DATA MINING PROJECT REPORT

KIRAN.N

GREAT LEARNING

Table of Contents

List Of Figures

List of Tables

Problem 1

Executive Summary

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

Data Dictionary

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)

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

Sample of the 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 : Sample DataSet

Dataset has 7 columns with 210 rows. Each row in the dataset corresponds to individual user's financial activities during past few months.

Exploratory Data Analysis

Let us check the types of variables in the data frame.

```
spending                      float64
advance_payments              float64
probability_of_full_payment   float64
current_balance               float64
credit_limit                  float64
min_payment_amt               float64
max_spent_in_single_shopping  float64
```

All the seven columns in the dataset are of float data type.

Check for missing values in the dataset

```
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
```

From the above results we can see that there is no missing value present in the dataset.

Check for Outliers in the dataset

Let us plot the boxplot for all the numeric columns of the dataset.

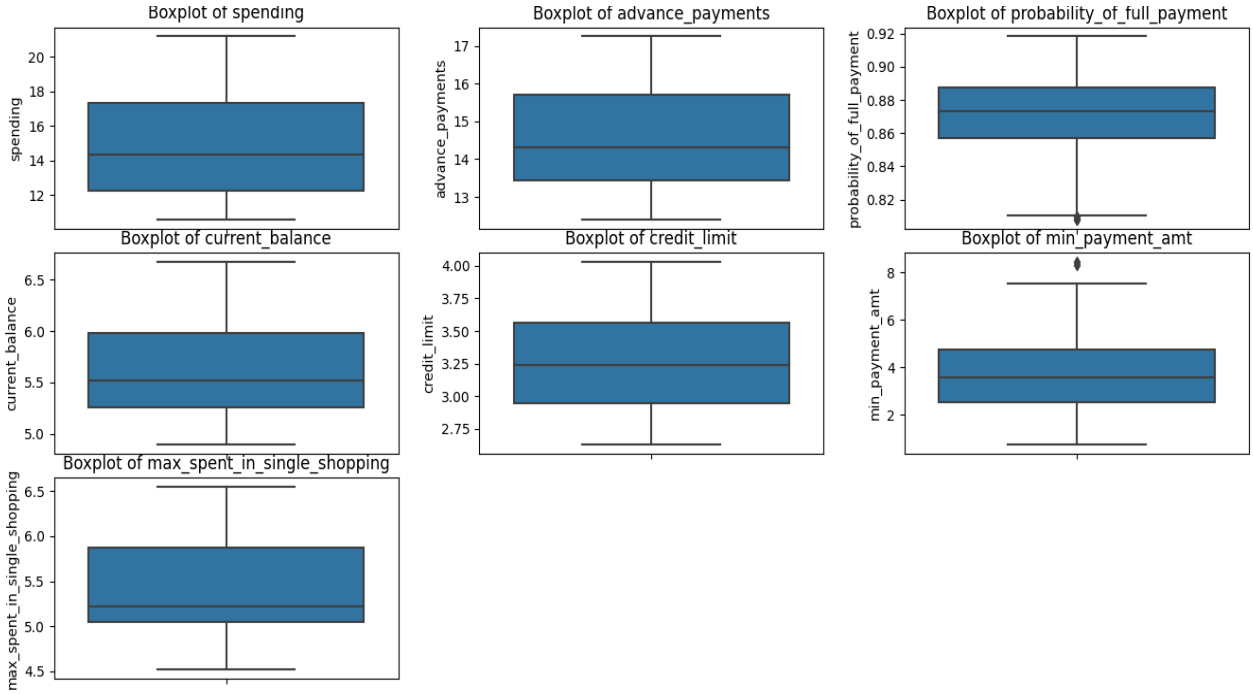

Figure 1 : BoxPlot of All Numeric Columns

From the above figure we observe outliers are present in probability_of_full_payment and min_payment_amt. Since we have very few outliers we proceed without treating them.

Distribution Analysis

Let us plot the histogram for all the numeric columns of the dataset.

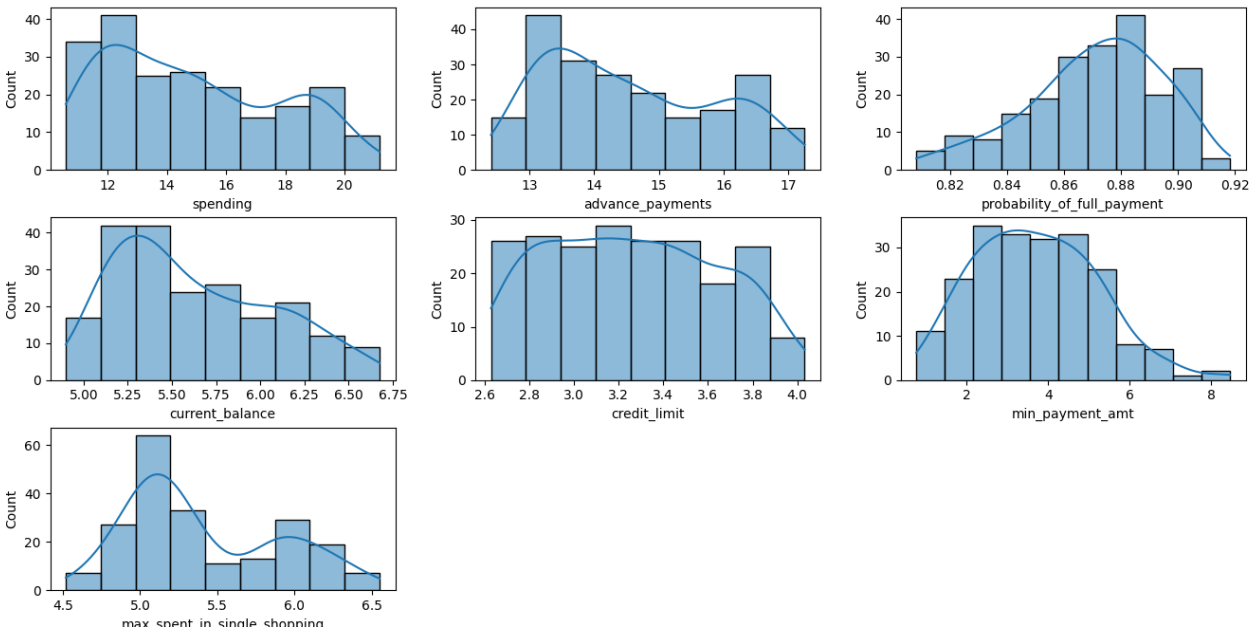

Figure 2 : Histogram of All Numeric Columns

From the above figure we infer following information:

- Data in 'spending' column is right skewed.
- Data in 'advance_payments column is right skewed.
- Data in 'probability_of_full_payment' column is left skewed.
- Data in 'current_balance' column is rightly skewed.
- Data in 'credit_limit' column is almost normally distributed.
- Data in 'min_payment_amt' column is rightly skewed.
- Data in 'max_spent_in_single_shopping' column is rightly skewed.

Descriptive Statistics

Descriptive statistics include those that summarize the central tendency, dispersion and shape of a dataset's distribution,

	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 : Descriptive Statistics of Data

Pair Plot

Pairplot shows the relationship between the variables in the form of scatterplot and the distribution of the variable in the form of histogram.

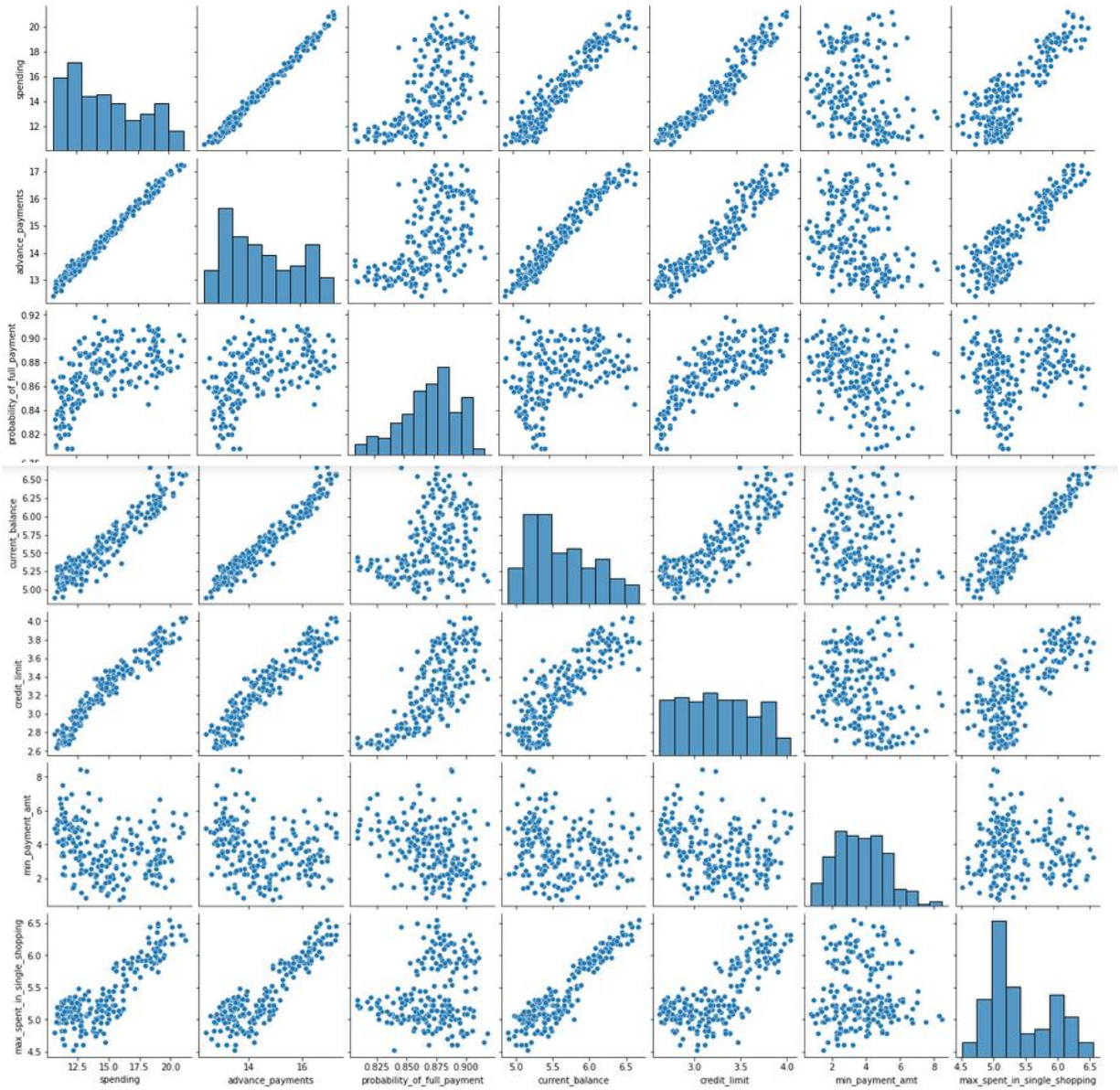

Figure 3 : PairPlot

Correlation Plot

From the correlation plot, we can see the correlation among different variables. Correlation values near to 1 or -1 are highly positively correlated and highly negatively correlated respectively. Correlation values near to 0 are not correlated to each other.

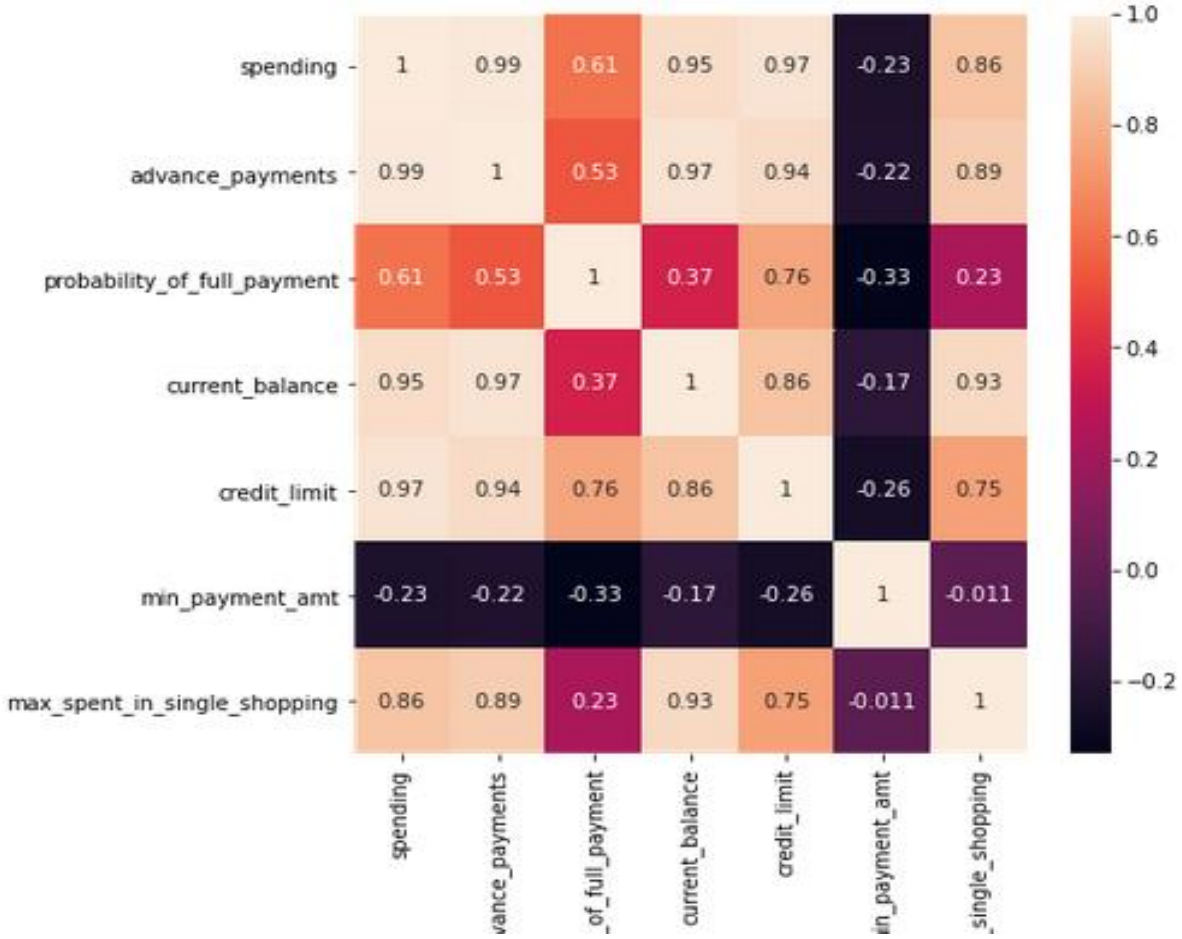

Figure 4 : Correlation Plot

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

Let us compute the standard deviation and variance for all numeric columns.

	columns	Standard Deviation
0	spending	2.909699
1	advance_payments	1.305959
2	probability_of_full_payment	0.023629
3	current_balance	0.443063
4	credit_limit	0.377714
5	min_payment_amt	1.503557
6	max_spent_in_single_shopping	0.491480

Table 3 : Standard Deviation of Numeric Columns

	columns	Variance
0	spending	8.466351
1	advance_payments	1.705528
2	probability_of_full_payment	0.000558
3	current_balance	0.196305
4	credit_limit	0.142668
5	min_payment_amt	2.260684
6	max_spent_in_single_shopping	0.241553

Table 4 : Variance of Numeric Columns

The data given in the dataset are in different scales. The probability value varies from o to 1 and other columns vary in a slightly higher range. From the above two tables we observe there is drastic variation in the values of standard deviation and variance of different numeric columns. Data being fed to the models should be in standard form else the different scales in different columns would hamper the model performance.

So, to have the data in normalized or standardised form scaling is necessary.

The different scaling techniques available are:

1. **Z-score Scaling:** Here each value is computed using following formula.

   $Z = (X - \mu) / \sigma$

   where X -> current value from dataset
   μ -> mean value of the column.
   σ -> Standard Deviation of the column.

   - Scaled data will have mean tending to 0 and standard deviation tending to 1.
   - This scaling techniques works well with Outliers also.
   - In python we have StandardScaler under sklearn.preprocessing to do this task.

2. **Min-Max scaling:** Here each value is computed using following formula.

   Val = (X-Xmin)/(Xmax-Xmin)

   where X -> current value from dataset
   Xmin -> min value of the column
   Xmax -> max value of the column

   - Scaled data will range between 0 and 1
   - In python we have MinMaxScaler under sklearn.preprocessing to do this task.

Here we will be using Z-score scaling and scale our data.

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754355	1.811968	0.178230	2.367533	1.338579	-0.298806	2.328998
1	0.393582	0.253840	1.501773	-0.600744	0.858236	-0.242805	-0.538582
2	1.413300	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	1.196340	0.591544	1.155464	-1.088154	0.874813

Table 5 : Sample Scaled Dataset

	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	1.243978e-15	-1.089076e-16	-2.994298e-16	5.302637e-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 6 : Descriptive Statistics of Scaled Data

After scaling the data, we observe mean value of all columns tends to '0' and standard deviation of all columns tends to '1'.

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

Hierarchical clustering method is based on hierarchy representation of clusters where parent cluster node is connected to further to child cluster node. A node represents collection of data points to one cluster.

The agglomerative clustering is the most popular and common hierarchical clustering. Records are sequentially grouped to create clusters, based on distance between records and distance between clusters. Also produces a useful graphical display of the clustering process and results, called a dendogram.

Hierarchical Clustering Steps:

- Star with 'n' clusters where each record will be a cluster.
- Two closest records are merged into a cluster.
- At every step, 2 closest clusters with smallest distance are merged.
    - Either single records are added to existing clusters or
    - Two existing clusters are merged.
- Repeat till there is a single cluster that includes all the records.

To calculate the distance between the clusters we have various Linkage types available like Single Linkage, Complete Linkage, Average Linkage, Centroid Linkage and Ward Linkage.

In python we have linkage and dendogram functions available in “scipy.cluster.hierarchy” . Using the ward method for distance calculations we have computed the following dendogram.

A dendogram is a treelike diagram that summarises the process of clustering. On the x-axis we have records. Similar records are joined by lines whose vertical length reflects the dis b/w the records. The greater the diff in height, the more dissimilarity. By choosing a cut-off distance on the y-axis a set of clusters is created. If the difference in the height of the vertical lines of the dendrogram is small, then the clusters that are formed will be similar.

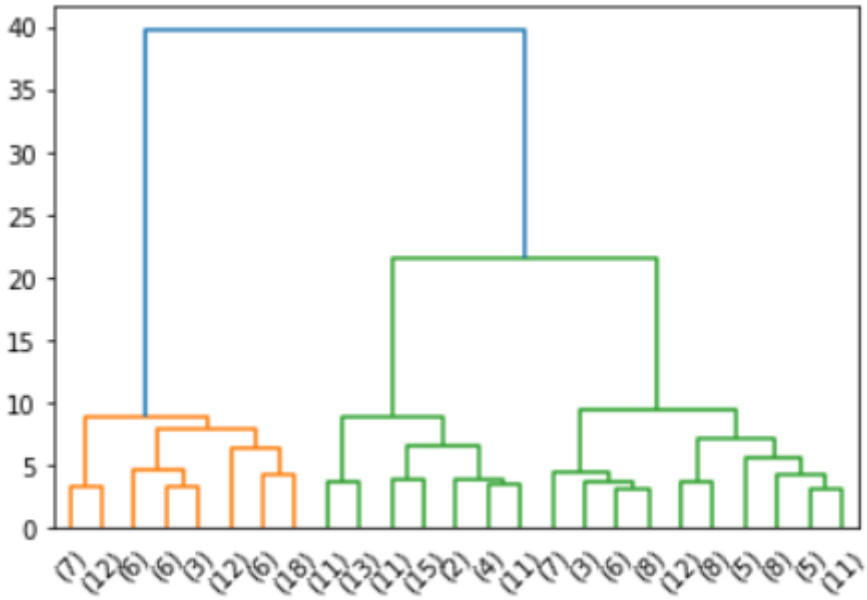

Figure 5 : Dendogram

Since we have large number of records in the dataset, the dendogram will be large enough, so we have considered to show last 25 merges in the cluster. The clusters which are similar are highlighted in same colour in the above dendogram. From the above figure we conclude the given dataset can be divided into 2 clusters optimally.

Using the Fcluster method available in scipy.cluster.hierarchy we have created the clusters using ‘maxclust’ criterion by passing 2 as maxclust value. Also, clusters can be created based on distance criteria.

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

Table 7 : Sample Dataset with Cluster Information

The H_clusters column in the above table will have the information about the cluster to which the current record belongs.

After dividing the given data into 2 clusters based on maxclust criteria and ward linkage type we have 70 records in cluster-1 and 140 records in cluster-2.

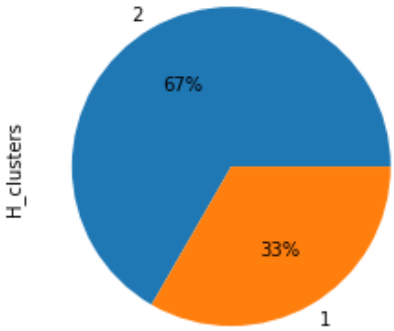

Figure 6 : Distribution of Customers in H-Clusters

Let us aggregate mean values of all numeric columns cluster-wise.

H_clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
1	18.371429	16.145429	0.884400	6.158171	3.684629	3.639157	6.017371	70
2	13.085571	13.766214	0.864298	5.363714	3.045593	3.730723	5.103421	140

Table 8 : Cluster-wise Aggregated Mean Values

Inferences from above table:

- Cluster-1 represents the customers who spend more with the average spends of 18.37 thousand, and Cluster-2 represents the customers who spend less with the average spends of 13.08 thousand.
- Customers from cluster-1 spend max of 6.01 thousand on a single payment while customers from cluster-2 spend max of 5.1 thousand on a single payment.
- Cluster-1 customers make more advanced payments compared to cluster-2 customers.
- Cluster-1 customers have a credit limit of 36.8 thousand which is more compared to the Cluster-2 customers credit limit of 30.4 thousands.
- Customers from cluster-1 make full payments with the probability of 88.4% while customers from cluster-2 make full payments with the probability of 86.4%
- Cluster-1 customers have higher current balance to make purchase.
- Cluster-2 customers pay higher minimum payment amount compared to Cluster-1 customers.

Q 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.

K-means clustering is an unsupervised learning algorithm whose goal is to find groups or assign the data points to clusters on the basis of their similarity. Which means the points in same cluster are similar to each other and in different clusters are dissimilar with each other. Here K means the number of clusters. K-means clustering is very useful and beneficial when most of the data is in unorganized manner.

The working of K-means clustering is depends on the distance metrics, which are used to find the similarity within data points. The popular distance metric are: Euclidean Distance, Manhattan Distance, Chebyshev Distance, Minkowski Distance, Mahalanobis Distance.

K-means Clustering Steps:

- Specify value of 'k'
- Partition dataset to k initial clusters with random centroid to each cluster.
- Assign each record to cluster with nearest centroid.
- Recalculate centroid for losing and receiving clusters.
- Do reassignments occur ? If YES repeat step-3
- Else Finalize clusters.

In python to perform K-means clustering, we have KMeans function in sklearn.cluster module. Just by passing the 'k' value and fitting the scaled data in it we can create clusters.

Elbow method is most popular and well-known method to find the optimal no. of clusters or the value of k in the process of clustering. This method is based of plotting the value of cost function against different values of k.

	k_values	wss
0	1	1470.000000
1	2	659.171754
2	3	430.658973
3	4	371.385091
4	5	327.212782
5	6	289.315995
6	7	262.981866
7	8	241.818947
8	9	223.912542
9	10	206.396122

Table 9 : K Value verses Within Sum of Squares

Using the above table, we will plot Elbow method.

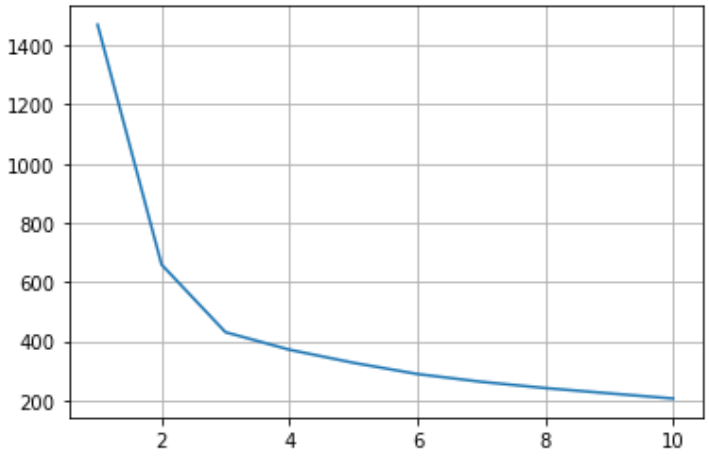

Figure 7 : Elbow Method Plot

As the K-value increases the WSS/Inertia value keeps dropping. From above figure, the distortion declines most at 3. Hence the optimal value of k will be 3 for performing the clustering. In other words, the plot looks as an arm with an elbow at k = 3.

Let us proceed by creation K-means cluster with k=3.

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

Table 10 : Sample Dataset with K-means Cluster Information

K_clusters column in the above table will have the information about the cluster to which the current record belongs.

After dividing the given data into 3 clusters based on Eucledian distance we have 71 records in cluster-0, 72 records in cluster-1 and 67 records in cluster-2.

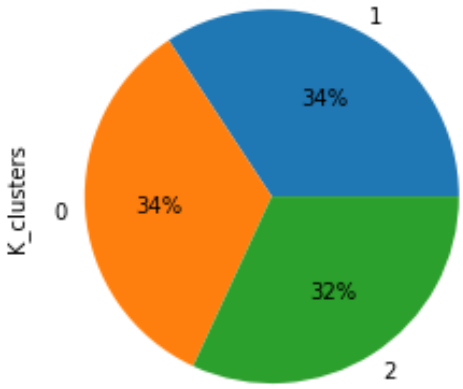

Figure 8 : Distribution of Customers in K-Clusters

K_clusters	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	Freq
0	14.437887	14.337746	0.881597	5.514577	3.259225	2.707341	5.120803	71
1	11.856944	13.247778	0.848253	5.231750	2.849542	4.742389	5.101722	72
2	18.495373	16.203433	0.884210	6.175687	3.697537	3.632373	6.041701	67

Table 11 : K-Means Cluster-wise Aggregated Mean Values

Inferences from above table:

- Cluster-1 represents the customers who average spends are 11.8 thousand, Cluster-2 represents the customers who average spends are 18.5 thousand, Cluster-0 represents the customers who average spends are 14.4 thousand.
- We can summarise Customers in Cluster-0 spend moderately, Customers in Cluster-2 spend more and Customers in Cluster-1 spend less compared to other two clusters.

Silhouette Method

Silhouette is a different method to determine optimal number of clusters for given dataset. It defines as a coefficient of measure of how similar an observation to its own cluster compared to that of other clusters. The range of silhouette coefficient varies between -1 to 1. 1 value indicate that an observation is far from its neighbouring cluster and close to its own whereas -1 denotes that an observation is close to neighbouring cluster than its own cluster. The 0 value indicate the presence of observation on boundary of two clusters.

The silhouette_score for k = 2 is 0.466, for k = 3 is 0.4 and k = 4 is 0.327. The count of silhouette_samples less than 0 for k = 2 is 1, for k = 3 is 0 and for k = 4 is 3

For k = 2 we have max silhouette_score but we have one silhouette_sample which is negative. For k=3 there is no negative silhouette_sample so k = 3 is the optimal value for number of clusters.

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

By applying Hierarchical clustering on given dataset we have observed dividing the dataset into 2 clusters is optimal based on similarity in the data.

By applying K-means clustering on given dataset we have observed dividing the dataset into 3 clusters is optimal based on distance between records

Table 8 and Table 11 gives the cluster wise means of all the columns present in the dataset w.r.t Hierarchical Clustering & K-means clustering respectively.

Considering Hierarchical Clustering Results:

By looking at the mean values in the table we can infer that the clustering has been made based on their financial activities like spending, ways of repayment, usage of credit limit and so on.

One group of customers make higher spends, higher advance payments, higher probability of making full payments, higher current balance, higher credit limit, higher maximum spends on single shopping and lower minimum payment by the customer while making payments for purchases made monthly compared to another group. So we can summarise it as two groups with one group with higher value financial activities and the other group with slightly lower value financial activities.

Considering K-Means Clustering Results:

By looking at the mean values in the table we can infer that the clustering has been made based on their financial activities like spending, ways of repayment, usage of credit limit and so on.

Customers in Cluster-0 have carried out moderate valued financial activities, Customers in Cluster-2 have carried out higher valued financial activities, and Customers in Cluster-1 have carried out lower valued financial activities compared to other two clusters.

Cluster Group Profiles

Group 1 : High Spending
Group 3 : Medium Spending
Group 2 : Low Spending

Promotional strategies for each cluster

Group 1: High Spending Group

- Strategy of giving reward points on their spends could drive in more business.
- As the probability of their full payment is quite high and the current balance is high they can be provided offers related to high valued or branded products.
- Increase their credit limits and thereby allowing then to make more and more purchases.
- Based on their repayment record they could be provided personal loans.

Group 3: Medium Spending Group

- Giving more and more discount offers on regularly used/ basic need products and motivating them to spend more.
- As the probability of their full payment is quite high increase their credit limit and allowing them to use it to the full potential.
- These customers are more loyal and are quite in good number so promote premium cards/loyalty cars to increase transactions.
- Link them with the E-commerce promotional offers.

Group 2: Low Spending Group

- Since the probability of their full payment is quite low compared to other to groups they have to be given loans/ credit limits very cautiously.
- They could turn defaulters.
- They must be motivated to do full payments on time by providing rewards/ loyalty bonus.

Problem – 2

Executive Summary

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.

Data Dictionary

1. Claimed - Target: Claim Status
2. Agency_Code - Code of tour firm
3. Type - Type of tour insurance firms
4. Channel - Distribution channel of tour insurance agencies
5. Product - Name of the tour insurance products
6. uration in days - Duration of the tour
7. Destination - Destination of the tour
8. Sales - Amount worth of sales per customer in procuring tour insurance policies in rupees
9. Commision - The commission received for tour insurance firm in percentage of sales
10. Age - Age of insured

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

Sample of the 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 12 : Sample Dataset

Dataset has 3000 rows and 10 columns. Each row in the dataset represent tour insurance claim made to an Insurance firm.

Exploratory Data Analysis

Let us check the types of variables in the data frame.

```
Age               int64
Agency_Code      object
Type             object
Claimed          object
Commision       float64
Channel          object
Duration          int64
Sales           float64
Product Name     object
Destination      object
```

There are total 10 columns in the dataset out of which 6 are of object datatype, 2 are of integer type and 2 are of float type.

Check for missing values in the dataset

```
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
```

From the above results we can see that there is no missing value present in the dataset.

Univariate Analysis

Let us plot the histogram for all the numeric columns of the dataset.

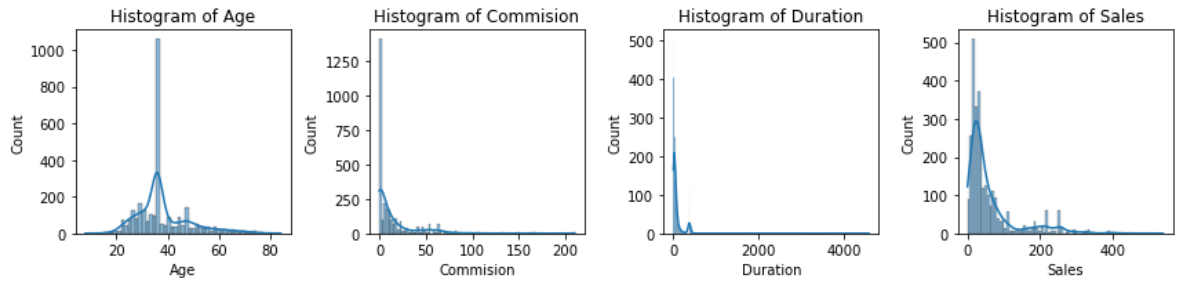

Figure 9 : Histogram of all Numeric Columns

From the above plots we infer data in Age column is almost normally distributed. Data in Commission, Duration and Sales column right skewed

Let plot the Count Plot for all object columns of the dataset.

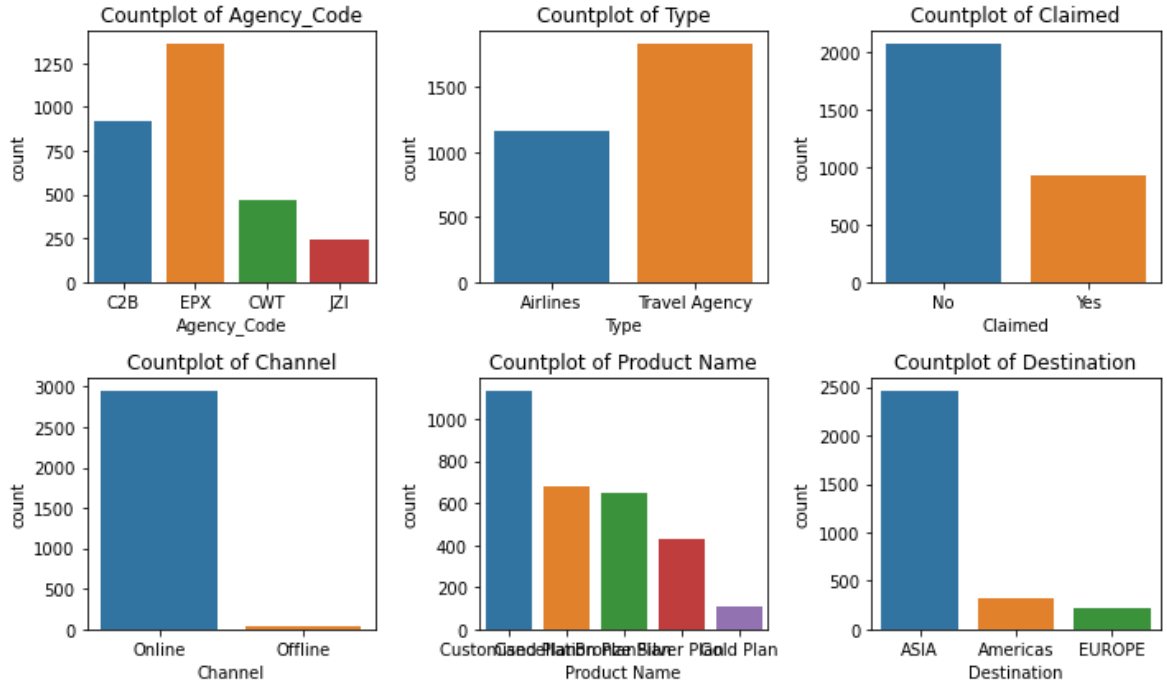

Figure 10: Countplot of all Object Columns

Multivariate Analysis: Pairplot

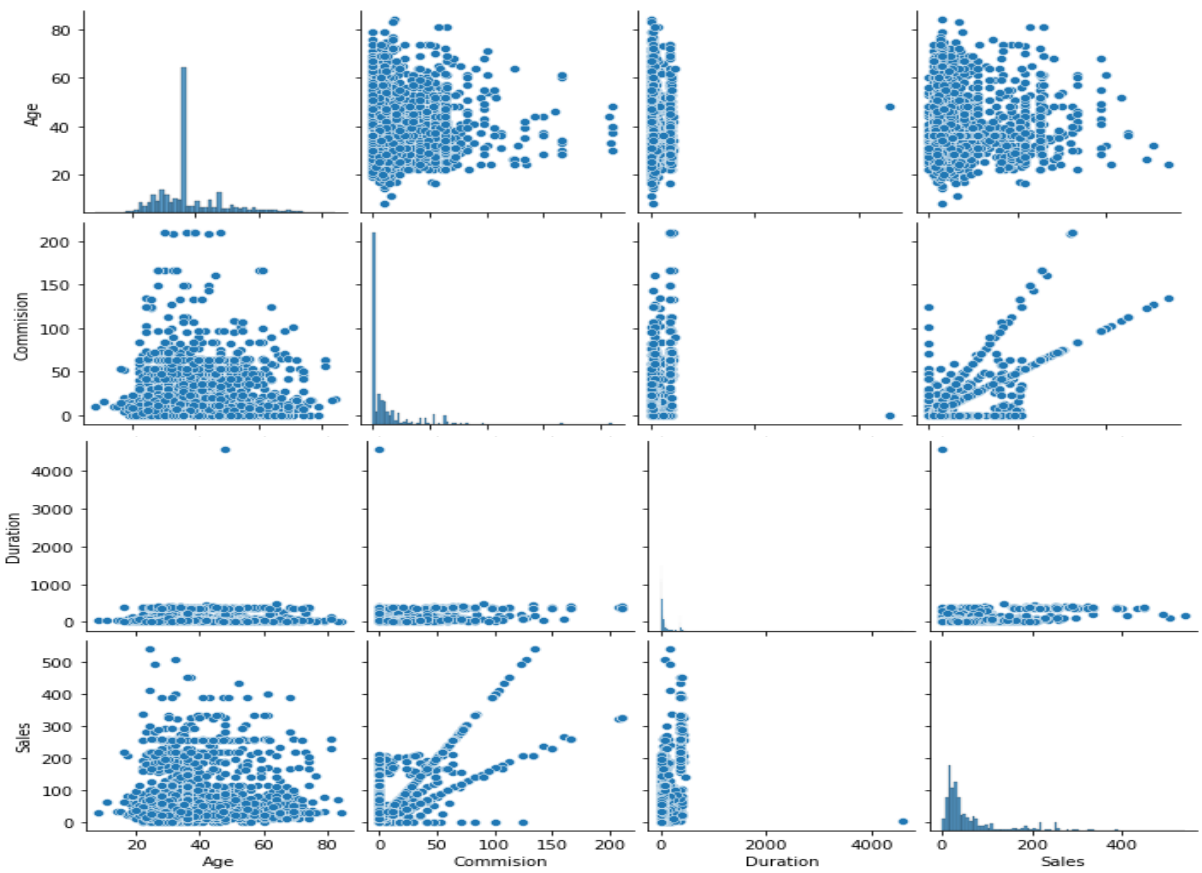

Figure 11 : Pairplot

Bivariate Analysis

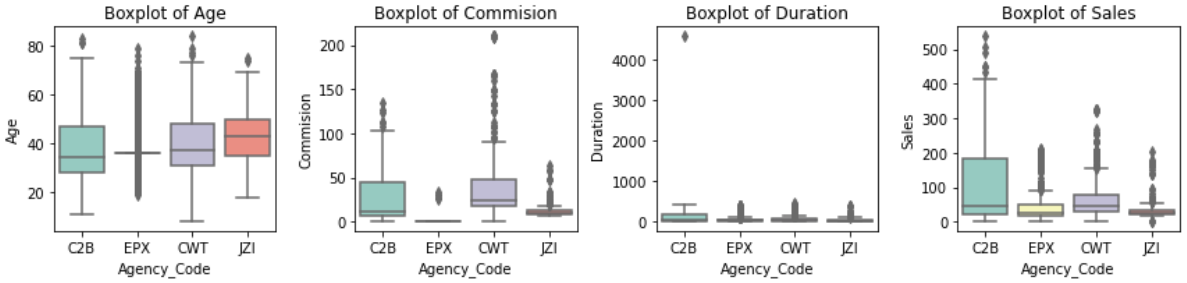

Figure 12 : Bivariate Analysis Agency Code Verses All Numeric Columns

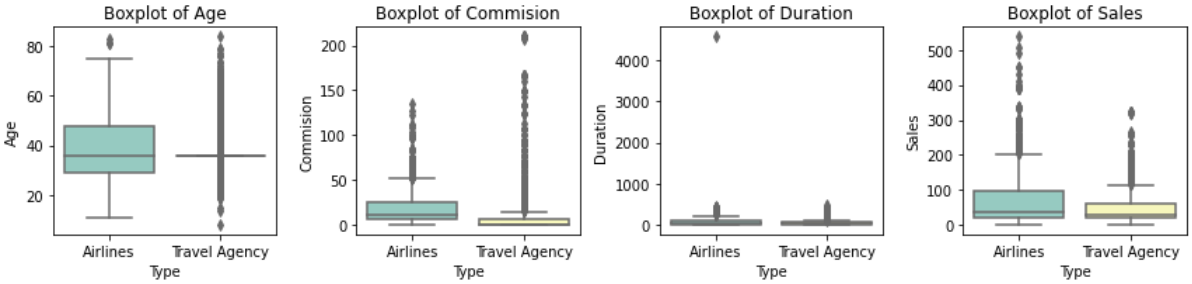

Figure 13 : Bivariate Analysis Type Verses All Numeric Columns

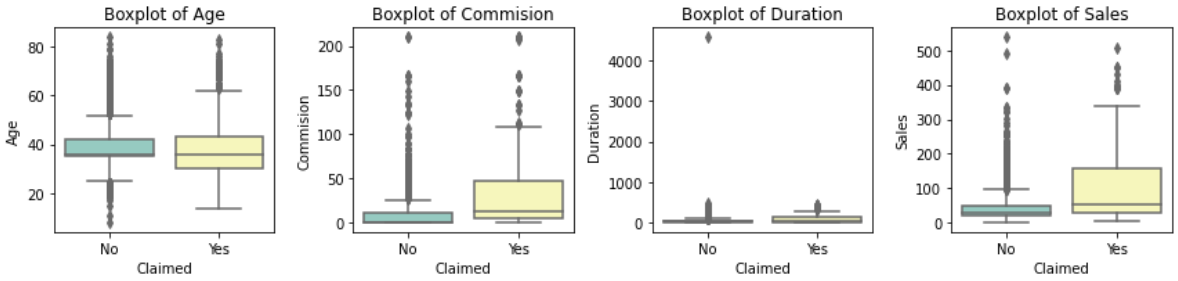

Figure 14 : Bivariate Analysis Claimed Verses All Numeric Columns

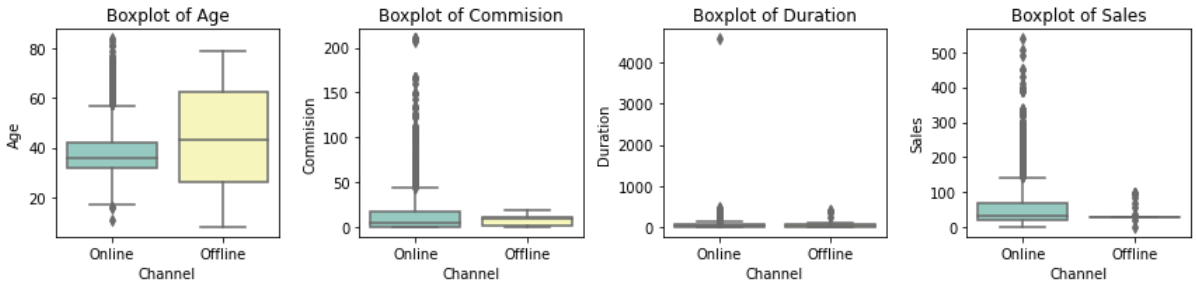

Figure 15 : Bivariate Analysis Channel Verses All Numeric Columns

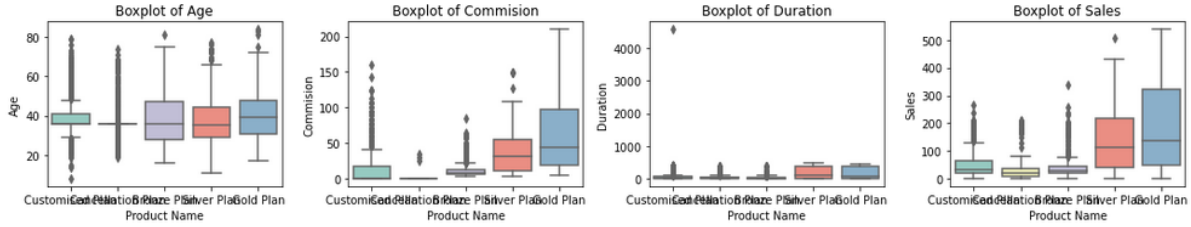

Figure 16 : Bivariate Analysis Product Name Verses All Numeric Columns

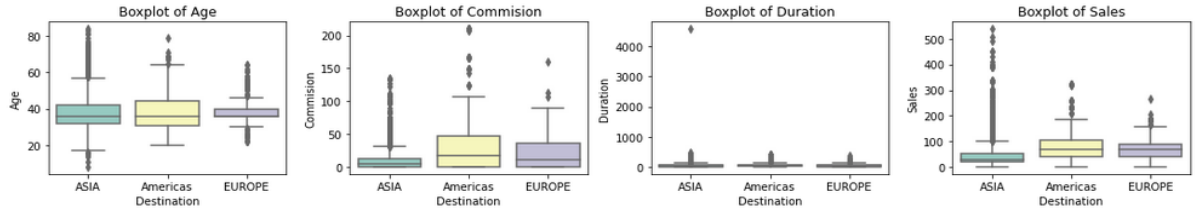

Figure 17 : Bivariate Analysis Destination Verses All Numeric Columns

Correlation Plot

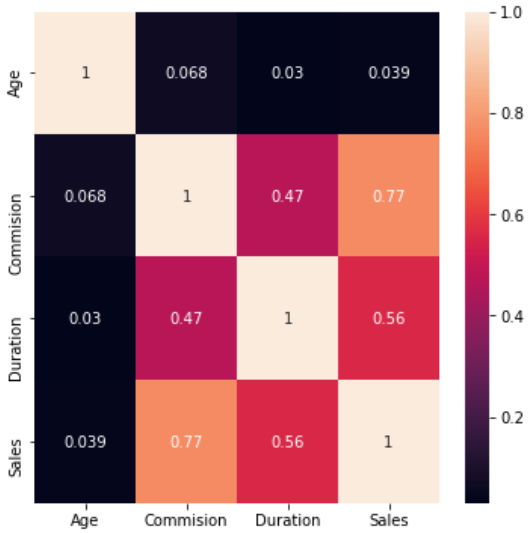

Figure 18 : Correlation Plot

Check for Outliers in the dataset

Let us plot the boxplot for all the numeric columns of the dataset.

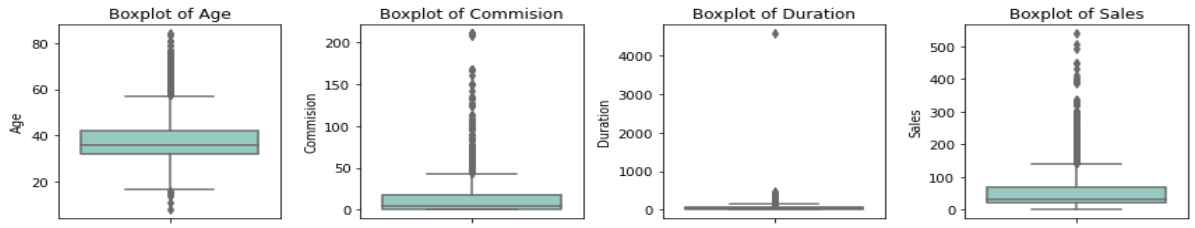

Figure 19 : Boxplot of Numeric Columns

From the above plot we observe outliers are present in all numeric columns. We will treat all the outliers.

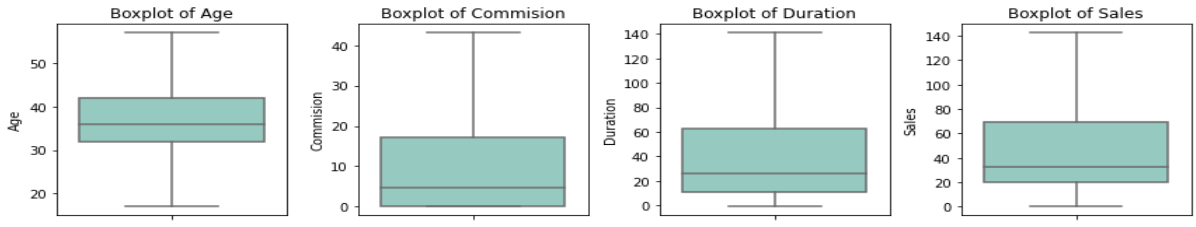

Figure 20 : Boxplot of Numeric Columns After Outlier Treatment

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

Before we proceed with model building activity we need to check if duplicates are present in the given dataset and remove them as they will hamper the model performance. In our dataset we have found 286 duplicate rows and we have removed them.

Before feeding the data into model all the object datatype columns should be converted into integer type. Here we had 6 object columns in given dataset we have converted them into integer type using pd.categorical().codes() .

Data Split

Before splitting the data into train and test data we segregate Dependent and Independent columns in the dataset. In the current dataset Claimed is the independent column and all other columns are independent columns.

We split the data into test and train samples using the train_test_split functionality present in sklearn.model_selection module. Sklearn's train_test_split splits arrays into random train and test subsets. When we run the algorithm without specifying a random_state, we will get a different result every time, this is an expected behaviour. Random State controls the shuffling applied to the data before applying the split.

Here we are splitting data as 70% to training set and 30% to testing set using random_state = 1. After splitting training set has 1899 rows and testing set has 815 rows.

Decision Tree

A decision Tree is one of most popular and effective supervised learning technique for classification problem that equally works well with both categorical and quantitative variables. It is a graphical representation of all the possible solution to a decision that is based on certain condition.

The tree accuracy is heavily affected by the split point at decision node. Decision trees use different criteria to decide split on decision node to get two or more sub nodes. Gini criteria is one of the most popular criteria used for splitting a node into child nodes. The resultant sub nodes must increase in the homogeneity of data points also known as the purity of nodes with respect to target variable. The split decision is tested on all available variables and then the split with maximum purity sub nodes is get selected.

In python for building Decision Tree / CART model we have DecisionTreeClassifier under sklearn.tree module.

While building the decision tree model, the tree building activity continues until we get all terminal nodes as pure nodes. This will cause an overfitting of model. To avoid this overfitting, we can prune the tree by passing few parameters like

- max_depth - indicates max levels up to which trees can extend
- min_samples_leaf - indicate minimum number of samples to be present on each leaf.
- min_samples_split - indicate minimum number of samples to be required for splitting current node.

When we are not sure with the optimal value for each of the parameters, we can pass a list of values to each parameter. We create a dictionary with each parameter as a key and list of values to each key. Using the GridSearchCV functinality available in sklearn.model_selection module we pass the above dictionary and build the Decision Tree model. The resultant model will select the optimal values for each parameter and build the model. The optimal values set for each parameter will be stored in grid_search.best_params_ variable.

Out of the different list of values passed to each parameter we have got best values to each parameter as:

```
{'criterion': 'gini', 'max_depth': 10,'min_samples_leaf': 50,
'min_samples_split': 300}
```

Using these best params we proceed to build the Decision Tree model.
In the current built model, we will find how much percent each feature/column is important:

```
Agency_Code    0.654860
Sales          0.232013
Product Name   0.094996
Commision      0.011202
Duration       0.006928
Age            0.000000
Type           0.000000
Channel        0.000000
Destination    0.000000
```

Random Forest

Random Forest technique is an ensemble technique wherein we construct multiple models and take the average output of all the models to take final decision/make prediction. For constructing multiple models/decision trees using same dataset we go for boot strapped dataset where:

- Samples are randomly selected from the given dataset.
- Random samples selected with rows replacement.
- Random Samples selected with random subset of independent variables.

The Prediction strength of every individual tree must be high. The decision trees must not be correlated to each other.
Like decision Tree model takes in following few parameters:

- max_depth - indicates max levels up to which trees can extend
- min_samples_leaf - indicate minimum number of samples to be present on each leaf.
- min_samples_split - indicate minimum number of samples to be required for splitting current node.
- n_estimators - Number of Decision Trees to be constructed
- max_features - Number of columns randomly selected for decision making at each stage.

Here also we will make use of GridSearch by passing multiple values to each of above-mentioned parameters and construct the Random Tree model.

Out of the different list of values passed to each parameter we have got best values to each parameter as:

```
{'max_depth': 15, 'max_features': 5,'min_samples_leaf': 30,
 'min_samples_split': 60, 'n_estimators': 101}
```

Using these best params we proceed to build the Decision Tree model.
In the current built model, we will find how much percent each feature/column is important:

```
Agency_Code    0.404396
Sales          0.183341
Product Name   0.112619
Commision      0.090377
Duration       0.076120
Type           0.075081
Age            0.043274
Destination    0.014793
Channel        0.000000
```

Artificial Neural Networks

ANN is a machine learning algorithm that is roughly modelled around what is currently known about how the human brain functions. It models the relationship between a set of input signals and an output. It is like a biological brain response to stimuli from sensory inputs. ANN uses a network of artificial neurons or nodes to solve challenging learning problems. ANN has ability to learn, ability to generalize and is adaptable.

Descriptive statistics of Data

	Age	Agency_Code	Type	Commision	Channel	Duration	Sales	Product Name	Destination
count	2714.000000	2714.000000	2714.000000	2714.000000	2714.000000	2714.000000	2714.000000	2714.000000	2714.000000
mean	37.557848	1.338615	0.623803	10.110249	0.983051	41.779293	46.687524	1.559322	0.273766
std	9.211914	0.988286	0.484520	13.798433	0.129105	41.374383	38.316646	1.205857	0.597202
min	17.000000	0.000000	0.000000	0.000000	0.000000	-1.000000	0.000000	0.000000	0.000000
25%	31.250000	0.000000	0.000000	0.000000	1.000000	11.000000	20.000000	1.000000	0.000000
50%	36.000000	2.000000	1.000000	4.630000	1.000000	26.000000	32.000000	2.000000	0.000000
75%	42.750000	2.000000	1.000000	15.000000	1.000000	57.000000	62.000000	2.000000	0.000000
max	57.000000	3.000000	1.000000	43.087500	1.000000	141.000000	142.500000	4.000000	2.000000

Since different columns are having different weights we need to scale the data before feeding it to the ANN model. We will scale the data using StandardScaler technique.

Different parameters used in ANN are:

- hidden_layer_sizes - Number of neurons to be present at each layer of hidden layer.
- max_iter - Max number of iterations which a model is allowed to take for updating the synaptic weights
- solver -> The solving function used to calculate the optimal weights.
- tol -> Calculating the o/p at each iteration and the diff b/w iterations is less than this threshold value continuously for 10 iterations to stop the iteration.

Out of the different list of values passed to each parameter we have got best values to each parameter as:

```
{'activation': 'relu','hidden_layer_sizes': 200,'max_iter': 10000,'solver':
'adam','tol': 0.01}
```

Q 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.

- Confusion Matrix – A 2X2 tabular structure reflecting the performance of the model in four block.

Confusion Matrix	Predicted Positive	Predicted Negative
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

- Accuracy – How accurately / cleanly does the model classify the data points. Lesser the false predictions, more the accuracy
- Sensitivity /Recall – How many of the actual True data points are identified as True data points by the model.
- Precision – Among the points identified as Positive by the model, how many are really Positive.
- Specificity – How many of the actual Negative data points are identified as negative by the model.

Confusion Metrices

1) Decision Tree Model Training Set

```
array([[1162,  190],
       [ 256,  291]], dtype=int64)
```

2) Decision Tree Model Testing Set

```
array([[491,  78],
       [120, 126]], dtype=int64)
```

3) Random Forest Model Training Set

```
 array([[1223,  129],
       [ 282,  265]], dtype=int64)
```

4) Random Forest Model Testing Set

```
array([[516,  53],
       [145, 101]], dtype=int64)
```

5) ANN Model Training Set

```
array([[1352,    0],
       [ 547,    0]], dtype=int64)
```

6) ANN Model Testing Set

```
array([[569,   0],
       [246,   0]], dtype=int64)
```

Classification Report

1) Decision Tree Model Training Set

	precision	recall	f1-score	support
0	0.82	0.86	0.84	1352
1	0.60	0.53	0.57	547
accuracy			0.77	1899
macro avg	0.71	0.70	0.70	1899
weighted avg	0.76	0.77	0.76	1899

2) Decision Tree Model Testing Set

	precision	recall	f1-score	support
0	0.80	0.86	0.83	569
1	0.62	0.51	0.56	246
accuracy			0.76	815
macro avg	0.71	0.69	0.70	815
weighted avg	0.75	0.76	0.75	815

3) Random Forest Model Training Set

	precision	recall	f1-score	support
0	0.81	0.90	0.86	1352
1	0.67	0.48	0.56	547
accuracy			0.78	1899
macro avg	0.74	0.69	0.71	1899
weighted avg	0.77	0.78	0.77	1899

4) Random Forest Model Testing Set

	precision	recall	f1-score	support
0	0.78	0.91	0.84	569
1	0.66	0.41	0.51	246
accuracy			0.76	815
macro avg	0.72	0.66	0.67	815
weighted avg	0.74	0.76	0.74	815

5) ANN Model Training Set

	precision	recall	f1-score	support
0	0.71	1.00	0.83	1352
1	0.00	0.00	0.00	547
accuracy			0.71	1899
macro avg	0.36	0.50	0.42	1899
weighted avg	0.51	0.71	0.59	1899

6) ANN Model Testing Set

	precision	recall	f1-score	support
0	0.70	1.00	0.82	569
1	0.00	0.00	0.00	246
accuracy			0.70	815
macro avg	0.35	0.50	0.41	815
weighted avg	0.49	0.70	0.57	815

Accuracy Score

1) Decision Tree Model Training Set – 0.765
2) Decision Tree Model Testing Set – 0.757
3) Random Forest Model Training Set – 0.783
4) Random Forest Model Testing Set – 0.757
5) ANN Model Training Set – 0.661
6) ANN Model Testing Set – 0.647

AUC and ROC Curve

1) Decision Tree Model Training Set
AUC – 0.784

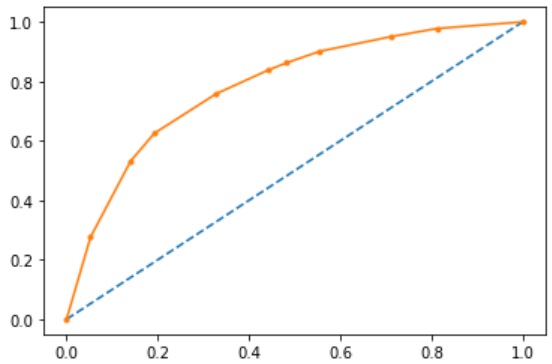

Figure 21 : CART Model ROC Curve for Training Data

2) Decision Tree Model Testing Set
AUC – 0.789

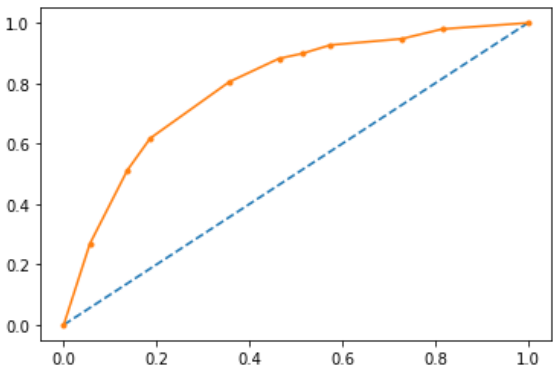

Figure 22 : CART Model ROC Curve for Testing Data

3) Random Forest Model Training Set
AUC – 0.817

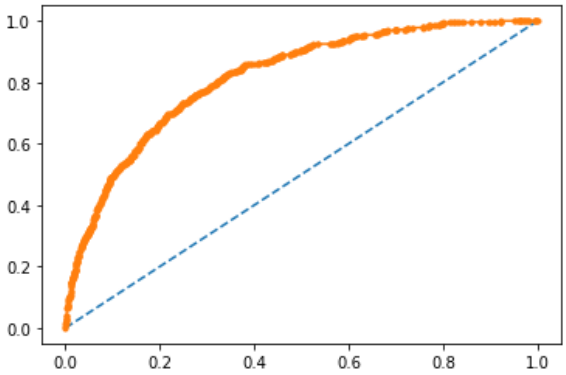

Figure 23 : Random Forest Model ROC Curve for Training Data

4) Random Forest Model Testing Set
AUC – 0.798

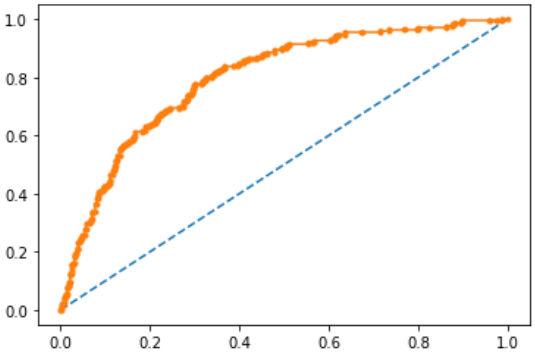

Figure 24 : Random Forest Model ROC Curve for Testing Data

5) ANN Model Training Set
AUC – 0.595

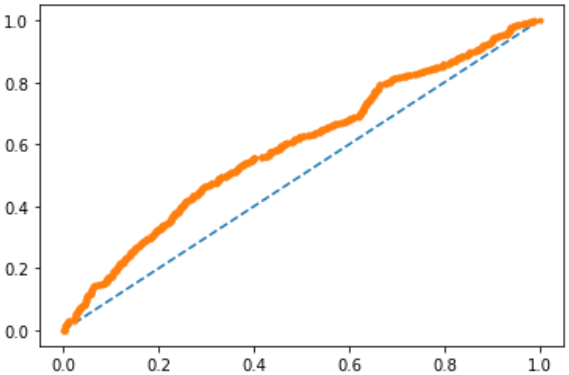

Figure 25 : ANN Model ROC Curve for Training Data

6) ANN Model Testing Set
AUC – 0.598

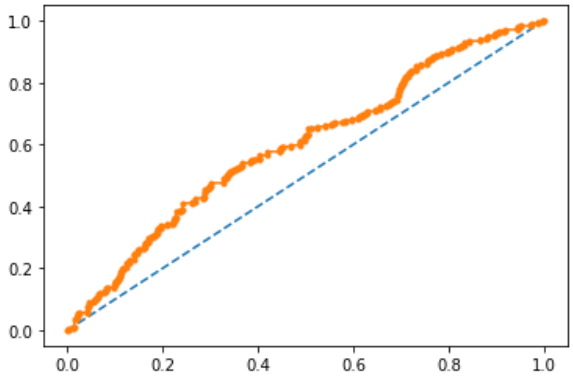

Figure 26 : ANN Model ROC Curve for Testing Data

Comments on analysing above data:

- Accuracy score, AUC, f1-score, recall, precision for both training and testing data is almost similar in all 3 models.
- Compared to all 3 models Random Forest model has good accuracy.
- AUC in ANN model is least and best in Random Forest model.
- F1-score in ANN model is least and best in Random Forest model.
- Recall is overfitted in ANN and good in Random Forest model.
- Precision is overfitted in ANN and good in Random Forest model.

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

Comparing all the stats of all 3 models in following table.

Model		Accuracy	F1-score	AUC
CART Model	Training Data	0.77	0.57	0.78
	Testing Data	0.76	0.56	0.79
Random Forest	Training Data	0.78	0.56	0.81
	Testing Data	0.76	0.51	0.8
ANN Model	Training Data	0.66	0	0.59
	Testing Data	0.64	0	0.59

Table 13 : Comparison of all 3 models

Here f1-score values corresponding to '1' are tabulated.

All the stats are similar for training and testing data. Based on accuracy, f1-score and AUC Random Forest model is having better values compared to other 2 models for given dataset.

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

In the given data set we have 70.7% of data as claimed state as 'No' and only 29.3% od data as claimed state as 'Yes'. So data is not equally distributed. It would be better if we get additional data having equally distributed target column and then proceed on model building and evaluating activity.

Recommendations:

- Capture the customer experience who have got the benefits from insurance claims and showcase them publicly to attract more customers thereby increasing the business.
- Educate and Market various available insurance plans.
- Showcase and market about the online channel availability so that need not visit the office personally.
- Target more and more customers visiting existing destinations and additional destinations.
- From the data it is evident 62% of claims are coming from Travel Agency so need to check the genuinity of claims coming from Travel agency.
- Tie up with the International airlines and market about the travel insurance to all their customers and increase the business.
- More claims are coming from age group of 35 – 40 so scrutinize such claims thoroughly.
- Process the claims faster.
- More sales happen via Agency than Airlines and the data shows the claim are processed more at Airline need to figure out what is lacking in claims made via Agency.