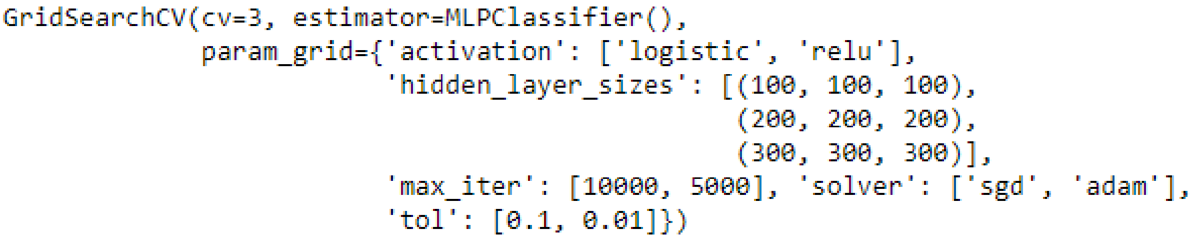


```
(max_depth=8, min_samples_leaf=20, min_samples_split=100,
 n_estimators=101)
```

Fig 2.9. Best optimized parameters (RF)

MLP Classifier (Artificial Neural Network) –

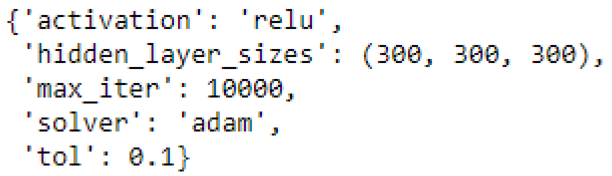
Optimization metrics used:



```
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]})
```

Fig 2.10. Optimization metric (ANN)

Best optimized parameters:



```
{'activation': 'relu',
 'hidden_layer_sizes': (300, 300, 300),
 'max_iter': 10000,
 'solver': 'adam',
 'tol': 0.1}
```

Fig 2.11. Best optimized parameters (ANN)

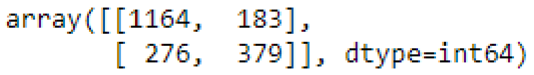
Data is split into train and test for three different models namely Decision Tree, Random Forest and Artificial Neural Network is built.

Q2.3 Performance Metrics: Comment and Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score, classification reports for each model.

Solution:

CART Decision Tree –

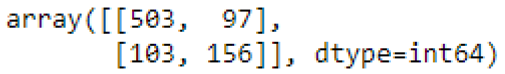
- Accuracy score (Train data): 0.7707292707292708
- Accuracy score (Test data): 0.7671711292200233
- Confusion Matrix (Train data):



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

Fig 2.12. Confusion Matrix (CART Train data)

- Confusion Matrix (Test data):



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

Fig 2.13. Confusion Matrix (CART Test data)

- ROC_AUC score (Train data): 0.8007
- ROC_AUC score (Test data): 0.7869

- ROC curve (Train data):

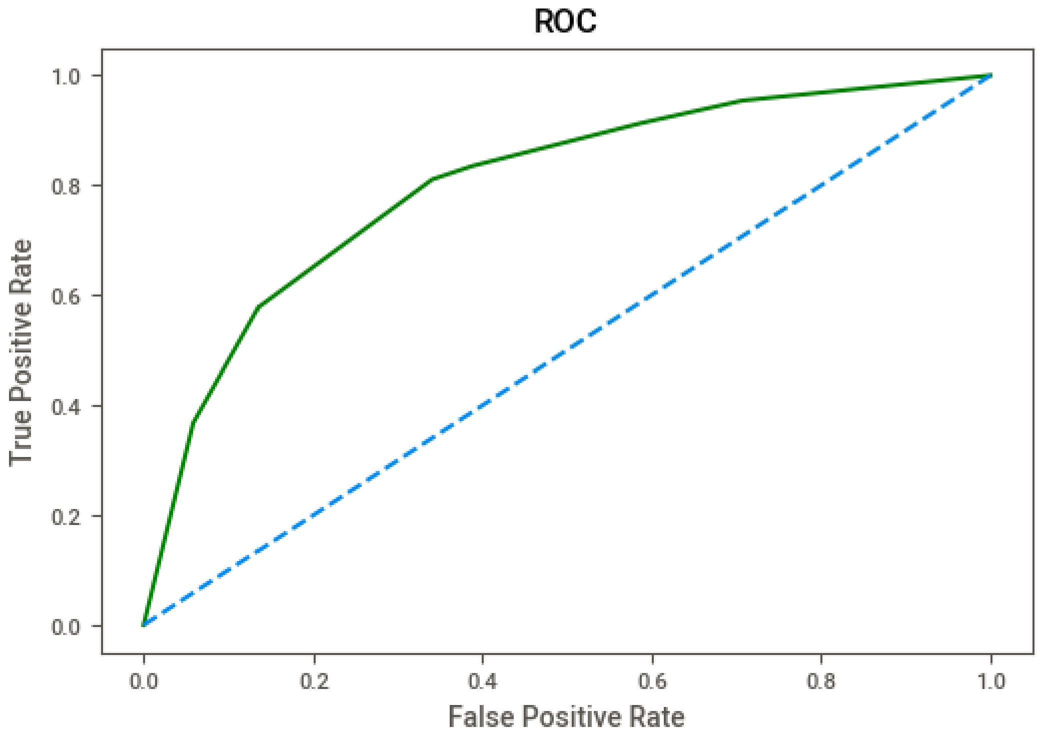


Fig 2.14. ROC curve (CART Train data)

- ROC curve (Test data):

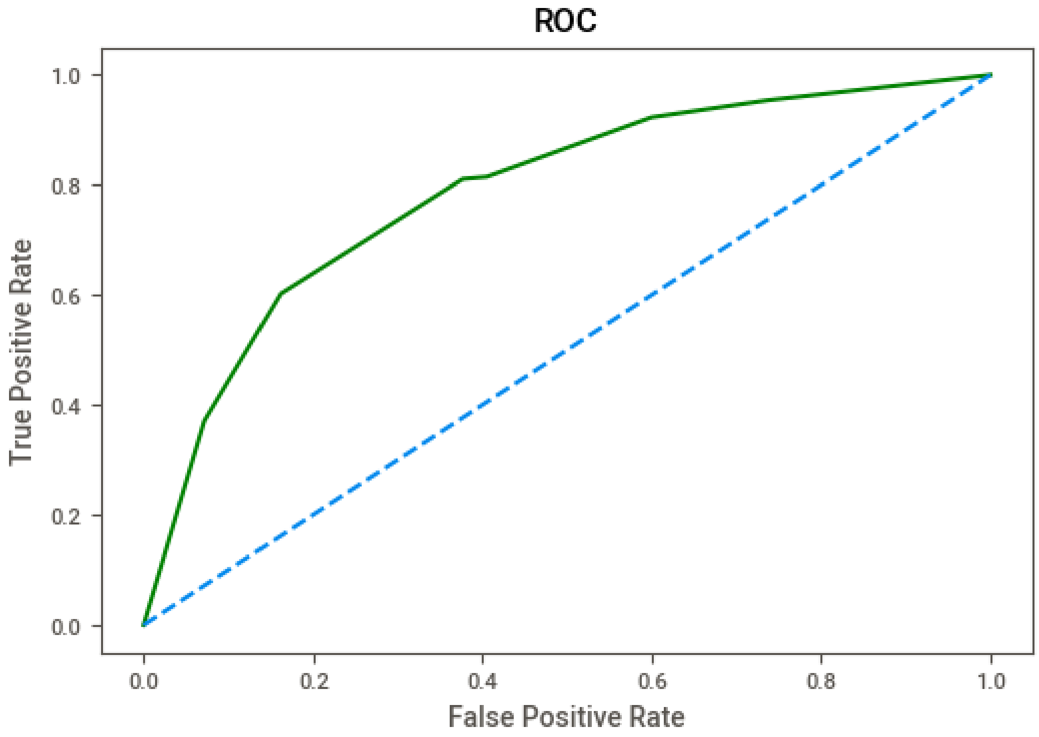
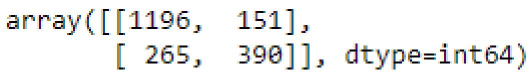


Fig 2.15. ROC curve (CART Test data)

Random Forest –

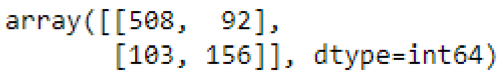
- Accuracy score (Train data): 0.7922077922077922
- Accuracy score (Test data): 0.7729918509895227
- Confusion Matrix (Train data):



```
array([[1196,  151],
       [ 265,  390]], dtype=int64)
```

Fig 2.16. Confusion Matrix (RF Train data)

- Confusion Matrix (Test data):



```
array([[508,  92],
       [103, 156]], dtype=int64)
```

Fig 2.17. Confusion Matrix (RF Test data)

- ROC_AUC score (Train data): 0.8359
- ROC_AUC score (Test data): 0.8013
- ROC curve (Train data):

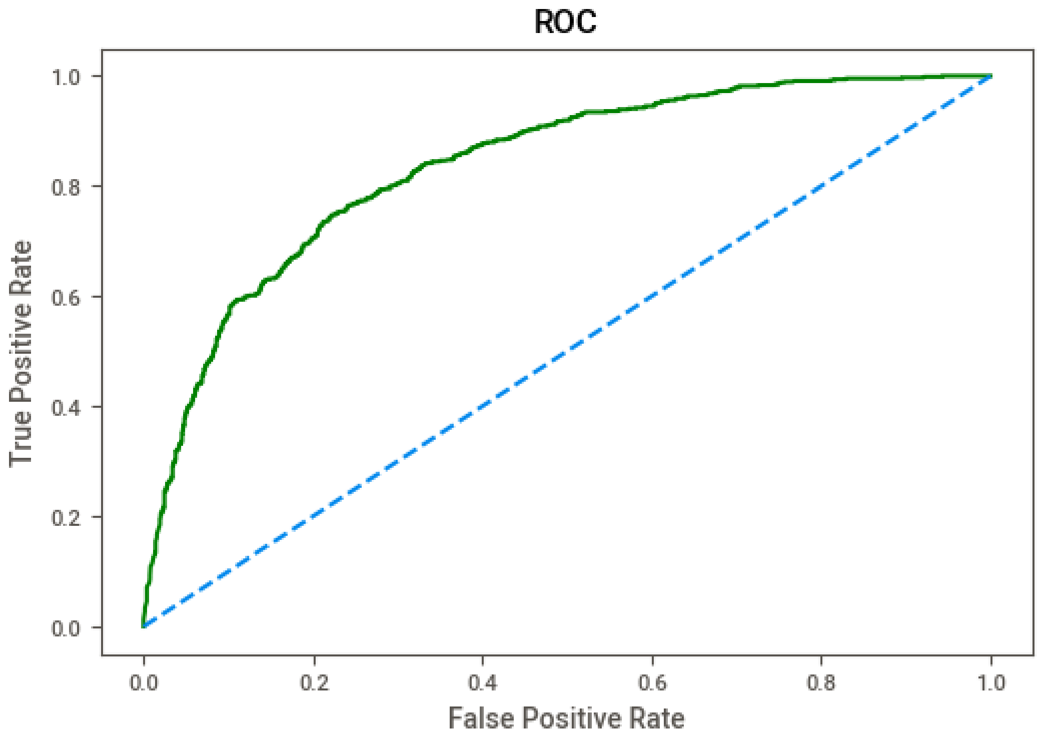


Fig 2.18. ROC curve (RF Train data)

- ROC curve (Test data):

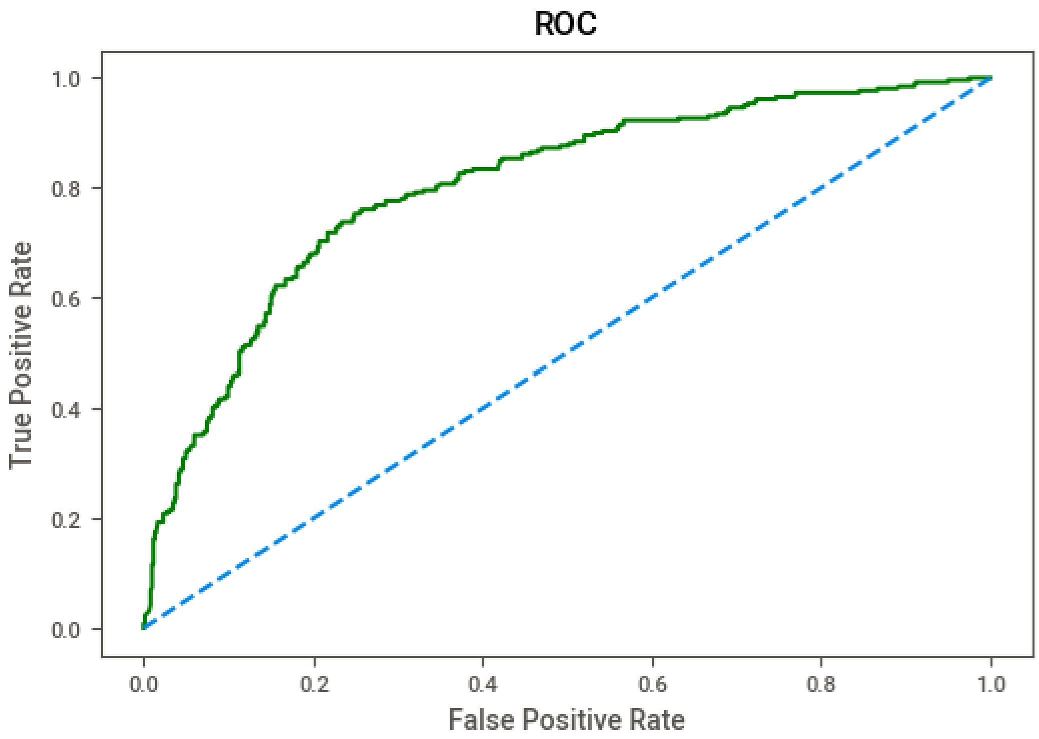
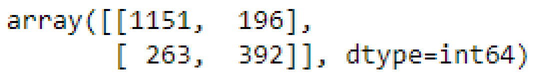


Fig 2.19. ROC curve (RF Test data)

MLP Classifier (Artificial Neural Network) –

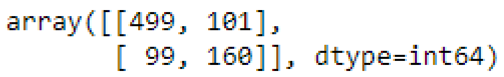
- Accuracy score (Train data): 0.7707292707292708
- Accuracy score (Test data): 0.7671711292200233
- Confusion Matrix (Train data):



```
array([[1151,  196],
       [ 263,  392]], dtype=int64)
```

Fig 2.20. Confusion Matrix (ANN Train data)

- Confusion Matrix (Test data):



```
array([[499, 101],
       [ 99, 160]], dtype=int64)
```

Fig 2.21. Confusion Matrix (ANN Test data)

- ROC_AUC score (Train data): 0.8082
- ROC_AUC score (Test data): 0.7903

- ROC curve (Train data):

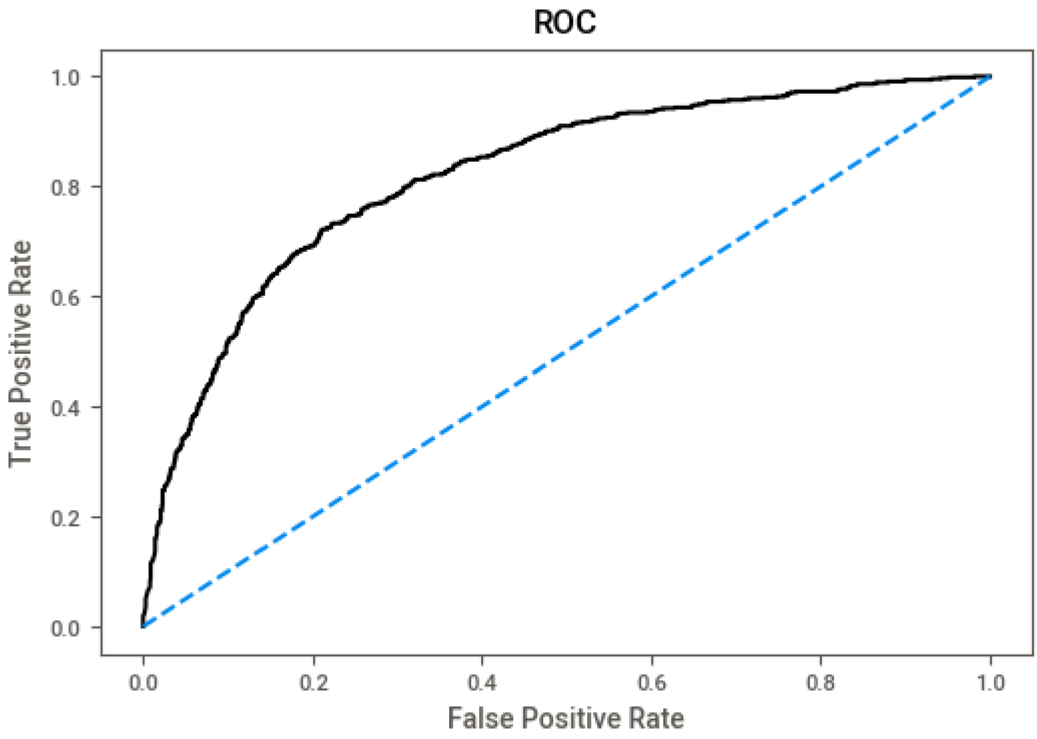


Fig 2.22. ROC curve (ANN Train data)

- ROC curve (Test data):

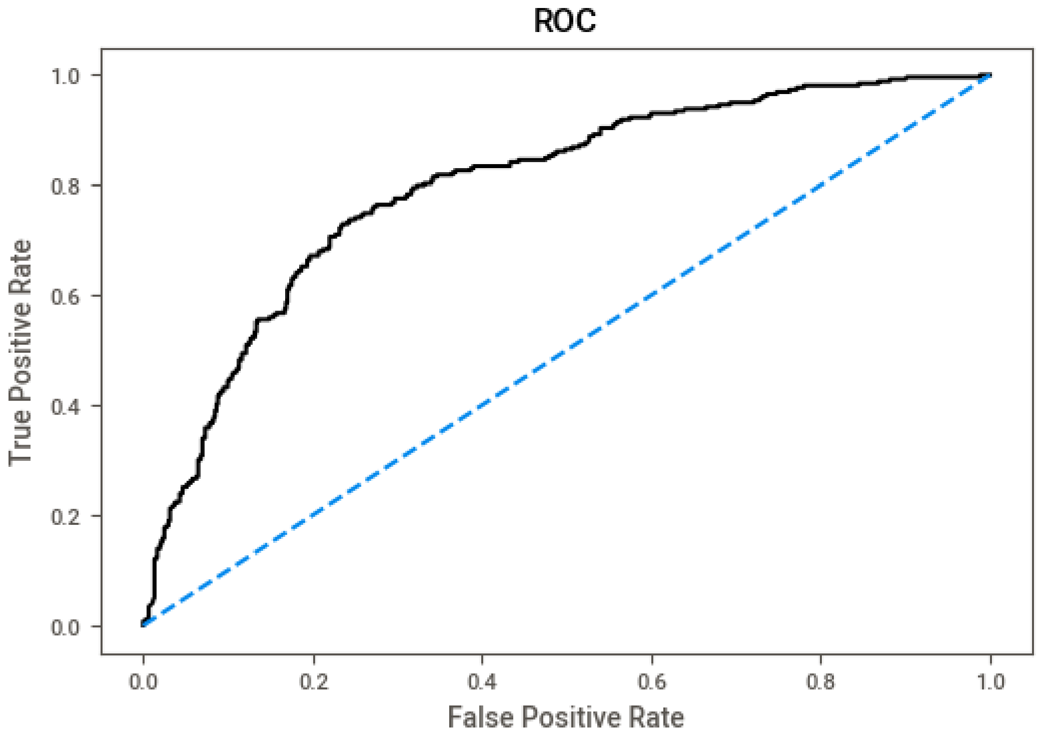


Fig 2.23. ROC curve (ANN Test data)

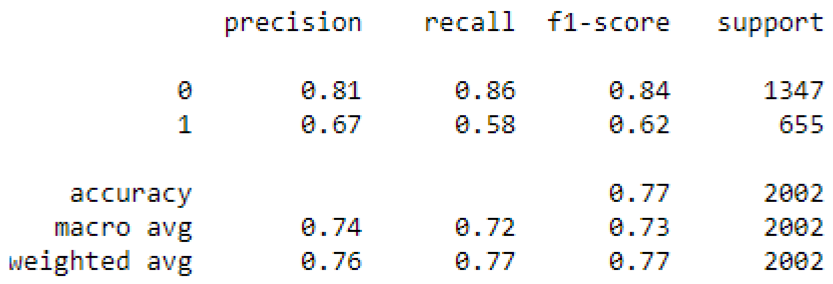
Training and Test set results are almost similar, and with the overall measures high, the model is a good model.

Q2.4. Final Model: Compare all the models and write an inference which model is best/optimized.

Solution:

CART Decision Tree performance matrix–

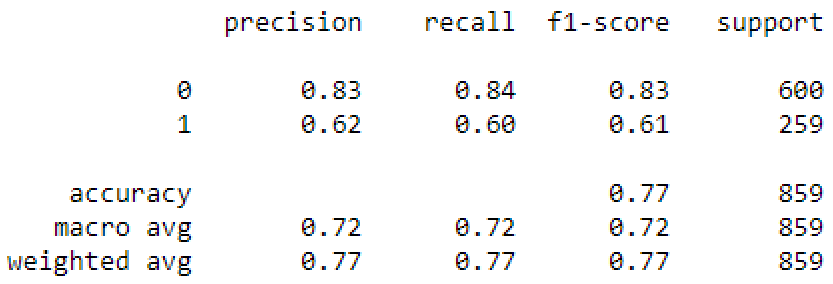
- Training set:



```
              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
```

Table 20. Performance matrix (CART – Train set)

- Testing set:

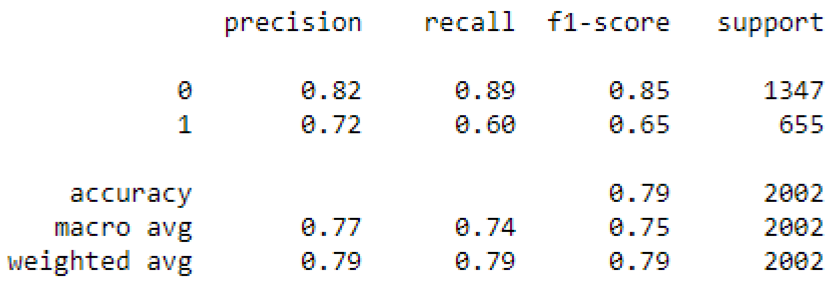


```
              precision    recall  f1-score   support

           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
```

Table 21. Performance matrix (CART – Test set)

Random Forest performance matrix–

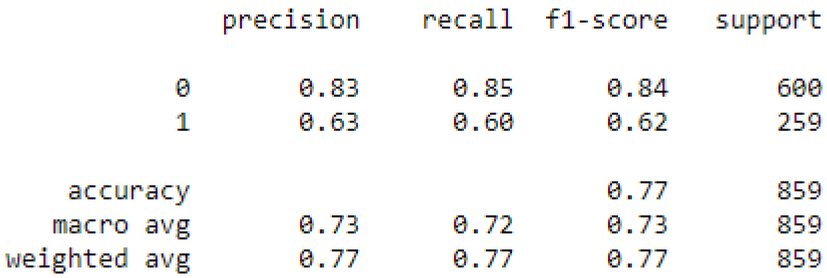
- Training set:



```
              precision    recall  f1-score   support

           0       0.82      0.89      0.85      1347
           1       0.72      0.60      0.65       655

    accuracy                           0.79      2002
   macro avg       0.77      0.74      0.75      2002
weighted avg       0.79      0.79      0.79      2002
```

Table 22. Performance matrix (RF – Train set)

- Testing set:

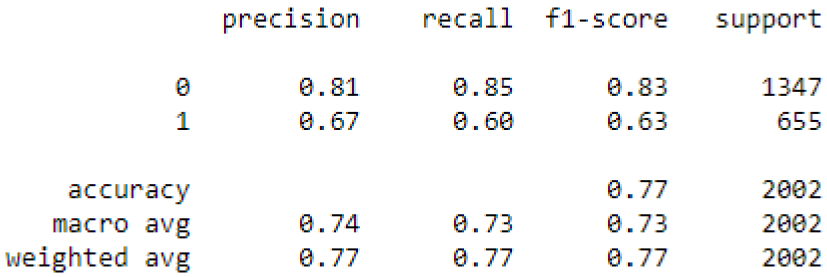


```
              precision    recall  f1-score   support

           0       0.83      0.85      0.84       600
           1       0.63      0.60      0.62       259

    accuracy                           0.77       859
   macro avg       0.73      0.72      0.73       859
weighted avg       0.77      0.77      0.77       859
```

Table 23. Performance matrix (RF – Test set)

Artificial Neural Network performance matrix–

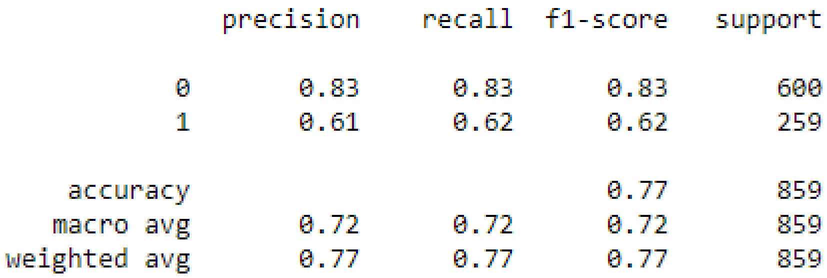
- Training set:



```
              precision    recall  f1-score   support

           0       0.81      0.85      0.83      1347
           1       0.67      0.60      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
```

Table 24. Performance matrix (ANN – Train set)

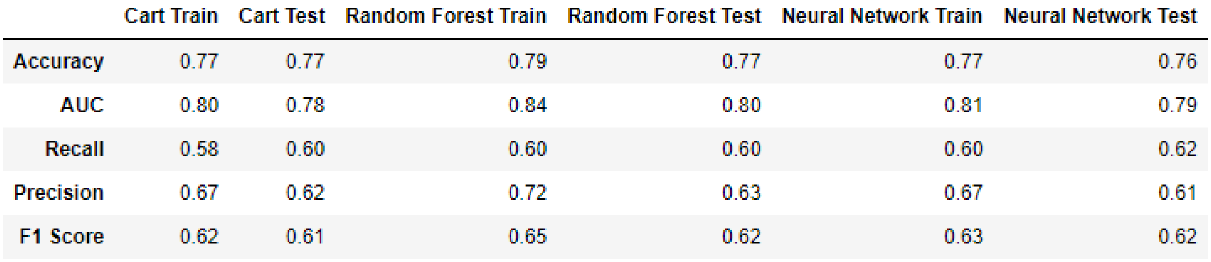
- Testing set:



```
              precision    recall  f1-score   support

           0       0.83      0.83      0.83       600
           1       0.61      0.62      0.62       259

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

Table 25. Performance matrix (ANN – Test set)

Comparison of the performance metrics from the 3 models:



	Cart Train	Cart Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.77	0.77	0.79	0.77	0.77	0.76
AUC	0.80	0.78	0.84	0.80	0.81	0.79
Recall	0.58	0.60	0.60	0.60	0.60	0.62
Precision	0.67	0.62	0.72	0.63	0.67	0.61
F1 Score	0.62	0.61	0.65	0.62	0.63	0.62

Table 26. Performance matrix (All three models)

ROC Curve for the 3 models on the Training data:

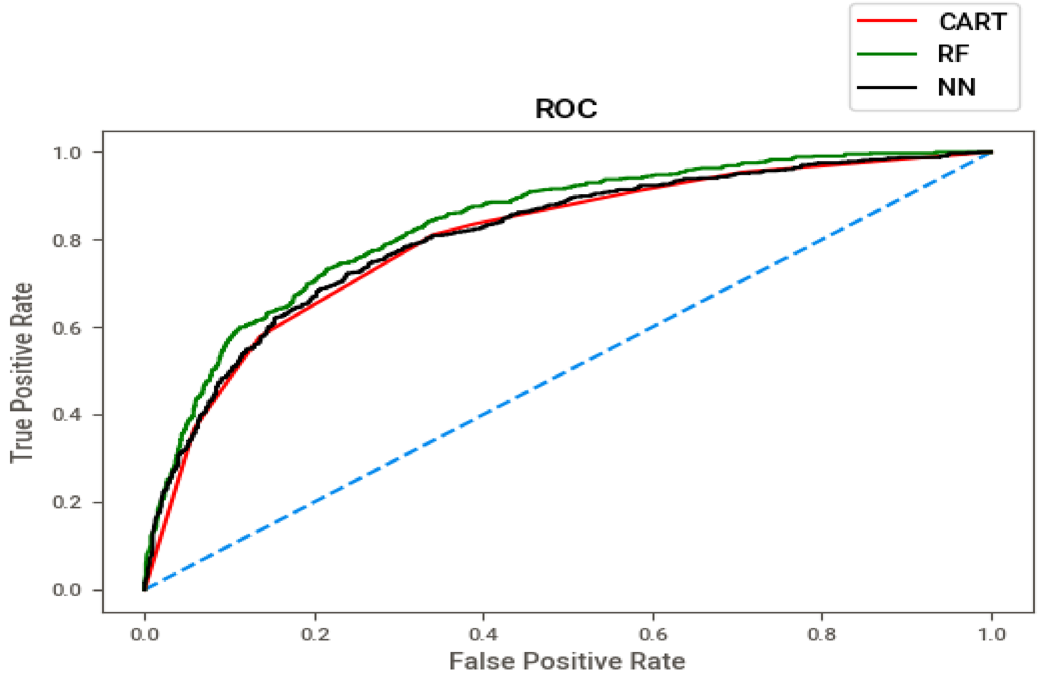


Fig 2.24. ROC curve All 3 models (Train Data)

ROC Curve for the 3 models on the Testing data:

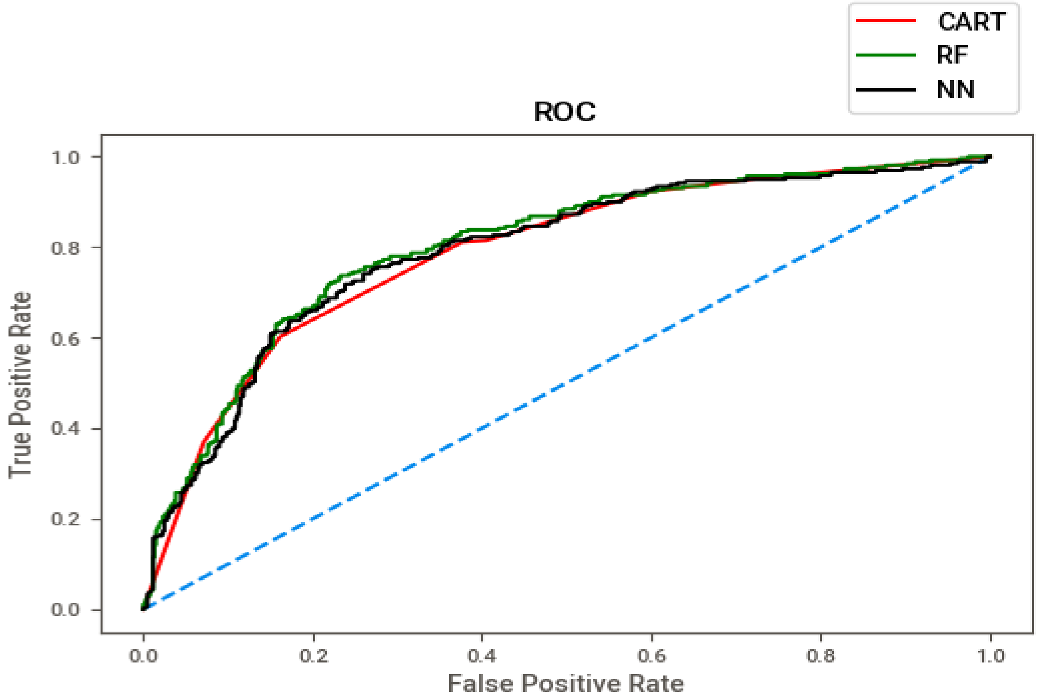


Fig 2.25. ROC curve All 3 models (Test Data)

CONCLUSION: Selecting the RF model, as it has better accuracy, precision, recall, f1 score better than other two CART & ANN

Q2.5. Inference: Based on the whole Analysis, what are the business insights and recommendations

Inference:

I strongly recommended we collect more real time unstructured data and past data if possible.

This is understood by looking at the insurance data by drawing relations between different variables such as day of the incident, time, age group, and associating it with other external information such as location, behavior patterns, weather information, airline/vehicle types, etc.

- Streamlining online experiences benefitted customers, leading to an increase in conversions, which subsequently raised profits.
- As per the data 90% of insurance is done by online channel.
- Other interesting fact, is almost all the offline business has a claimed associated, need to find why?
- Need to train the JZI agency resources to pick up sales as they are in bottom, need to run promotional marketing campaign or evaluate if we need to tie up with alternate agency.
- Also based on the model we are getting 80%accuracy, so we need customer books airline tickets or plans, cross sell the insurance based on the claim data pattern.
- Other interesting fact is more sales happen via Agency than Airlines and the trend shows the claim are processed more at Airline. So, we may need to deep dive into the process to understand the workflow and why?

Key performance indicators (KPI) The KPI's of insurance claims are:

- Reduce claims cycle time
- Increase customer satisfaction
- Combat fraud
- Optimize claims recovery
- Reduce claim handling costs Insights gained from data and AI-powered analytics could expand the boundaries of insurability, extend existing products, and give rise to new risk transfer solutions in areas like a non-damage business interruption and reputational damage

Thanks & regards,
Pavan Kumar R Naik