DATA MINING ASSIGNMENT

SABITA NAIR PANCHAL

DSBA - BATCH FEBRUARY, 2021

SUBMISSION DATE - 27/06/2021

PROBLEM 1: CLUSTERING

Executive Summary:

A leading bank wants to develop a customer segmentation to give promotional offers to its customers. A sample data set of 210 customers has been collected that summarizes their activities over the past few months. The task on hand is to identify various segments of customers based on credit card usage and to profile them, so as to provide a sound basis for differentiated campaigns / targeted activities in future.

1.1 Exploratory data analysis:

The sample data has 210 entries, with values under 7 columns. The data provided is clean and precise, with no duplicated values or null values.

Table: 1.1.1

	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

A brief description of the column heads:

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)

On checking the data, every column contained exactly 210 non-null values of the type ‘float’.

Table: 1.1.2

Columns	Non-null count	Data-type
spending	210 non-null	float64
advance_payments	210 non-null	float64
probability_of_full_payment	210 non-null	float64
current_balance	210 non-null	float64
credit_limit	210 non-null	float64
min_payment_amt	210 non-null	float64
max_spent_in_single_shopping	210 non-null	float64

Using the ‘describe()’ function, the statistical summary of the data was derived as follows:

Table: 1.1.3

	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	**14.847524**	**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**	**0.49148**
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**

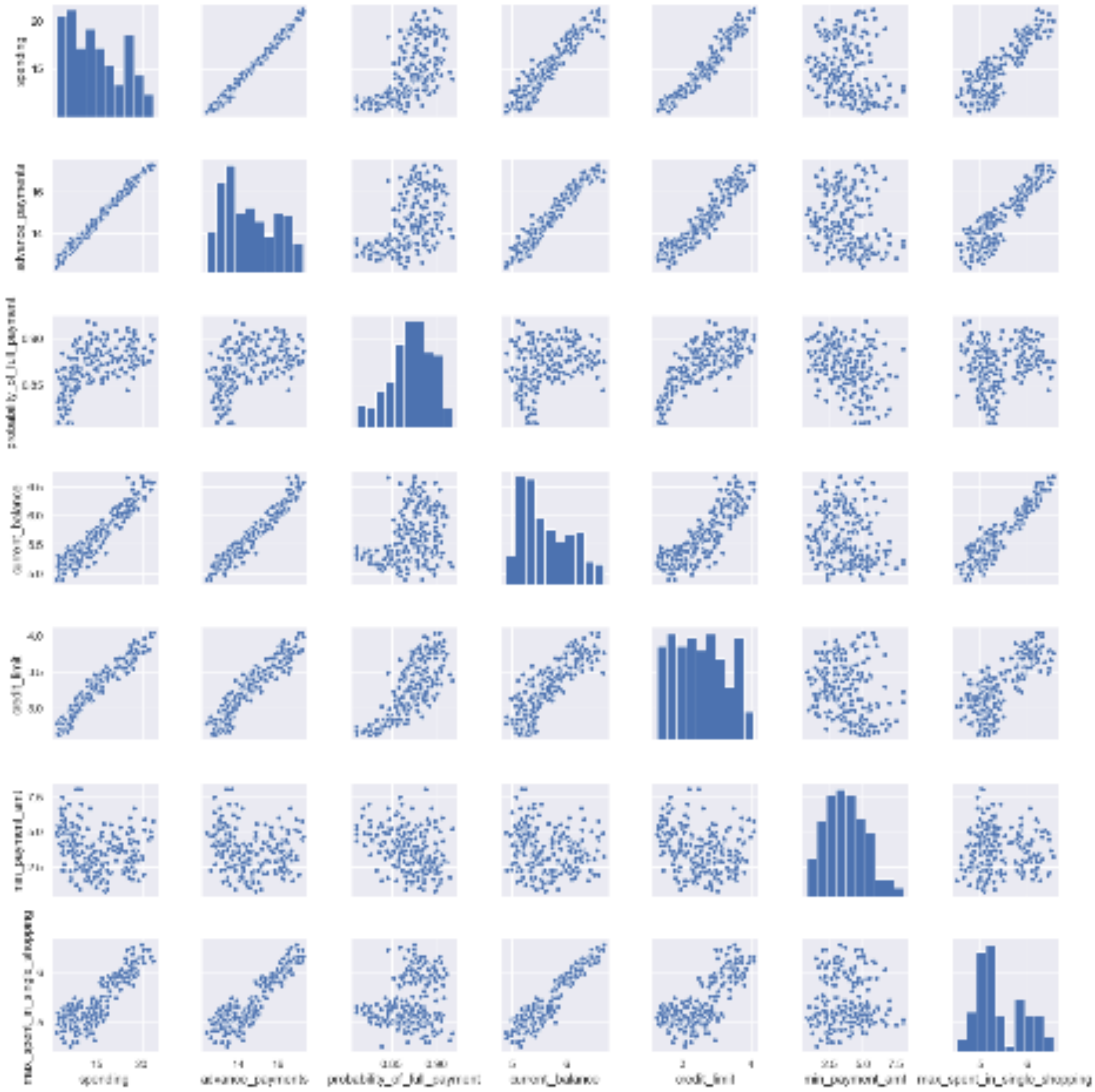
Based on the statistical summary, we can make a few observations:

- On an average, customers spent about Rs. 14,847.524 per month on purchases and transactions.
- They paid almost 10% of that amount (approx. Rs. 1,456) in advance credit card payments in cash.
- The average probability that customers make a full payment of their credit cards to the bank is approx. 87%.
- On an average, customers maintain approx. Rs. 5,628 in their accounts to make future purchases.

- By and large, customers had a credit limit of approx. Rs. 32,586 on their credit cards.
- Ordinarily, the minimum amount paid by customers while making payments for purchases made monthly is approx. Rs.370.
- Generally speaking, the maximum amount spent by customers in a single purchasing trip was about Rs. 5408.
- The variation within the data (measured here by standard deviation) shows little deviation. Maximum variation in the data is seen in the spending patterns of customers, followed by minimum amounts paid for purchases and the advance payments done on the credit cards.

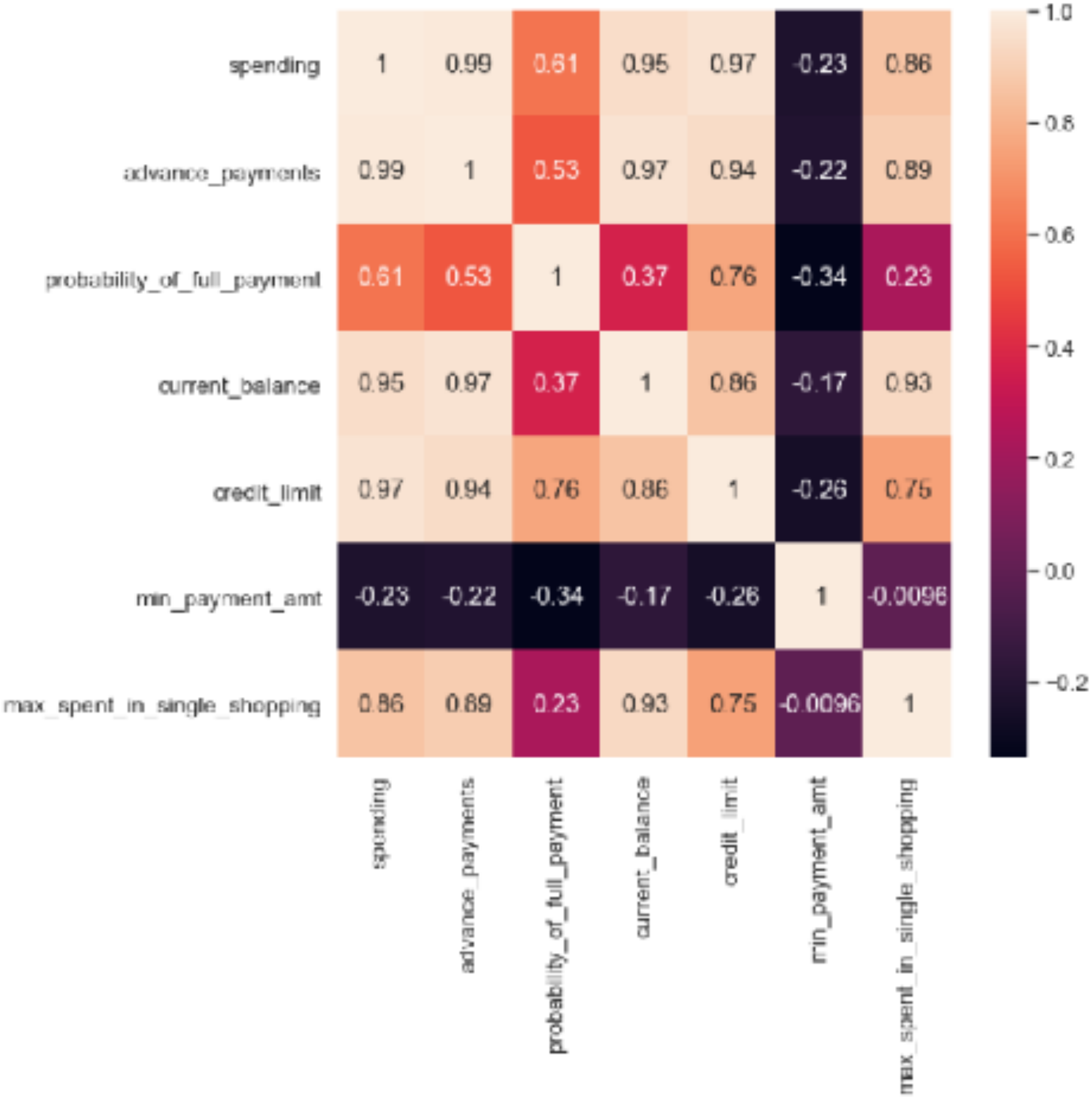
Further, correlation among variables was checked using both a pair plot, so that a fair idea of the interdependence of variables could be derived. The results are as follows:

Graph: 1.1.1



A heat map was further used to derive the exact correlations.

Graph: 1.1.2



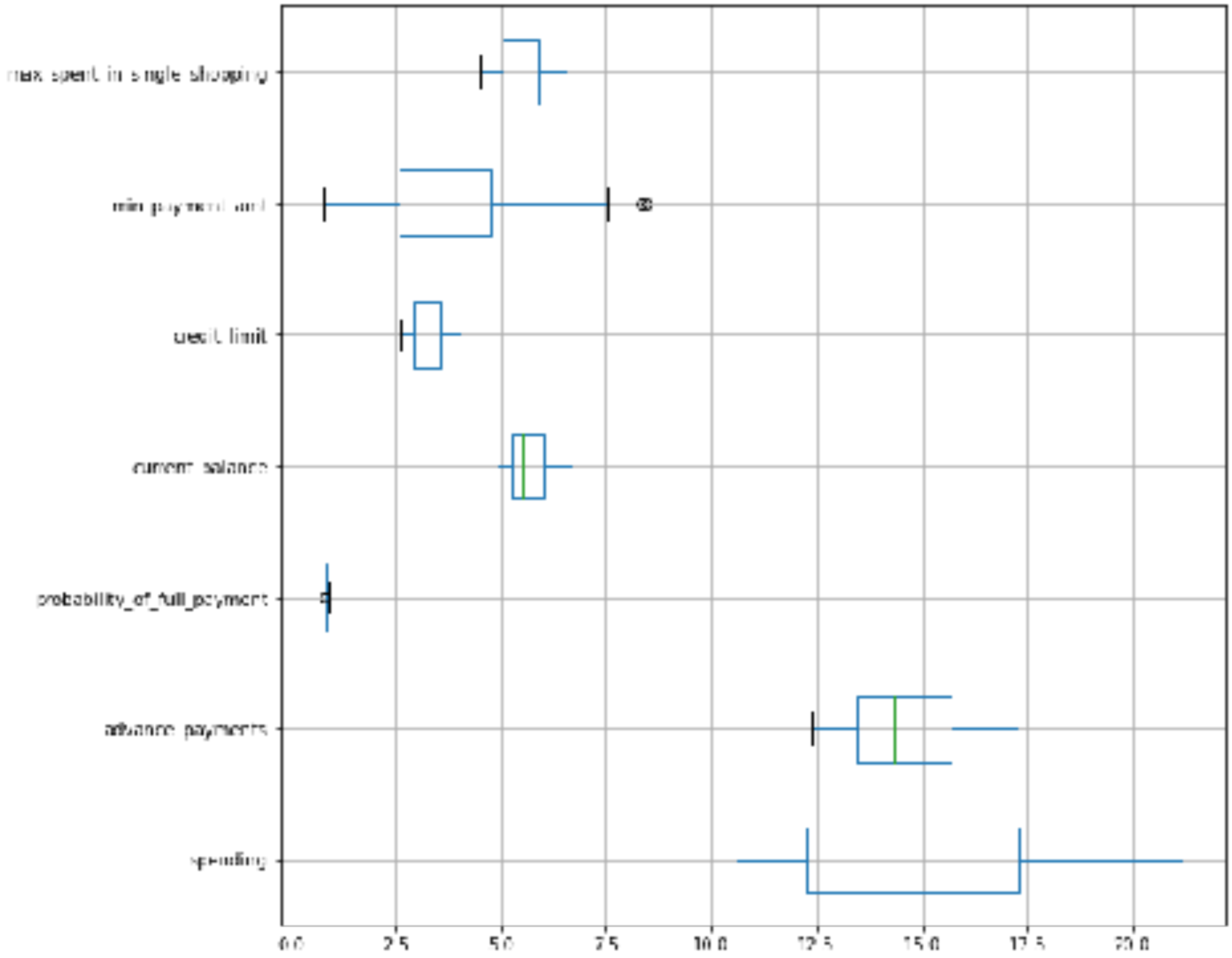
Based on the pair plot and the heat map, it was evident that strong correlations existed between the variables. The prominent pairs of correlated variables, who seemed to enjoy a high degree of interdependence amongst them, were:

- ‘spending’ and ‘advance payments’ - implying that the higher the spending patten of a customer, the higher amount of advance payments he/she was likely to make.
- ‘spending’ and ‘credit_limit’ - meaning that the higher the credit limit a customer enjoyed, the more like he/she was to spend high on purchases.
- ‘advance_payments’ and ‘current_balance’ - this means that the higher balance a customer maintains in his/her account, the more he was likely to pay a higher amount in advance towards credit card payments.

The least correlation was seen between the variables ‘probability_of_full_payment’ and ‘min_payment_amt’, meaning that there is almost no bearing of the minimum amount paid by a customer towards credit card bills to the probability that he/she would make a full payment against his credit card outstanding.

Since clustering techniques are sensitive to outliers, a box plot was used to check for outliers present in the data. The following box plot was revealed.

Graph: 1.1.3



There were presence of few outliers in two of the columns, viz. ‘min_payment_amt’ and ‘probability_of_full-payment’. Although outliers were not too many, they were still removed (via the flooring and capping method), to avoid any negative impact on the clustering process.

1.2 Scaling of data:

- The data across columns had a large difference in magnitudes - some variables were in the range of 1,000s, while some in 100s and others in 10,000s. This deviation in magnitudes might affect the clustering process, thus scaling of the data was necessary.

- Also, one of the variables is a probability, unlike other variables, which are all in terms of money. This probability is an important indicator in terms of customer segmentation, hence could not be avoided.

- Scaling was done using the function StandardScaler() - a function for z-scaling - wherein values are recalculated taking mean=0 and standard deviation=1. The data post scaling was as follows:

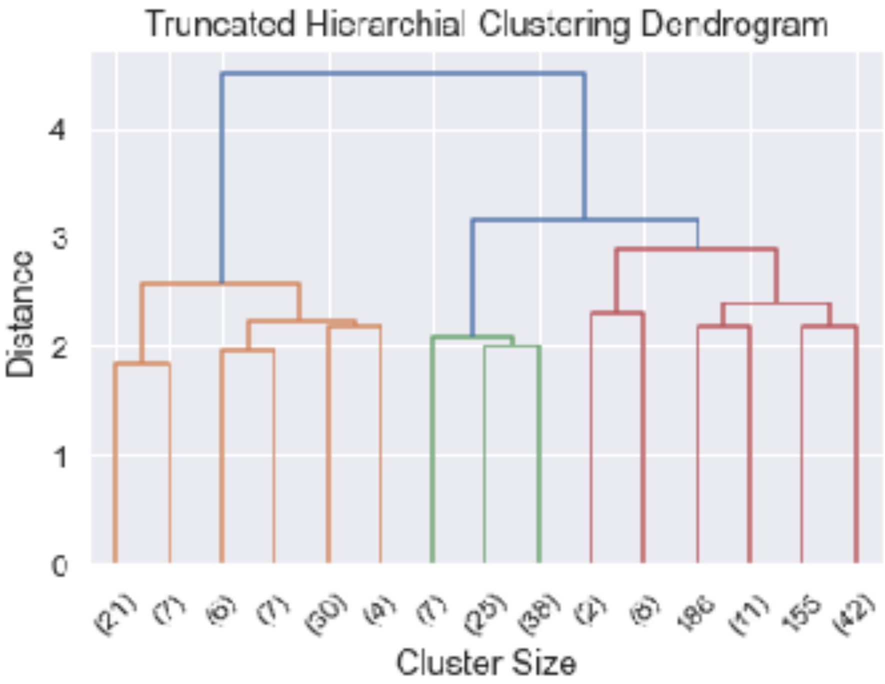
Table: 1.2.1

	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	-0.298806	2.328998
1	0.393582	0.25384	1.501773	-0.600744	0.858236	-0.242805	-0.538582
2	1.4133	1.428192	0.504874	1.401485	1.317348	-0.221471	1.509107
3	-1.384034	-1.227533	-2.591878	-0.793049	-1.639017	0.987884	-0.454961
4	1.082581	0.998364	1.19634	0.591544	1.155464	-1.088154	0.874813
...	...	...	...	...	...	...	...
205	-0.329866	-0.413929	0.721222	-0.428801	-0.158181	0.190536	-1.366631
206	0.662292	0.814152	-0.305372	0.675253	0.476084	0.813214	0.789153
207	-0.281636	-0.306472	0.364883	-0.431064	-0.152873	-1.322158	-0.830235
208	0.438367	0.338271	1.230277	0.182048	0.600814	-0.953484	0.071238
209	0.248893	0.453403	-0.776248	0.659416	-0.073258	-0.706813	0.960473

1.3 Hierarchical clustering of the scaled data.

The scaled data was subjected to hierarchical clustering. A dendrogram was used to identify the optimum number of clusters (as shown below).

Graph: 1.3.1



Although the dendrogram points to 3 distinct clusters, the fcluster() function was used to try clustering the data into 3 as well as 4 clusters. When splitting into 4 clusters, the data distinction between two of the clusters was minuscule, so it was optimum to go with 3 clusters. The clusters formed using fcluster() is shown in the table given below, along with the mean values of variables (their characteristics), as depicted in each column.

Table: 1.3.1

clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
1	18.1292	16.058	0.881595	6.135747	3.64812	3.6502	5.98704	75
2	11.916857	13.291	0.846845	5.2583	2.846	4.619	5.115071	70
3	14.217077	14.195846	0.884869	5.442	3.253508	2.759007	5.055569	65

Clustering was done again using the Agglomerative clustering technique to cross-check. It yielded the exact same results, as shown below:

Table: 1.3.2

Agglo_Clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
0	14.217077	14.195846	0.884869	5.442	3.253508	2.768418	5.055569	65
1	18.1292	16.058	0.881595	6.135747	3.64812	3.6502	5.98704	75
2	11.916857	13.291	0.846766	5.2583	2.846	4.619	5.115071	70

The 3 clusters formed can be summarized in the following words:

- **Cluster-1:** (75 records) These are the highest spenders in the sample data - spending over Rs.18000 per month on purchases, maintaining approx. Rs. 6,000 balance in their accounts. They have highest limits on their credit cards (approx. Rs.36,500) and are used to spending the highest on a single purchase (almost Rs.6000)
- **Cluster-2:** (70 records) these customers exhibit the lowest spending trend, on an average spending just about Rs.12000 on a monthly basis. They also have the lowest probability of making a full payment of bills (84.7%). Despite holding the least balance in their accounts (approx. Rs.5251), they tend to spend

almost an equivalent amount on a single shopping spree (approx. Rs.5115). Surprisingly, they are also the most probable customers to pay the highest percentage of monthly spending as advance payments (almost 11.15%).

- **Cluster-3:** (65 records) These are the medium spenders in the ample data, spending close to Rs. 14,218 on an average every month. They are good customers as they pay almost 10% of the monthly spending (approx. Rs.1420) in advance. Surprisingly, they have the highest probability of making full payments against bills (approx. 88.49%).

1.4 K-Means clustering on scaled data:

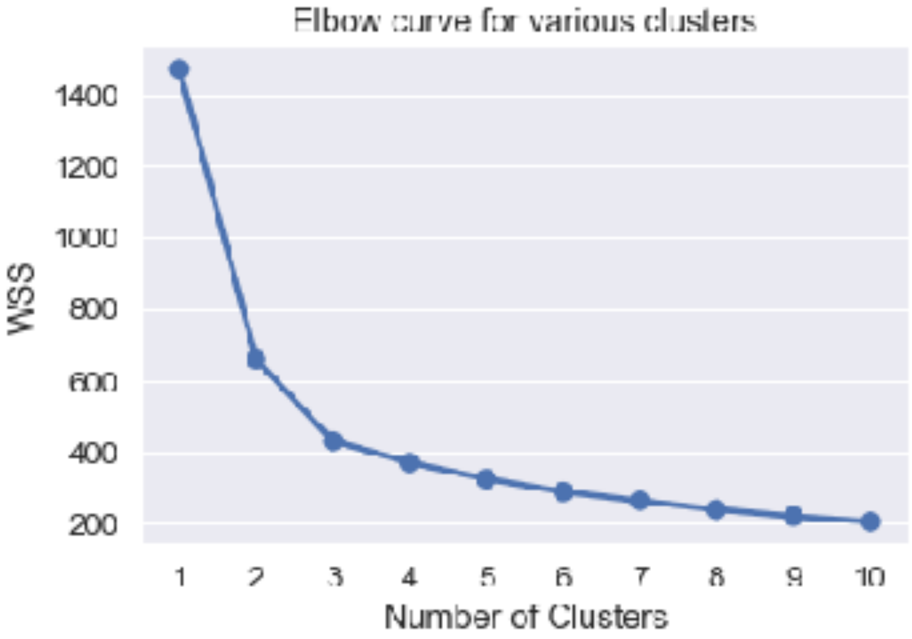
The K-Means clustering technique was used to further determine the optimum number of clusters from the sample data. K-Means algorithm was used on the already scaled data, and the WSS (Within Sum of Square values) were determined for clusters 1-10. The table below depicts the WSS values for various clusters.

Table: 1.4.1

Clusters	wss
1	1470
2	659.171754
3	430.658973
4	371.301721
5	327.960824
6	290.590031
7	264.831531
8	240.683726
9	220.852858
10	206.38291

Based on the WSS values, an elbow curve was plotted using ‘point plot()’ function in seaborn to check the drop of values, as shown below:

Graph: 1.4.1



The elbow curve is steep till the 3rd cluster, and thereafter the slope is mostly steady. This indicates that 3 clusters are optimum for the given sample of data. To get further clarity, the ‘Silhouette Score’ method was utilized to compare between the silhouette scores for 3 clusters and 4 clusters. The results were as given below:

Table: 1.4.2

	3 clusters	4 clusters
Silhouette score	**0.400805**	**0.337366**

Here, the silhouette score for 3 clusters is greater than for 4 clusters, therefore it is decisively proved that 3 clusters are optimum in this sample data.

Further, the data was grouped into 3 clusters. The clusters were profiled with the mean of the variables for each cluster and the frequency of records per cluster, which is depicted in the table below:

Table: 1.4.3

Clus_kmeans_3	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	freq
0	11.856944	13.247778	0.84833	5.23175	2.849542	4.733892	5.101722	72
1	18.495373	16.203433	0.88421	6.175687	3.697537	3.632373	6.041701	67
2	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803	71

1.5 Cluster profiles and promotional strategies for different clusters:

We can now conclusively talk about the characteristics of the 3 final clusters, corroborated by the hierarchical as well as the K-Means clustering techniques. There are minor discrepancies in the cluster frequencies as compared to the hierarchical clustering techniques, but they are inconsequential compared to the large data set. The cluster characteristics and certain suggestions to the bank are enlisted below:

- **Cluster-1:** These are the highest spenders, and maintain the highest balances in their accounts. however, their ratio of minimum payments to current balance is very low. Similarly, the ratio of their advance payments to their spending patterns are low. Hence bank could offer some attractive pay back schemes to encourage them to pay more and in advance.

- **Cluster-2:** These customers are low spenders, but pay the highest minimum payment amongst all the clusters (high risk). They also have a moderate max spending on a single purchase, but their credit limit is lowest among the 3 clusters. The bank may consider increasing their credit limit, or pitch new credit card schemes, in order to motivate them to spend more. Also bank can offer them some cash back scheme so that they continue to maintain high payment amounts, and to encourage a full payment.
- **Cluster-3:** These customers are medium spenders, but show highest probability of full payment (low risk), despite the fact that their average minimum payment amounts are low. For these customers bank may consider offering higher credit limit to encourage greater spending, as well as offer some attractive payment options or rebates to encourage higher payback.

PROBLEM 2: CART-RF-ANN

Executive summary:

An Insurance firm providing tour insurance is facing higher claim frequency. The management wants to be able to predict claim generation in an efficient manner. So, based on data from the past few years and with the help of Classificaton models namely Decision Trees (CART), Random Forest (RF) and Neural Networks (NN), a dependable model for claim prediction needs to be worked out.

2.1 Exploratory data analysis

The sample data consists of entries spread across 3000 rows and 10 columns.

Of these, the column ‘Claimed’ is the target variable, while the rest of the columns contain the prediction variables. The first 5 rows of the dataset are given below:

Table: 2.1.1

	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

The command info() gave an insight into the datatypes contained in the dataset. The variables belonging to the different data types are as follows:

- Integer/Float type (numeric): ’Age’, ‘Duration’, ‘Commission’ and ‘Sales’
- Object type (nominal): ‘Agency_code’, ‘Type’, ‘Claimed’, ‘Channel’, ‘Product name’ and ‘Destination’

Further exploration showed that there are no null values in the dataset.

On checking the unique values in each variable, we found that several variables have very large number of unique values. The number of unique values per variable is given below:

- Age - 70
- Agency_Code - 4
- Type - 2
- Claimed - 2
- Commission - 324
- Channel - 2
- Duration - 257
- Sales - 380
- Product Name - 5
- Destination - 3

The describe() command was used to derive the statistical summary of the dataset. The result is given in the following table:

Table: 2.1.2

	Age	Agency_Code	Type	Claimed	Commission	Channel	Duration	Sales	Product Name	Destination
count	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000
unique	NaN	4	2	2	NaN	2	NaN	NaN	5	3
top	NaN	EPX	Travel Agency	No	NaN	Online	NaN	NaN	Customised Plan	ASIA
freq	NaN	1365	1837	2076	NaN	2954	NaN	NaN	1136	2465
mean	38.091	NaN	NaN	NaN	14.529203	NaN	70.001333	60.249913	NaN	NaN
std	10.463518	NaN	NaN	NaN	25.481455	NaN	134.053313	70.733954	NaN	NaN
min	8	NaN	NaN	NaN	0	NaN	-1	0	NaN	NaN
25%	32	NaN	NaN	NaN	0	NaN	11	20	NaN	NaN
50%	36	NaN	NaN	NaN	4.63	NaN	26.5	33	NaN	NaN
75%	42	NaN	NaN	NaN	17.235	NaN	63	69	NaN	NaN
max	84	NaN	NaN	NaN	210.21	NaN	4580	539	NaN	NaN

Observations based on the statistical summary:

- The average age of insurance claimants is round 36-38 years.
- Majority of the data is from the agency with code EPX (1365 out of 3000).
- Out of the 3000 entries, majority have not claimed insurance (2076 out of 3000)
- Variables like ‘Commission’, ‘Duration’ and ‘Sales’ show high level of skewness, since their range is very high and their standard deviations are also high.

- Out of the three destinations, majority data belongs to ‘Asia’ destination (2465 out of 3000)
- Of the five different insurance products, the ‘Customised Plan’ is most popular (1136 out of 3000).

Using the skew() command to check skewness of the numerical variable, the following was observed. Variation is highest within the variable ‘Duration’.

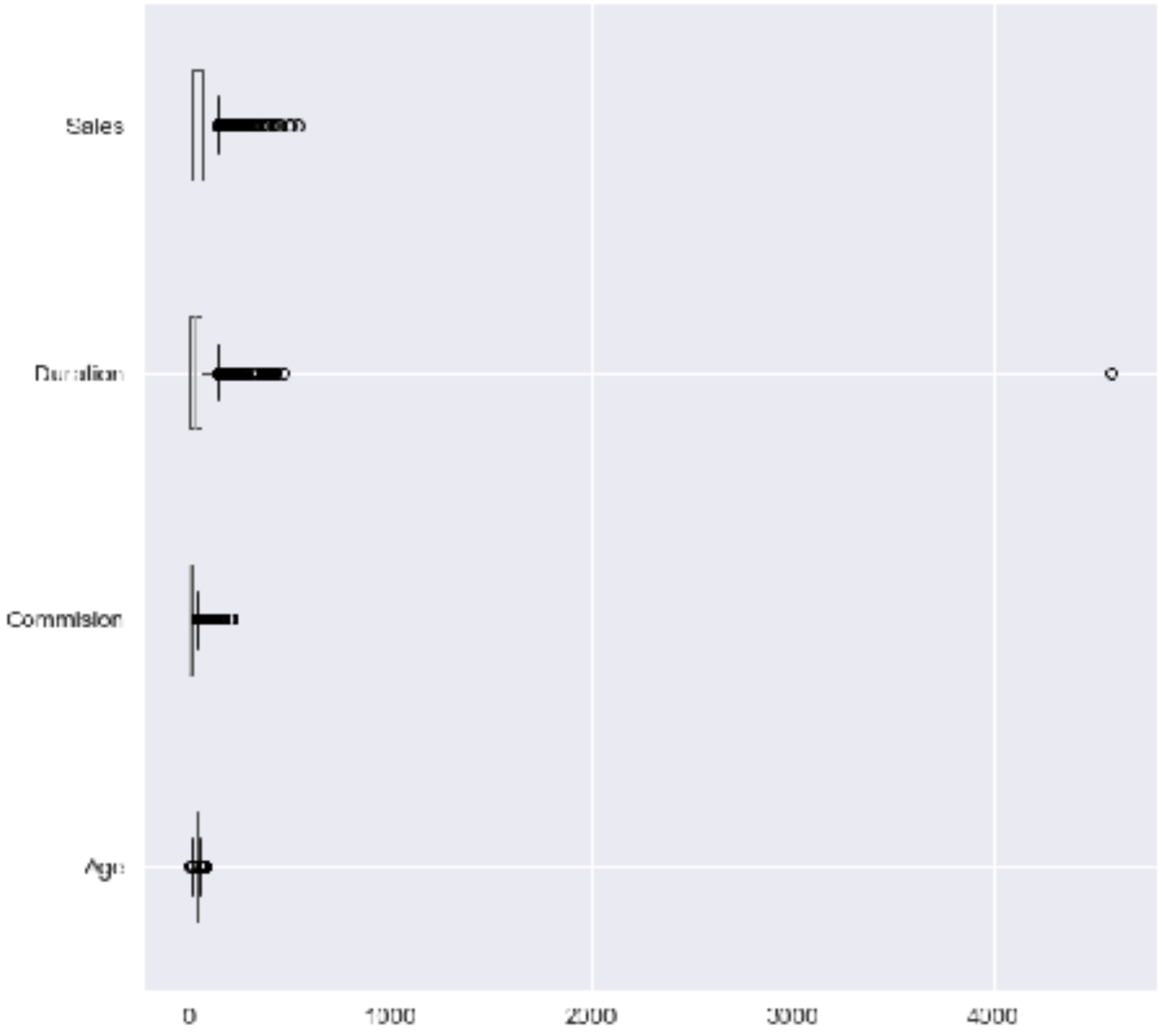
Table: 2.1.3

Variables	Skewness
Age	1.149713
Commision	3.148858
Duration	13.784681
Sales	2.381148

On checking the data for duplicates, it was revealed that the dataset contains 139 duplicate rows. However, this dataset does not contain a unique identifier for each record, so it may very well be possible that the same policy package has been sold to various customers who belong to the same demography. Hence, it is better to retain the rows in question.

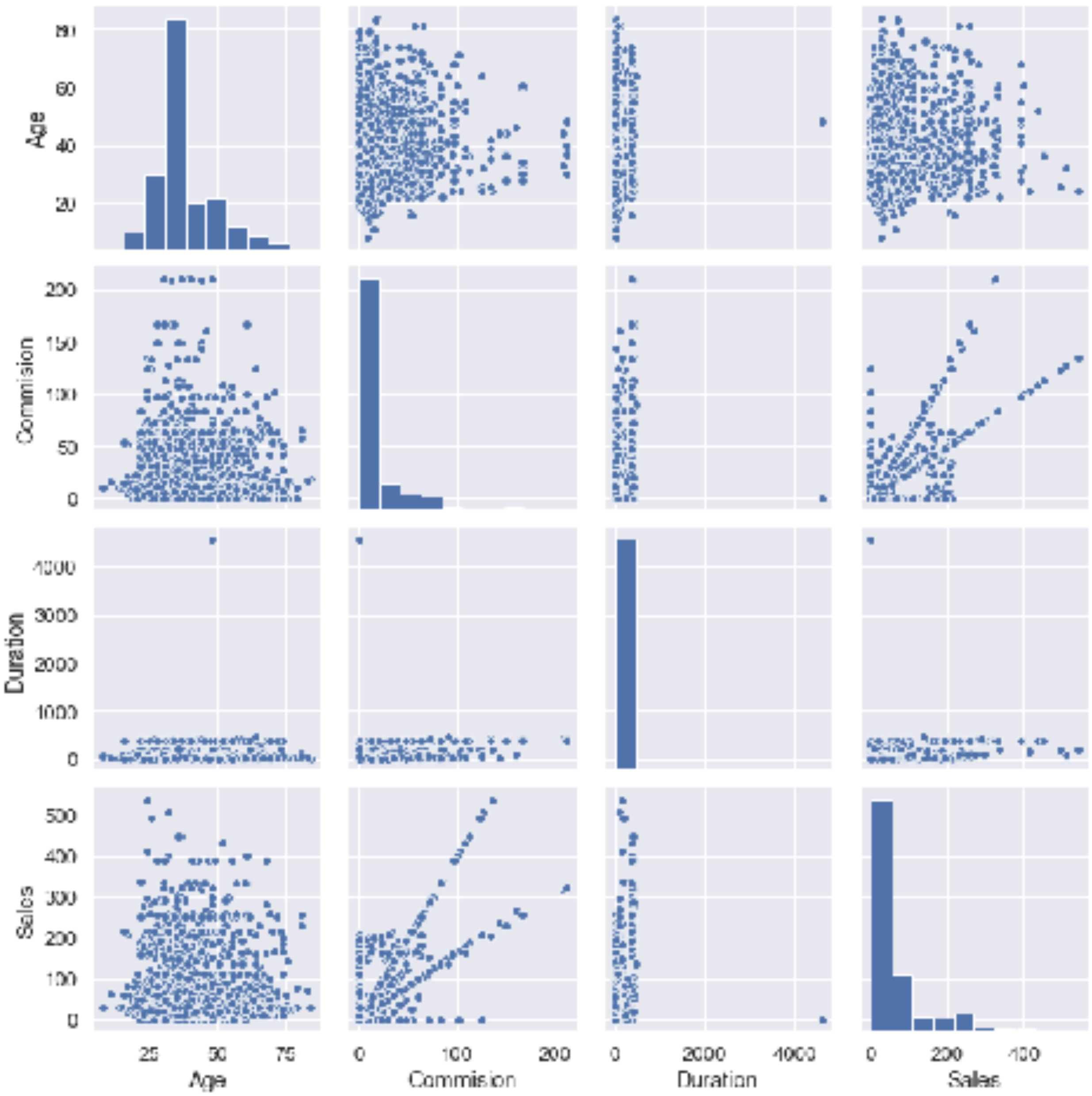
A box plot was created to check for outliers (as shown below). It revealed the presence of several outliers in the numeric variables. However, since CART, Random Forest and ANN are robust to outliers, it need not be treated.

Figure: 2.1.1



On checking the pairwise distribution for the continuous variables using a pairplot, the following was obtained:

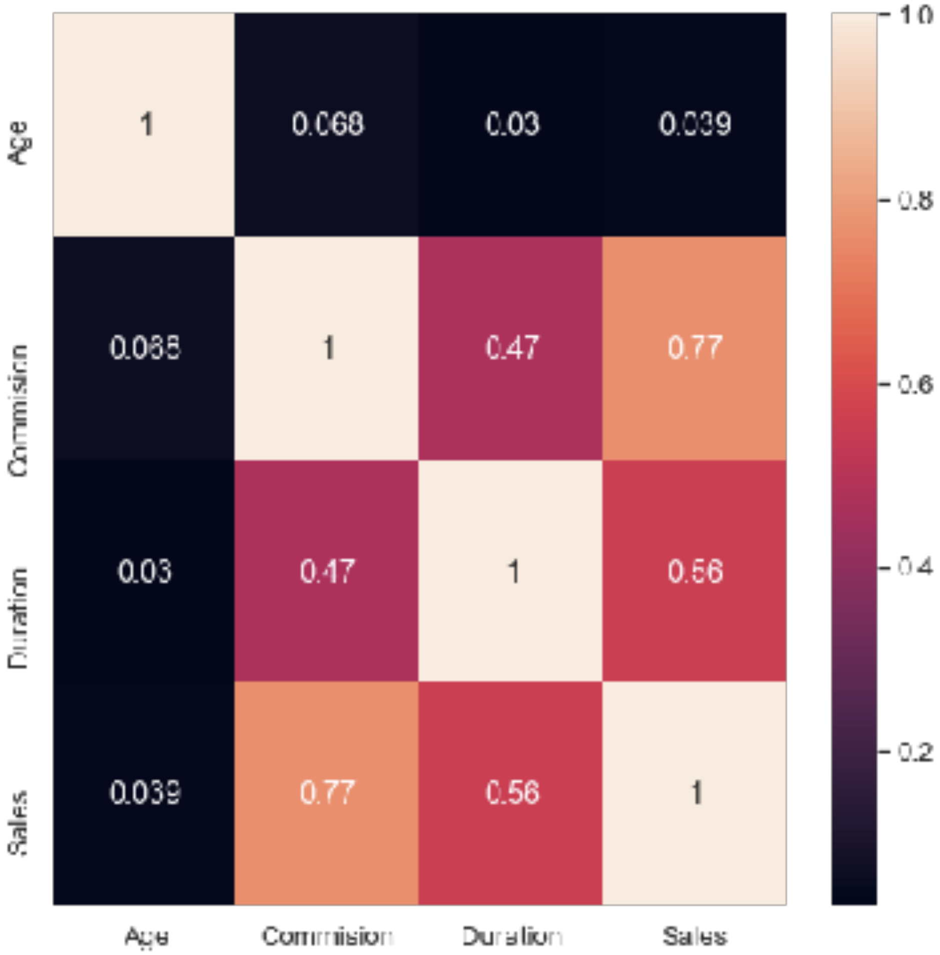
Figure: 2.1.2



No substantial correlation is evident in the graph, although some correlation between the variables ‘Commission’ and ‘Sales’ can be seen.

To corroborate, a heat map was created, as follows - it proves that there is no substantial correlation between the numerical variables.

Figure: 2.1.3



The heat map decisively confirmed that no strong correlation exist between the variables.

Considering that 6 out of the 10 variables in the dataset are of ‘object’ type, it was necessary to convert them into binary codes in order to enable error-free processing of Neural networks, Random forest and decision trees.

Post converting the variables to codes, the dataset displayed codes as given below:

Table: 2.1.4

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	0	0	0	0.7	1	7	2.51	2	0
1	36	2	1	0	0	1	34	20	2	0
2	39	1	1	0	5.94	1	3	9.9	2	1
3	36	2	1	0	0	1	4	26	1	0
4	33	3	0	0	6.3	1	53	18	0	0

On checking the proportion of the distribution of the target variable (Claimed) - there is no gross imbalance in the distribution, as follows:

- No (code-0) - 69.2%
- Yes (code-1) - 30.8%

2.2. Split the data into test and train sets:

From the original dataset, the variable ‘Claimed’ was dropped to create 2 different vectors so that data could be split into Training and Testing sets.

- Vector x —> all variables excluding ‘Claimed’
- Vector y —> variable ‘Claimed’

The data was then split into Training data (70%) and Testing data (30%). The data now was in 4 parts to enable the classification on the basis of various models, as follows:

Table: 2.2.1

Data set	Shape
x_train	**(2100, 9)**
x_test	**(900, 9)**
y_train / train_labels	**(2100,)**
y_test / test_labels	**(900,)**

2.3 Performance Metrics for all 3 models:

Post splitting the data, it was fitted into the 3 models - CART , random forest and Neural networks.

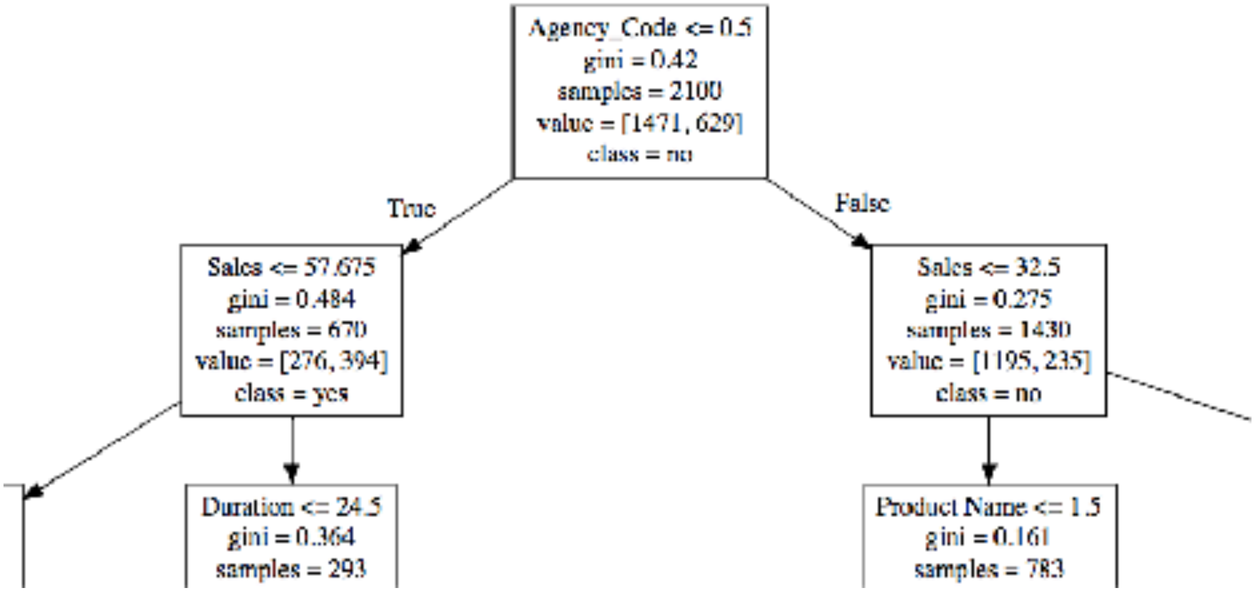
Decision Tree (CART) model:

A Decision Tree was built using the ‘gini’ criterion. The parameters for the model were as follows:

- Maximum depth - 10
- Minimum sample leaf - 50
- Minimum samples split - 150

The codes for the CART model was saved in a .dot file (tree_regularized.dot). These codes were used to generate the actual decision tree on the website www.webgraphviz.com. The root node of the tree and initial few nodes are given below:

Figure: 2.2.1



It is very evident from the Decision Tree that the variable ‘Agency_Code’ is the most important variable in terms of predicting claims from customers. It is followed by ‘Sales’, ‘Product name’ and ‘Duration’.

To ascertain the exact contribution of the variables to the prediction model, their feature importance was calculated - the results are shown below:

Table: 2.2.2

Feature	Importance
Agency_code	0.599363
Sales	0.255785
Product Name	0.056555
Duration	0.037945
Age	0.030261
Commision	0.012676
Type	0.007416
Channel	0
Destination	0

The performance metrics of the CART model are shared below:

Table: 2.2.3

Metric	CART Train	CART Test
Accuracy	0.79	0.75
AUC	0.84	0.79
Recall	0.5	0.38
Precision	0.72	0.73
F1 Score	0.59	0.5

Random Forest Classifier:

A random forest classifier was built with the following parameters:

- Maximum depth - 20
- Maximum features - 8
- Minimum sample leaf - 50
- Minimum sample split - 50
- Number of estimators - 100

The feature importances as per the Random Forest (RF) model are given below:

Table: 2.2.4

Feature	Importance
Agency_Code	0.529208
Sales	0.205338
Product Name	0.132938
Commision	0.052595
Duration	0.040586
Age	0.030295
Type	0.00689
Destination	0.002149
Channel	0

The performance metrics of the RF model are as given below:

Table: 2.2.5

Metrics	RF Train	RF Test
Accuracy	0.8	0.76
AUC	0.84	0.82
Recall	0.56	0.45
Precision	0.71	0.73
F1 Score	0.63	0.55

Neural Network:

A neural network (NN) was built for classification, the optimum parameters chosen by the machine were as follows:

- Hidden layers - 100
- Maximum iteration - 2500
- Solver type - ‘adam’
- Tolerance level - 0.01

The performance metrics of the NN model are shared below:

Table:2.2.6

Metrics	NN Train	NN Test
Accuracy	0.79	0.76
AUC	0.82	0.78
Recall	0.58	0.48
Precision	0.67	0.71
F1 Score	0.62	0.57

2.4 Comparison of all the models:

To conclude, performance of all 3 models was evaluated and compared to derive at the optimum model for claim prediction.

The ROC curves of the 3 models for training and testing data are given below:

Figure: 2.4.1

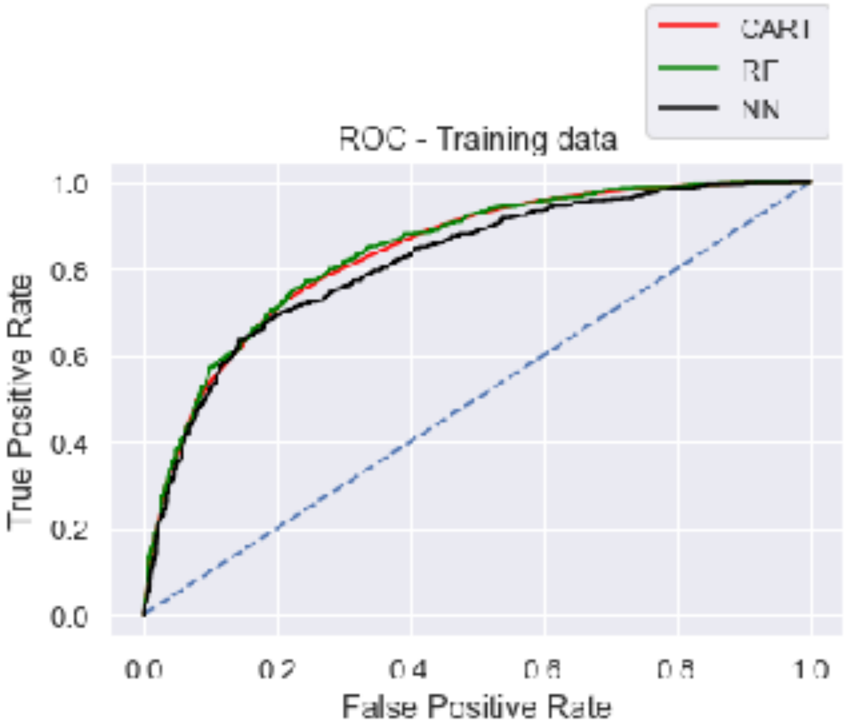
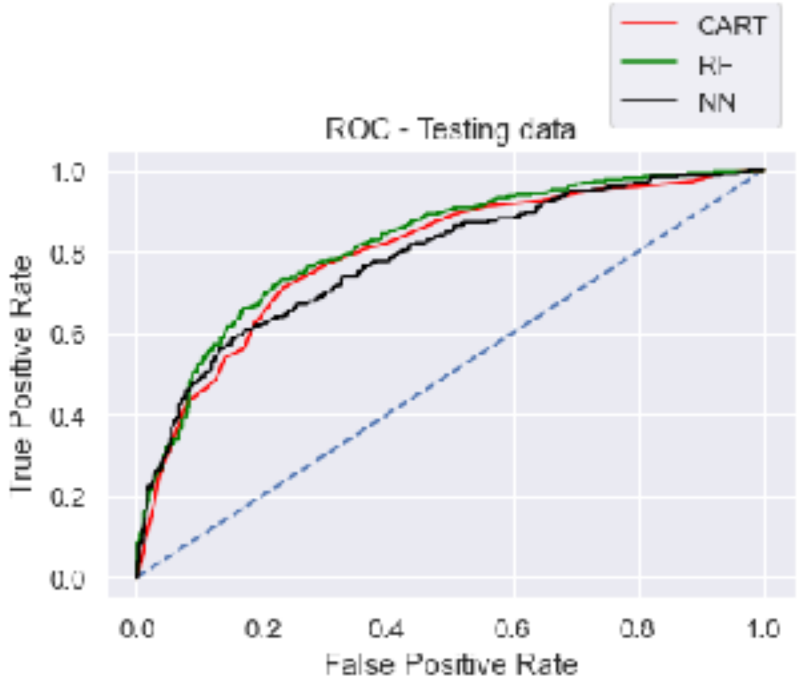


Figure: 2.4.2



The comparison of the evaluation metrics for the 3 models are summarized in the table below:

Table: 2.4.1

Metrics	CART Train	CART Test	Random Forest Train	Random Forest Test	Neural Network Train	Neural Network Test
Accuracy	0.79	0.75	0.8	0.76	0.79	0.76
AUC	0.84	0.79	0.84	0.82	0.82	0.78
Recall	0.5	0.38	0.56	0.45	0.58	0.48
Precision	0.72	0.73	0.71	0.73	0.67	0.71
F1 Score	0.59	0.5	0.63	0.55	0.62	0.57

Based on both the above ROC curves, we can decisively say that the best model for claims prediction will be Random Forest classification, since in both cases the steepest curve is formed by the RF model, as well as the more area is covered by it.

Furthermore, to bring more clarity, the comparison of metrics show that though all 3 models yield moderate efficacy, the RF model shows consistently higher matrices. Thus, it is beyond doubt that the RF model has performed better than the other 2 models.

2.5 Inference, business insights and recommendations:

- It will be in the best interest of the Insurance company to predict claims using the Random Forest classification model.
- The company needs to pay close attention to the Agency_code. Highest incidents of claims occur in the agencies under code C2B. The company must study/audit the agency procedures and try to understand the reasons behind it. Further investigation may be carried out on the demography that these agencies are catering to.
- After agency code, the next variable that needs attention is sales. Those agencies with sales less than 57.5 show greater tendency for claims. The company needs to employ strategies to increase sales so that their claims losses are offset through higher sales margins.