DATA MINING

NAME: **ANISH DASGUPTA**

COURSE: **DSBA**

BATCH: **May'21**

TABLE OF CONTENTS:

DATA DICTIONARY:

Term/Abbr.	Description
spending	Amount spent by the customer per month (in 1000s)
advance_payments	Amount paid by the customer in advance by cash (in 100s)
probability_of_full_payment	Probability of payment done in full by the customer to the bank
current_balance	Balance amount left in the account to make purchases (in 1000s)
credit_limit	Limit of the amount in credit card (10000s)
min_payment_amt	Minimum paid by the customer while making payments for purchases made monthly (in 100s)
max_spent_in_single_shopping	Maximum amount spent in one purchase (in 1000s)

Claimed	Target: Claim Status
Agency_Code	Code of tour firm
Type	Type of tour insurance firms
Channel	Distribution channel of tour insurance agencies
Product	Name of the tour insurance products
Duration in days	Duration of the tour
Destination	Destination of the tour
Sales	Amount worth of sales per customer in procuring tour insurance policies in rupees (in 100's)
Commission	The commission received for tour insurance firm (Commission is in percentage of sales)
Age	Age of insured

LIST OF FIGURES:

LIST OF TABLES:

OBJECTIVE:

The primary objective of this report is to provide the business implications of two presented problem statements. The insights provided in this report primarily analyze the problem under hand and attempts to answer the question that follows it. The codes for deriving those insights are maintained separately.

Problem Statement 1:

INTRODUCTION:

The first problem statement talks about 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. Given task is to identify the segments based on credit card usage.

1.1 Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis).

Data definition:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	mir
0	19.94	16.92	0.8752	6.675	3.763	
1	15.99	14.89	0.9064	5.363	3.582	
2	18.95	16.42	0.8829	6.248	3.755	

Fig. 1.1.1 : Data definition

Inference:

- We see that there are 7 columns which are being considered as features for this dataset.

Data Info:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Nu
---  ------                        ------
 0   spending                      210 no
 1   advance_payments              210 no
 2   probability_of_full_payment   210 no
 3   current_balance               210 no
 4   credit_limit                  210 no
```

Fig. 1.1.2: Data Info

Inference:

- The features describe the banking details of 210 customers.
- We can see that there are no null values in the dataset which means all the columns for each customer is filled properly, as well as, all the values are float(numerical/continuous).We do not have any categorical columns.

Data Describe:

	count	mean	std	min	
spending	210.0	14.847524	2.909699	10.5900	12.2
advance_payments	210.0	14.559286	1.305959	12.4100	13.4
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.8
current_balance	210.0	5.628533	0.443063	4.8990	5.2

Fig. 1.1.3: Data describe

Inference:

- We can see that the data is not scaled as different features have values in different ranges/scales.
- The data set has data for which the mean and the modes are close to each other for some of the features. Hence we can deduce that the features credit_limit,probability_of_full_payment have an almost normal distribution.

Checking for Duplicate Data:

```
spending
advance_payments
probability_of_full_paym
current_balance
credit_limit
min_payment_amt
```

Fig. 1.1.4: Duplicate Data

Inference:

- The data set does not have any duplicate values. So we can infer from this that all the data is provided for distinct customers i.e. there are no customers repeated in the data set.

Detecting Outliers, Skewness and Kurtosis:

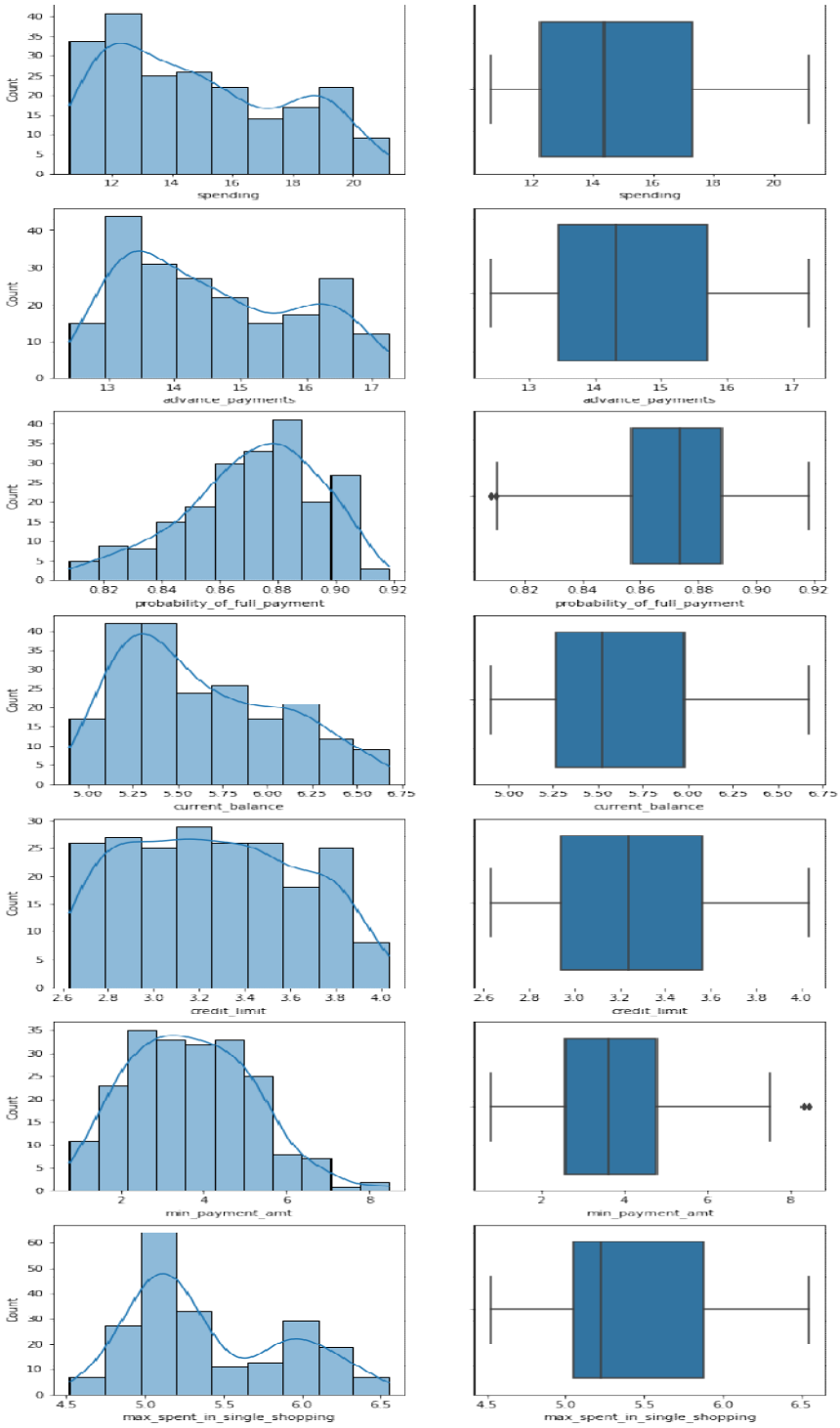


Fig. 1.1.5: Outliers,Skewness,Kurtosis

Inference:

- From the boxplots of 'min_payment_amt','probability_of_full_payment', we can observe that these two features contain outliers. The 'min_payment_amt' has outliers on the upper side where as 'probability_of_full_payment' has outliers on the lower side and very close to the minima, which might be the reason of the respective skewness in each column values.
- For 'probability_of_full_payment', we have got 3 outliers on the lower side whereas for 'min_payment_amt' , we have got 2 outliers on the upper side.
- From the hist plots, we can see that most of the features are right skewed by a heavy degree, except 'probability_of_full_payment' which is left skewed.
- From the skewness and kurtosis calculation, we see that for 'spending','advance_payments' and 'credit_limit', the kurtosis is beyond the ideal value of +1 to -1. The skewness for all the features are in the ideal range. 'credit_limit' has the least amount of skewness.

Outlier Value Record:

min_payment_amt -

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	mi
5	12.7	13.41	0.8874	5.183	3.091	

probability_of_full_payment -

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	mi
3	10.83	12.96	0.8099	5.278	2.641	
77	12.13	13.73	0.8081	5.394	2.745	

Correlation Heatmap:

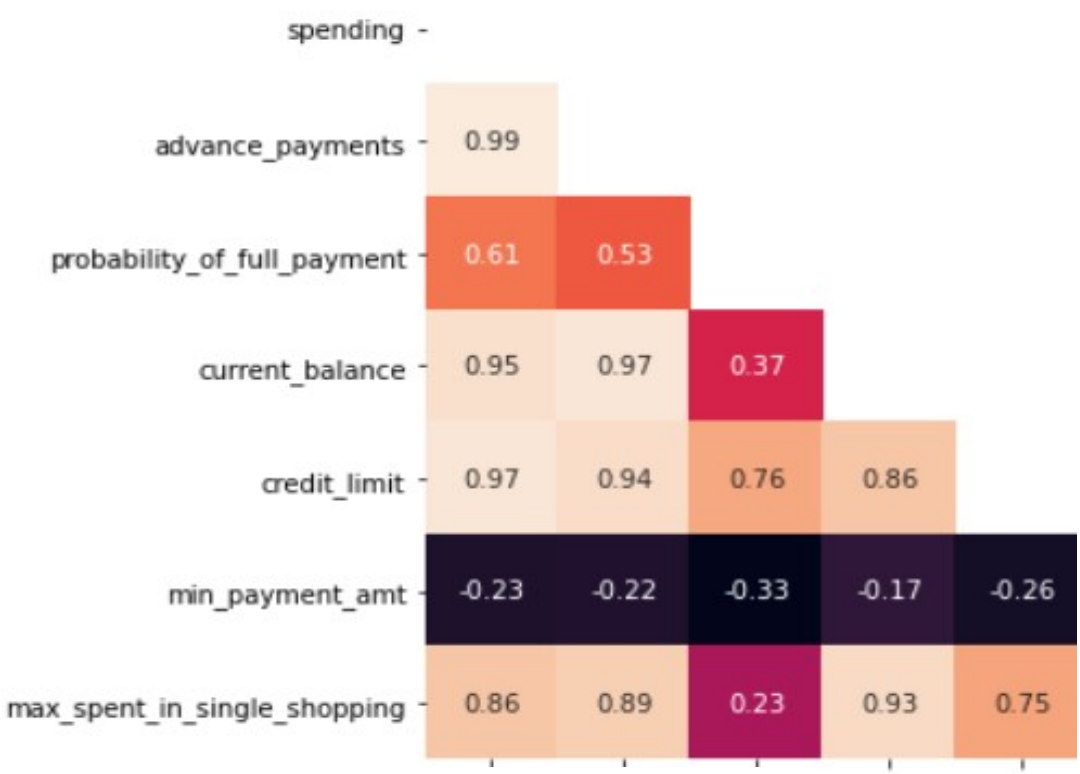


Fig. 1.1.6: Correlation Heatmap

Inference:

- From the correlation table, we see that most of the features are highly correlated with each other, except 'min_payment_amt', which is negatively correlated with the rest of the features.
- From the bivariate analysis using the pairplot, we observe that:

    - ❖ spending, advance_payments : extremely correlated
    - ❖ spending,current_balance : highly correlated
    - ❖ spending,credit_limit : highly correlated
    - ❖ advance_payments,current_balance : highly correlated
    - ❖ advance_payments,credit_limit : highly correlated
    - ❖ probability_of_full_payment,credit_limit : moderately correlated
    - ❖ probability_of_full_payment,min_payment_amt : negatively correlated
    - ❖ min_payment_amt,rest of features : low/negative correlation

Scatter Plot for Bivariate Analysis:

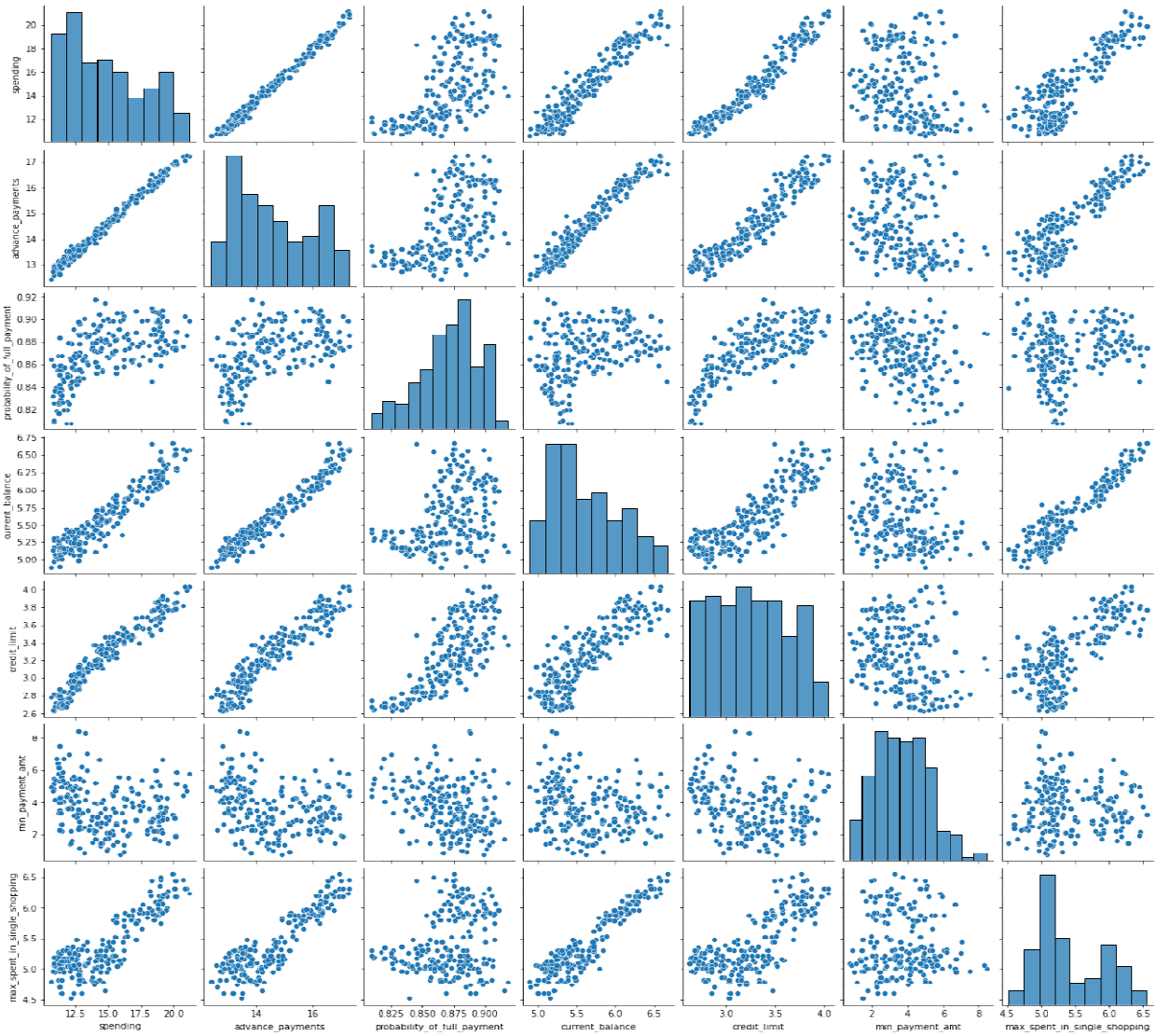


Fig. 1.1.7: Pairplot

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

	count	mean	std	min	
spending	210.0	14.847524	2.909699	10.5900	12.2
advance_payments	210.0	14.559286	1.305959	12.4100	13.4
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.8
current_balance	210.0	5.628533	0.443063	4.8990	5.2
credit_limit	210.0	3.258605	0.377714	2.6300	2.9

Fig. 1.2.1: Data Description

Inference:

- We can see that different measures such as mean, median for 'spending', 'advance_payments' are in the ranges of 10s whereas the same measures for 'current_balance','credit_limit','min_payment_amt','max_spent_in_single_shopping' are in the range of 1s and for 'probability_of_full_payment', it is in 0.1s.
- So we can see that mean,medians,standard deviations,max,min for these features are in different ranges and scales from 0.1s to 10s.
- So it is necessary for us to use scaling for this data set. Scaling is required since the different scale values are going to affect the euclidean distances between the different observations which might negatively affect the clustering process and give ineffective results.

We will be using the StandardScaler method for standardisation process. This method is going to scale the data for all the given features and scales the data such that the mean of all the features moves near to 0 and the standard deviation tends to 1.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min
0	1.754355	1.811968	0.178230	2.367533	1.338579	
1	0.393582	0.253840	1.501773	-0.600744	0.858236	
2	1.413300	1.428192	0.504874	1.401485	1.317348	

Fig. 1.2.2: Data after Scaling

	count	mean	std	min	
spending	210.0	9.148766e-16	1.002389	-1.466714	-0.8
advance_payments	210.0	1.097006e-16	1.002389	-1.649686	-0.8
probability_of_full_payment	210.0	1.260896e-15	1.002389	-2.668236	-0.5
current_balance	210.0	-1.358702e-16	1.002389	-1.650501	-0.8
credit_limit	210.0	-2.790757e-16	1.002389	-1.668209	-0.8

Fig. 1.2.3: Data Description after Scaling

As we can see, for the scaled data, the mean of all the features tends to 0 and the standard deviation tends to 1. Also, all the features now have the various measures in the same scale.

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

Fig. 1.3.1: Dendrogram

To create the dendrogram, we have used the linkage method as 'ward' which used the cluster variance amongst the observations to cluster.

For the above dendrogram, we have cut the tree using the lastp method which shows the dendrogram with p number of split counts. The dendrogram shows the last 10 merges which

gives us a clear picture of the splits and their counts as well. From the dendrogram we can see that there are 3 distinct clusters, namely blue, orange and green.

Similar records are joined by lines whose vertical length reflects the distance between the records. From the dendrogram, we can see that for the orange cluster, the records are comparitively closer than those in the green cluster.

The difference in height between the green and the orange cluster merges are significantly large depicting that the inter cluster distance is significant which is a favourable requirement.

Creating Clusters using fcluster method:

```
array([1, 3, 1, 2, 1, 2, 2, 3, 1, 2, 1, 3, 2, 1,
       1, 2, 3, 1, 3, 2, 2, 2, 3, 2, 2, 3, 2, 2,
       2, 2, 3, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 2,
       1, 3, 1, 2, 3, 2, 1, 1, 2, 1, 3, 2, 1, 3,
       1, 2, 3, 1, 3, 2, 2, 1, 1, 1, 2, 1, 2, 1,
       3, 3, 1, 2, 2, 1, 3, 3, 2, 1, 3, 2, 2, 2,
       3, 1, 2, 1, 1, 2, 1, 3, 3, 3, 2, 2, 3, 2,
```

We have used the 'maxclust' criterion for creating the clusters which finds a minimum threshold so that the cophenetic distance between any two original observations in the same flat cluster is no more than the threshold and no more than 3 flat clusters are formed.

We can see that the 'fcluster' method has alloted all the observations into 3 different clusters on the basis of the feature values.

Alternatively using Agglomerative clustering method;

Creating Clusters using Agglomerative method:

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

We have used affinity as Euclidean which is a measure to compute the linkage.

ıg	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spe
94	16.92	0.8752	6.675	3.763	3.252	
99	14.89	0.9064	5.363	3.582	3.336	
95	16.42	0.8829	6.248	3.755	3.368	
33	12.96	0.8099	5.278	2.641	5.182	
99	15.86	0.8992	5.890	3.694	2.068	
70	13.41	0.8874	5.183	3.091	8.456	

Fig. 1.3.2: Cluster columns appended to original data set

From the agglomerative clustering, we see a similar clustering distribution as that from the fcluster.

1.4 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.

Clustering using k value = 2:

```
array([1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0,
       1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0,
       1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1,
       1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
```

Clustering using k value = 3:

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

Clustering using k value = 4:

```
array([3, 1, 3, 0, 3, 0, 0, 1, 3, 0, 3, 1, 0, 3,
       3, 0, 1, 2, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0,
       0, 0, 1, 3, 3, 3, 0, 3, 3, 3, 3, 2, 0, 0,
       3, 1, 3, 1, 1, 0, 3, 3, 0, 3, 1, 0, 2, 1,
       2, 0, 1, 3, 1, 0, 1, 3, 3, 2, 0, 2, 1, 3,
       2, 1, 3, 0, 0, 2, 1, 1, 0, 3, 1, 0, 0, 0,
       1, 3, 0, 3, 3, 0, 2, 1, 2, 1, 0, 0, 1, 0,
```

Clustering using k value = 5:

```
array([3, 1, 3, 0, 3, 4, 0, 1, 3, 0, 3, 1, 0, 3,
       3, 0, 1, 2, 1, 0, 0, 4, 1, 0, 4, 1, 0, 0,
       0, 4, 1, 3, 3, 3, 0, 3, 3, 3, 3, 2, 4, 4,
       3, 1, 3, 4, 1, 4, 3, 3, 4, 3, 1, 0, 2, 1,
       2, 4, 1, 3, 1, 0, 4, 3, 3, 2, 0, 2, 4, 3,
       2, 1, 3, 4, 0, 2, 1, 4, 4, 3, 1, 0, 4, 0,
       1, 3, 0, 2, 3, 4, 2, 1, 2, 1, 0, 4, 1, 4,
```

WSS for k value from 1 to 10:

```
[1469.99999999999
 659.171754487041
 430.658973151300
 371.385090608010
 327.212781656613
 289.315995389594
 262.981865701622
```

Elbow Curve for k from 1 to 10:

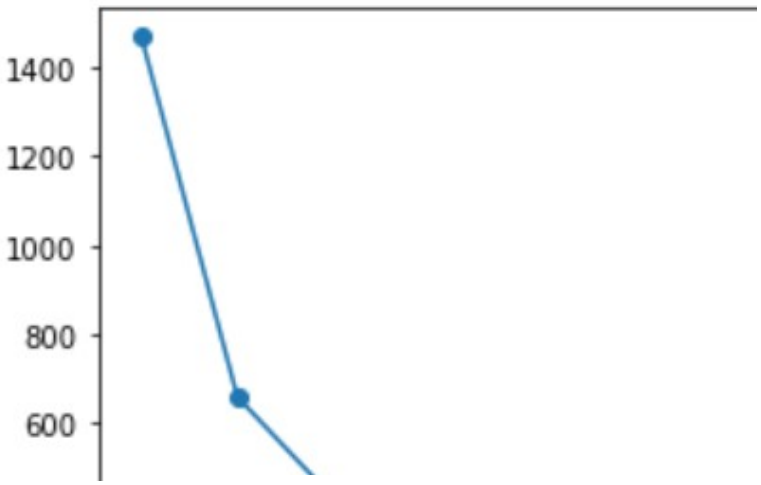


Fig. 1.4.1: Elbow Curve for 1<=k<=10

Inference:

From the WSS/Elbow plot we see that till value 3, the drop is significant, after which the drop becomes difficult to distinguish, so we ideally use k=3 for clustering.

For checking the validity of the model with 3 clusters we use Silhouette Score.

Since the minimum and maximum silhouette score for k value =3 is positive, all data are mapped correctly to the clusters and the clusters are well separated.

Cross Checking Model validity:

We calculate silhouette score for all the values of k to confirm if the model is validated for best performance with k=3.

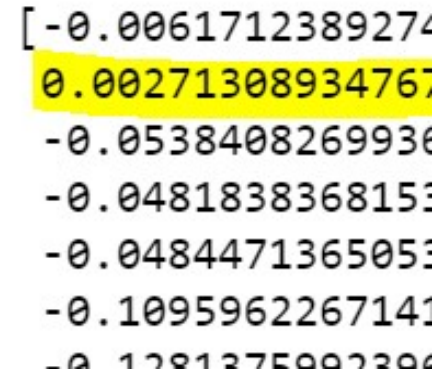


Fig. 1.4.2: Silhouette score for 2<=k<=10

We see that for except k=3, for all the values of k, the silhouette score is showing as negative which means that the clustering process hasn't been performed properly. Hence we move ahead with the k values of 3 as the number of clusters for the clustering process

bability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	clus
0.8752	6.675	3.763	3.252	6.550	
0.9064	5.363	3.582	3.336	5.144	
0.8829	6.248	3.755	3.368	6.148	

Fig. 1.4.3: K-means cluster appended to the original data set

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

Cluster profiles from f-cluster:

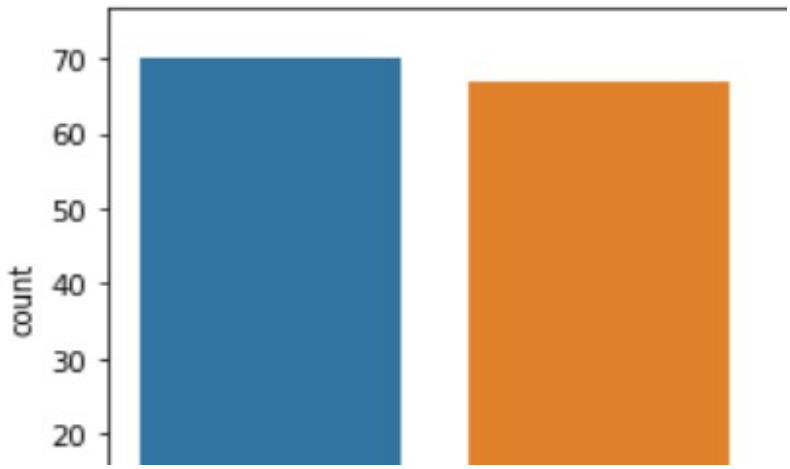
cluster_fcluster	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit
1	18.371429	16.145429	0.884400	6.158171	3.684629

Fig. 1.5.1: f-Cluster Profile with mean values

1	70
2	67
3	73

Fig. 1.5.2: f-Cluster frequency

Customer Segmentation from f-cluster:



Inference:

- We can see that the cluster 3 has the highest frequency and the maximum number of observations.
- All the clusters have almost equal number of observations.

- When we see the probability of full payment, we see that the mean value is highest for cluster 1.
- Both Agglomerative and f-cluster methods of clustering provide us a similar cluster distribution for all the observations.

Cluster profiles from k-means:

cluster_kmeans	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit
0	14.437887	14.337746	0.881597	5.514577	3.259225
1	11.856944	13.247778	0.848253	5.231750	2.849542

Fig. 1.5.3: k-means Cluster Profile with mean values

0	71
1	72
2	67

Fig. 1.5.4: k-means frequency

Customer Segmentation from k-means:

Inference:

- We can see that the cluster 1 has the highest frequency and the maximum number of observations.
- All the clusters have almost equal number of observations.
- When we see the probability of full payment, we see that the mean value is highest for cluster 2.
- Both Agglomerative and fcluster methods of clustering provide us a similar cluster distribution.

The clustering from k-means also depicts the similar observations to that of fcluster. If we look closely, the cluster from k means is very much identical to the f-cluster profile:

1. Cluster 1 from fcluster is similar to Cluster 2 from k-means.
2. Cluster 2 from fcluster is similar to Cluster 1 from k-means.
3. Cluster 3/0 from fcluster/agglomerative cluster is similar to Cluster 0 from k-means.

Final Data set with all clusters appended:

bability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shoppin
0.8752	6.675	3.763	3.252	6.55
0.9064	5.363	3.582	3.336	5.14
0.8829	6.248	3.755	3.368	6.14
0.8099	5.278	2.641	5.182	5.18
0.8992	5.890	3.694	2.068	5.83

Fig. 1.5.5: Final data set with all 3 clusters

Inference from Customer Segmentation from f-cluster:

- We can observe that customer from cluster 1 have significantly higher maximum expenditure in a single shopping, from where we can deduce that these customers are comparatively compulsive shoppers and are okay with spending a lot at one go.

- Customers belonging to cluster 3 have maximum expenditure lowest compared to the other 2, from where we can understand that these customers are extremely careful while making purchases.

- Customer from cluster 2 are moderate level shoppers.

- Customers from cluster 1 have higher amounts of current balance, which supports the point that they do not think too much before spending too much.

- The monthly spending of customers from cluster 1 is also significantly higher compared to 2 and 3. The monthly spending is the least for cluster 2 customers.

- When it comes to the minimum payment amount, it is the highest for cluster 2 customers. From this we can deduce that cluster 2 customers do spend moderately in shopping but they do not have the tendency to shop compulsively like cluster 1 customers who have the lowest minimum payment amount.

- Maybe cluster 1 customers shop a lot at a single shopping purchase whereas cluster 2 customers shop moderately but more regularly than cluster 1 customers.

Recommendation for promotional strategies:

1. From the cluster profiling, we can see that customers from cluster 3 have the least credit activities amongst all the other clusters. If we look at their balances and maximum spending, these customers probably come under a lower salary brackets and have moderate credit limit, hence they spend carefully. So in order to target these customers, the company can use promotional strategies like:

- offering the customers a higher credit limit,
- providing higher reward points for each purchase which makes them think of using the credit card more often.

In order to approach these customers, the personal interaction needs to be minimal since these customers don't depend on the credit card usage heavily. So promotional offers via calls can be a nuisance for these customers. Offers through sms/emails seem to be appropriate.

2. From the cluster 1 customers, we see that their purchase pattern behaviour works in a way where they can finish off most of their monthly purchases at one go. Since these customers buy a heavy purchase using their credit card, we can promote certain strategies to them the ways below:

- offer them increased cash backs/discount offers on a purchase amount above a certain limit.

Since these customers use credit card for their heavy purchases, providing them personalised offers via a regional finance manager allotted to them might help them motivate for heavier purchases.

3. From the cluster 2, we see that these customers make maximum of their purchases at a regular basis. So for these customers;

- if the company can provide some offers such as heavier discounts on certain portals/online shopping which have the credit card company as their partners.

These customers are already having a regular credit card usage and are used to the entire process, so promotional offers to them via their social media profiles like Facebook, Instagram, Youtube can alert them about any new offers/plans.

Problem Statement 2:

Introduction:

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

Understanding the Problem:

Due to some unforeseen circumstances, the insurance company is facing higher claim frequency than before. So the solution is to predict what are the driving factors which are making this happen.

2.1 Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis).

Data Definition:

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duratio
0	48	C2B	Airlines	No	0.70	Online	
1	36	EPX	Travel Agency	No	0.00	Online	3
2	39	CWT	Travel Agency	No	5.94	Online	

Fig. 2.1.1: Data Definition Insurance

Inference:

- We see that there are 9 columns which are being considered as features for this dataset and 1 target variable. Out of these 9 features, 4 are continuous and rest are categorical.

Data Info:

```
<class 'pandas.core.frame.Data
RangeIndex: 3000 entries, 0 to
Data columns (total 10 columns
 #   Column         Non-Null Co
---  ------         -----------
 0   Age            3000 non-nu
 1   Agency_Code    3000 non-nu
 2   Type           3000 non-nu
 3   Claimed        3000 non-nu
 4   Commision      3000 non-nu
 5   Channel        3000 non-nu
 6   Duration       3000 non-nu
```

Fig. 2.1.2: Data Info Insurance

Inference:

- The features describe the insurance details of 3000 customers.
- We can see that there are no null values in the dataset which means all the columns for each customer is filled properly.

Data Describe:

	count	mean	std	min	25%
Age	3000.0	38.091000	10.463518	8.0	32.0
Commision	3000.0	14.529203	25.481455	0.0	0.0
Duration	3000.0	70.001333	134.053313	-1.0	11.0

Fig. 2.1.3: Data Describe Insurance

Inference:

- We can see that the data is scaled as different features have values in different similar ranges/scales.
- 7. Few of the outlier values seem to be out of proportion. eg Duration has minimum value of -1 which is not possible. These need to be treated.
- The data set does has many duplicate values (139 nos). Duplicate values can increase the chances of a biased prediction, hence needs to be treated.
- Since we are supposed to remove the duplicate records, we need to reset the index.

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duratio
2856	28	CWT	Travel Agency	Yes	166.53	Online	364.
2857	35	C2B	Airlines	No	13.50	Online	5.
2858	36	EPX	Travel Agency	No	0.00	Online	54.
2859	34	C2B	Airlines	Yes	7.64	Online	39.

Fig. 2.1.4: Tail of Data set after Duplicates removal and index reset

Detect Outliers, Skewness and Kurtosis:

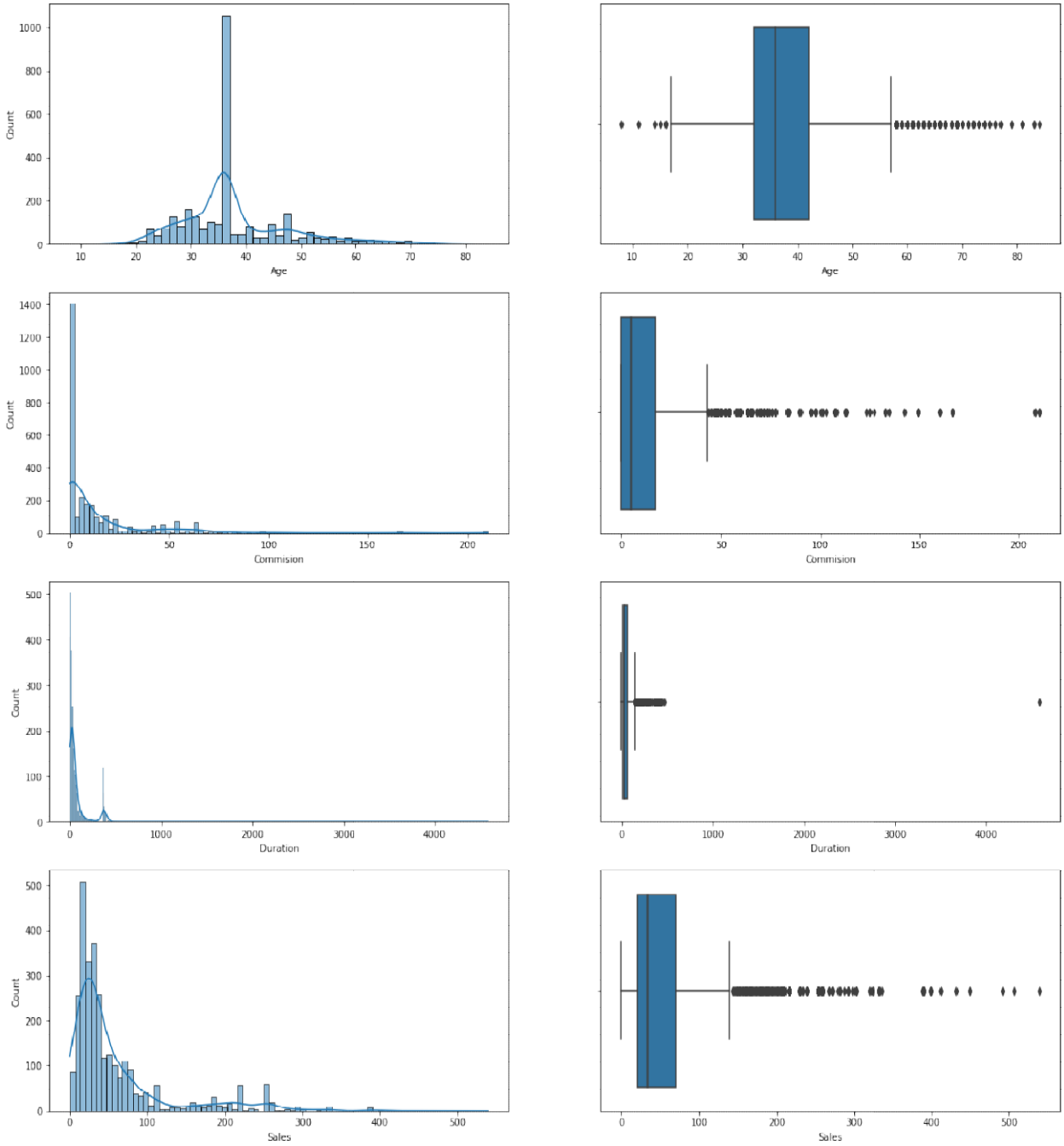


Fig. 2.1.5: Outlier, Skewness, Kurtosis for Insurance

Inference:

- From the boxplots of 'Age', 'Commission', 'Duration', 'Sales' we can observe that these features contain lot of outliers.
- From the hist plots, we can see that 'Commission', 'Duration', 'Sales' are right skewed by a heavy degree.

Treating anomalies for ‘Duration’:

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration
1508	25	IZI	Airlines	No	6.3	Online	1

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration
2845	40	C2B	Airlines	No	0.00	Online	4580

	count	mean	std	min	25
Age	2861.0	38.204124	10.678106	8.0	31
Commision	2861.0	15.080996	25.826834	0.0	0
Duration	2861.0	70.533030	134.234863	0.0	11

Fig. 2.1.6: Data description after anomaly treatment

Correlation Heatmap:

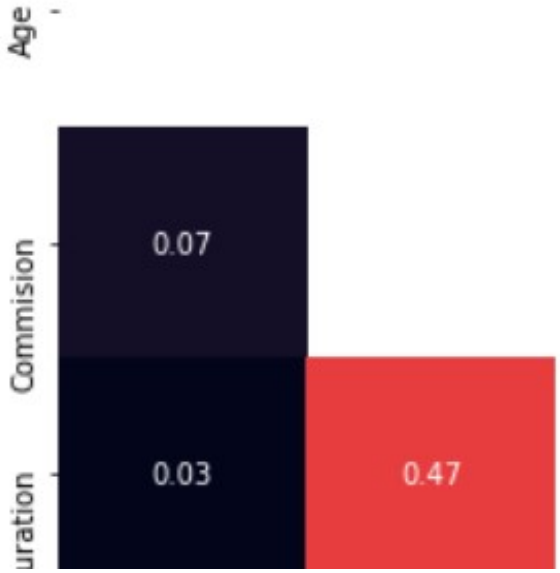


Fig. 2.1.7: Correlation Heatmap

Inference:

- From the correlation heat map we see that the continuous variable have very low correlation to each other.
- From the pairplot, we see that the only features which are highly correlated are Commission and Sales since bigger sales reward bigger commissions.

Scatter Plot for Bi-variate analysis:

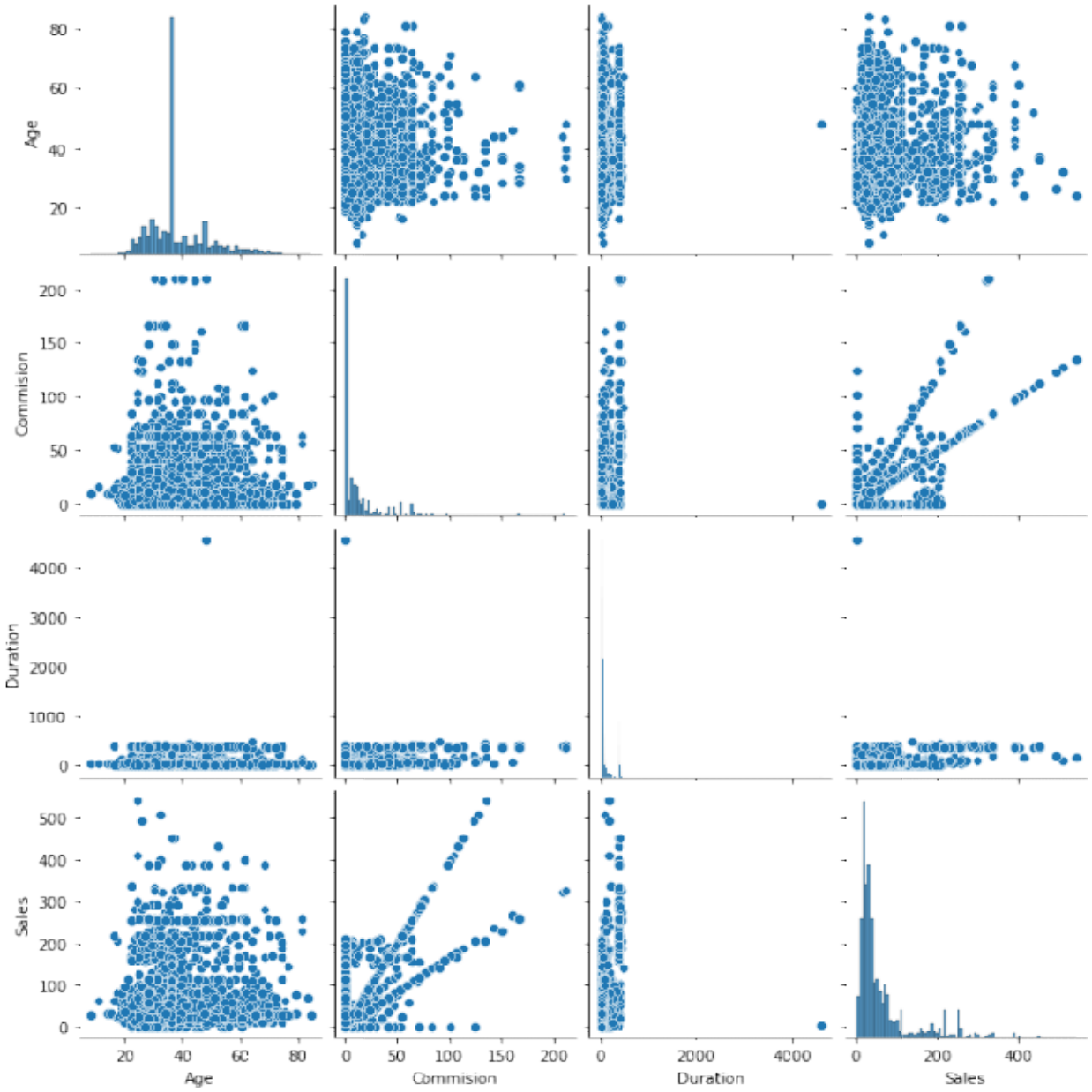


Fig. 2.1.8: Pairplot

Claim Rate:

```
0    1947
1     914
```

Fig. 2.1.9: Claim Rate of Insurance

From the above observation, we can see that the claim rate = 31.94%

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

```
<class 'pandas.core.frame.Dat
RangeIndex: 2861 entries, 0 t
Data columns (total 10 column
 #   Column         Non-Null C
---  ------         ----------
 0   Age            2861 non-n
 1   Agency_Code    2861 non-n
 2   Type           2861 non-n
 3   Claimed        2861 non-n
 4   Commision      2861 non-n
 5   Channel        2861 non-n
 6   Duration       2861 non-n
```

Fig. 2.2.1: Categorical values into Codes

We have chosen the test data size to be **30 percent** of the entire data set

	Age	Agency_Code	Type	Commision	Channel	Duration	Sa
630	31	1	1	0.00	0	402.0	
218	68	2	1	0.00	1	60.0	
1624	42	0	0	21.00	1	11.0	
932	44	1	1	23.76	1	51.0	
783	50	1	1	35.64	1	111.0	
...	...	...	...	...	...	...	
2812	19	1	1	10.50	0	32.0	
2107	45	2	1	0.00	1	37.0	

Fig. 2.2.2: Test data - 859 records

	Age	Agency_Code	Type	Commision	Channel	Duration	Sa
1255	69	0	0	6.00	1	7.0	1
2229	36	2	1	0.00	1	29.0	3
877	60	1	1	41.58	1	8.0	6
206	36	0	0	9.75	1	70.0	3
2207	36	2	1	0.00	1	39.0	5
...	...	...	...	...	...	...	
2763	27	2	1	0.00	1	19.0	2
905	36	2	1	0.00	1	30.0	4

Fig. 2.2.3: Train data - 2002 records

Decision Tree Model:

Using the grid search method to find the optimum value of the classifier attributes;

```
GridSearchCV(cv=3, estimator=DecisionTreeClass
             param_grid={'max_depth': [5, 6, 7
                         'min samples leaf': [
```

Fig. 2.2.4: GridSearch Values for parameters of DT

```
{'max_depth': 6, 'min_samples_leaf': 15, 'mi
```

Fig. 2.2.5: Best parameters for DT

- max_depth was chosen from a range of 5 to 10. The range was given since the original decision tree had depth of 14. So the range was given as 50-75% of the original depth.
- min_samples_leaf was given as 10 to 30 since if the values is too low then the tree is extremely outgrown and inefficient.
- min_samples_split was chosen in the above range since it covered about 1-2% of the total train data set.

Features Importance for DecisionTreeClassifier:

```
Age            0.
Agency_Code    0.
Type           0.
Commision      0.
Channel        0.
Duration       0.
```

Fig. 2.2.6: Features Importance for DT

Inference:

- We observe that 'Agency_Code' and 'Sales' have the highest importances whereas 'Channel' does not influence the prediction in any way.
- If we remove ‘Channel’ column, it will not affect the prediction in any way for the decision tree

Random Forest Model:

Using the grid search method to find the optimum value of the classifier attributes;

```
GridSearchCV(cv=3, estimator=RandomForestClassifier(
             param_grid={'max_depth': [6, 7, 8], 'ma
                         'min_samples_leaf': [15, 20
```

Fig. 2.2.7: GridSearch Values for parameters of RF

```
{'max_depth': 6,
 'max_features':
 'min_samples_lea
 'min_samples_spl
```

Fig. 2.2.8: Best parameters for RF

- ❖ max_depth was chosen from a range of 5 to 10. The range was given since the original decision tree had depth of 14. So the range was given as 50-75% of the original depth.
- ❖ max_features was chosen as 5,6,7 since data needs to be boot strapped for the model. Too high value would increase in Out-of-bag-error.

- ❖ min_samples_leaf was given as 10 to 30 since if the values is too low then the tree is extremely outgrown and inefficient.
- ❖ min_samples_split was chosen in the above range since it covered about 1-2% of the total train data set.
- ❖ n_estimators should be ideally in the multiples of 100s.

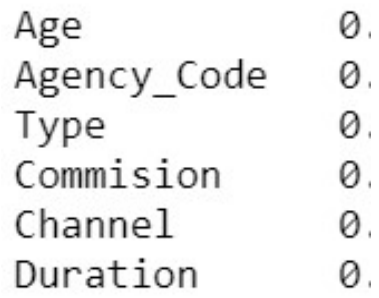


Fig. 2.2.9: Features Importance for RF

Inference:

- We observe that 'Agency_Code', 'Sales', 'Product Name' have the highest importances whereas 'Channel' has negligible influence on the prediction.

ANN Model:

Using the grid search method to find the optimum value of the classifier attributes;

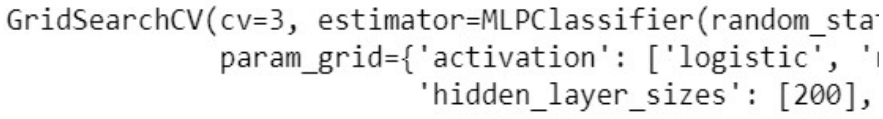


Fig. 2.2.10: GridSearch Values for parameters of ANN

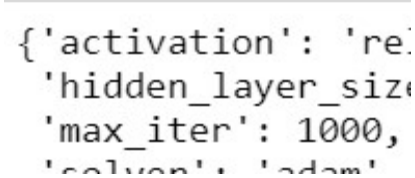


Fig. 2.2.11: Best parameters of ANN

- activation was chosen from as ‘logistic’ and ‘relu’ since these two functions are the most popular and accurate.
- Hidden_layer_sizes was chosen as 100,200,500 but later on finalized at 200 to reduce the complexity of the grid_search. Higher values of this attribute provide better performance.
- max_iter was chosen in multiples of 1000 since the higher the iteration, the higher is the model accuracy. Too high value of iterations will make the model run for a longer time.
- Solver of sgd and adam are the most popular and efficient functions.
- tol sets the tolerance for the model. A moderate value of 0.1 and 0.01 was chosen to optimize the accuracy and speed of the model run time.

2.3 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.

Performance Metrics for;

1. Decision Tree Model:

Accuracy:

```
The accuracy score for Decision Tree model for t
The accuracy score for Decision Tree model for
```

Confusion Matrix:

```
The Confusion matrix for the Decision Tree model

array([[1208,  151],
The Confusion matrix for the Decision Tree mode

array([[516,  72],
```

Classification Report:

Train data:

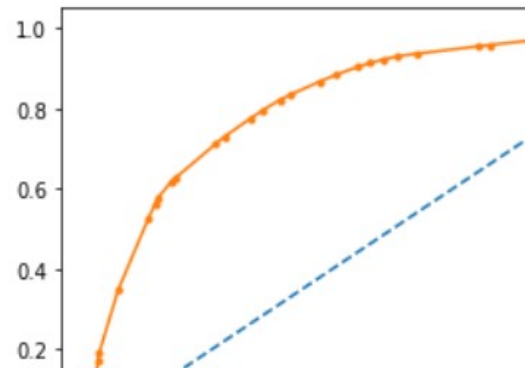
	precision	recall	f1
0	0.82	0.89	
1	0.71	0.58	
accuracy			

Test data:

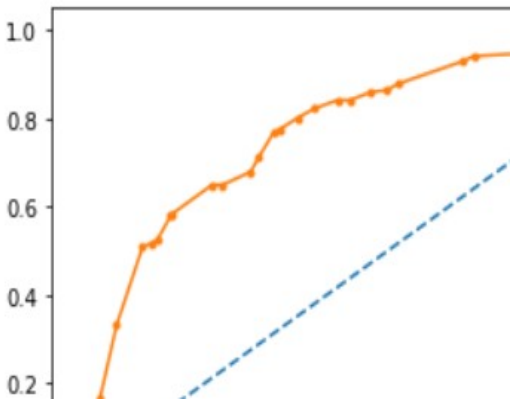
	precision	recall	f
0	0.80	0.88	
1	0.66	0.52	
accuracy			

ROC Curve and AUC score:

Train data:



Test data:



Inference:

- We see that the precision, recall and f1_score for train data and test data is nearly similar, the test data being few decimals lower. We have tried the training for 3 iterations.
- Also, the measures are slightly poorer for Claimed insurances because the claimed rate is only 31%. With further data, we might be able to get a better performance.
- The area under the train roc curve is neither too steep not too flat. The area under the test roc curve is a much flatter compared to the train curve. The model excels for the train data but does not perform that well for the test data. The model can be stated as slightly overfitted.

2. Random Forest Model:

Accuracy:

```
The accuracy score for Random Forest model for t
The accuracy score for Random Forest model for t
```

Confusion Matrix:

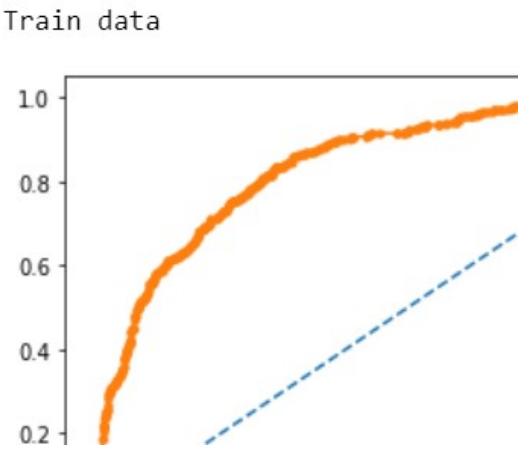
```
The Confusion matrix for the Random Forest mode

array([[1206,  153],
```

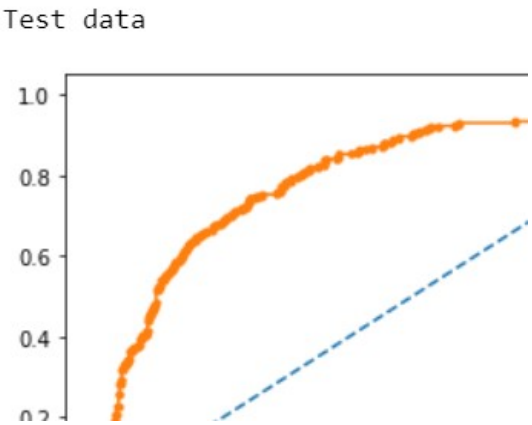
```
The Confusion matrix for the Random Forest mode

array([[514,  74]
```

Classification Report:

```
Train data
              precision    recall  f

           0       0.82      0.89
           1       0.72      0.60

    accuracy

Test data
              precision    recall  f

           0       0.82      0.87
           1       0.68      0.59

    accuracy
```

ROC Curve and AUC score:





Inference:

- We see that the precision,recall and f1_score for train data and test data is almost similar, the test data being few decimals lower. Hence we can say that the model is not overfitted or underfitted.
- Also, the measures are slightly poorer for Claimed insurances because the claimed rate is only 31%. With more data we might be able to get a better performance.
- The area under the train roc curve is neither too steep not too flat. The area under the test roc curve is slightly flatter compared to the train curve. The model performs better for train data than the test data. However the auc score is good for both the models.

3. ANN Model:

Accuracy:

```
The accuracy score for ANN model for trai
The accuracy score for ANN model for test
```

Confusion Matrix:

```
The Confusion matrix for the ANN model fc

array([[1283    76]
The Confusion matrix for the ANN model f

array([[557,  31],
```

Classification Report:

Train Data

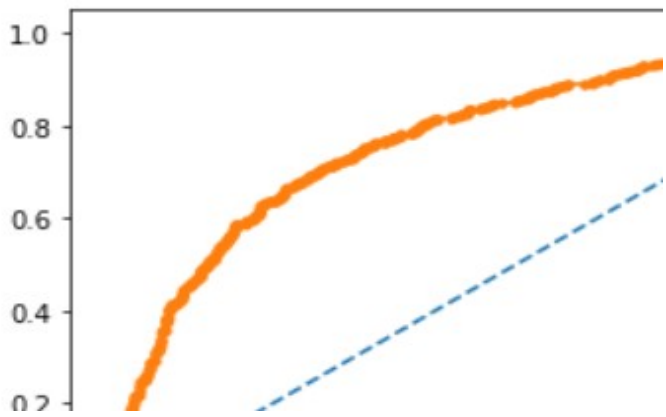
	precision	recall	f1
0	0.74	0.94	
1	0.73	0.31	
accuracy			

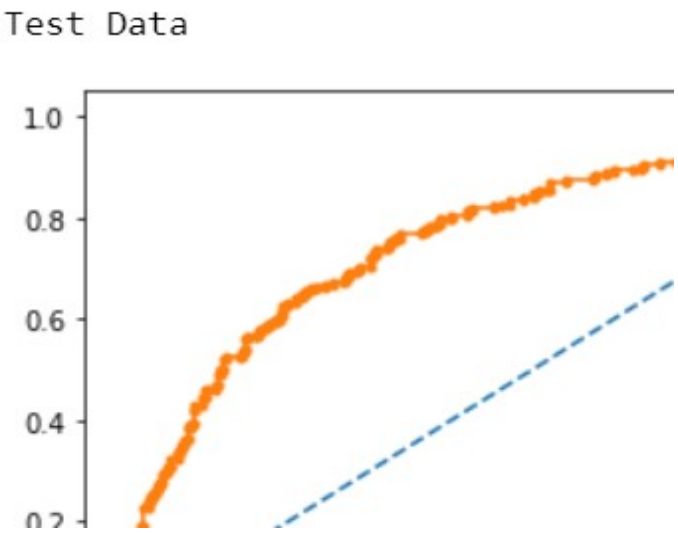
Test Data

	precision	recall	f1
0	0.75	0.95	
1	0.73	0.31	
accuracy			

ROC Curve and AUC score:

Train Data





Inference:

- We see that if our model's recall increases then it decreases the precision of the model and vice versa. Hence we choose these parameters which give an optimal values of recall and precision.
- Also, the measures are slightly poorer for Claimed insurances because the claimed rate is only 31%. With more data we might be able to get a better performance.
- The area under the train roc curve is neither too steep not too flat. The area under the test roc curve similar to the train curve. This means that the model is going to perform similarly for the train as well as the test data. AUC score being less than 0.8 which can be increased with more amounts of data.

In the above metrics we see different metrics providing us the respective performance of the models.

In our case, the metric which is the most important is the 'precision'. Our main objective is to predict which customer is going to claim for insurance thus saving the company from the extra insurance cost. So precision can help us to predict more accurately as to which customers might initiate claims for their travel purposes.

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

Model	Accuracy		Precision		Recall		f1-score		AUC_Score	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Decision Tree	0.79	0.77	0.71	0.66	0.58	0.52	0.64	0.59	0.83	0.79
Random Forest	0.8	0.78	0.72	0.68	0.6	0.59	0.65	0.63	0.85	0.81
ANN	0.74	0.75	0.73	0.73	0.31	0.31	0.44	0.43	0.79	0.78

Table. 2.4.1: Final Model Metrics

Inference:

- On the basis of the metrics summarized in the above table, we can see that the model which is best optimized is the Random Forest model.
- We can observe that the Random Forest model has the highest Accuracy, AUC Score and f-1 score among all the other models.
- When it comes to the precision, the random forest model optimizes its high accuracy and recall much better than the other 2 models, while still providing a high competent precision compared to the other 2 models.
- In order to increase the metrics of Decision Tree and ANN, it will require us for a larger data set of Claimed cases. To increase the recall for ANN, it will require further optimization of the classifier attributes but at the cost of precision.

For the problem at hand, the random forest performs the best among the three because even though we require a higher precision which is slightly higher in ANN model, the other metrics such as accuracy and recall are also important where ANN does not perform well enough. Random Forest balances all the metrics and performs in the best manner.

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

Key Insights:

- From the analysis of the data given, we could understand that a very important metrics which is 'precision' was not extremely high for any of the used models.

- The major reason for this could be that the data which was provided by the business for the customers who Claim the insurance was very limited.

- Whereas the metrics are extremely high for all the models for the data which did not Claim any insurance for their tours. Hence, the model performance is directly proportional to the relevant data provided.

Recommendations to Business to solve the objective:

- The business is using data like different channels(offline/online), which had the least Feature_Importance value in both the Decision Tree(0) as well as the Random Forest model. This suggests that such data is not amounting to anything or influencing the predictive modeling in anyway. So such columns can be discarded for further analysis.

- In order to predict the Claims more accurately and with higher precision, business needs to gather larger data(records) than provided.

- While gathering data, the columns which can heavily influence the predictions should be carefully designed and added to the data set. For eg. Mode of transport, type of employment, financial status, insurance policy terms and conditions, etc.

- More number of features which are relevant to the problem can be added to the data set.