DATA MINING PROJECT

Pooja Kabadi

Table of Contents:

Problem 1: Clustering.....3

1.1 Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis). 3

1.2 Do you think scaling is necessary for clustering in this case? Justify 11

1.3 Apply hierarchical clustering to scaled data. Identify the number of optimum clusters using Dendrogram and briefly describe them. 12

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. 14

1.4 Describe cluster profiles for the clusters defined. Recommend different promotional strategies for different clusters. 18

Problem 2: CART-RF-ANN.....21

2.1 Read the data, do the necessary initial steps, and exploratory data analysis (Univariate, Bi-variate, and multivariate analysis). 21

2.2 Data Split: Split the data into test and train, build classification model CART, Random Forest, Artificial Neural Network 29

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. 34

2.4 Final Model: Compare all the models and write an inference which model is best/optimized. 40

2.5 Inference: Based on the whole Analysis, what are the business insights and recommendations 42

List of Figures:

Figure 1: Boxplot & Displot of Spending …… …..5

Figure 2: Boxplot & Displot of Advance payments…… ..5

Figure 3: Boxplot & Displot of Probability of full payment…… ..5

Figure 4: Boxplot & Displot of Current Balance……...6

Figure 5: Boxplot & Displot of Credit limit……6

Figure 6: Boxplot & Displot of Minimum payment amount……..…6

Figure 7: Boxplot & Displot of Maximum spent in single shopping..……7

Figure 8: Joint plot of spending vs probability of full pay……..8

Figure 9: Joint plot of Credit limit vs Advance payments……..……..8

Figure 10: Pair plot of clustering……8

Figure 11: Heat map of clustering……9

Figure 12: Box plot of clustering……..10

Figure 13: Plots to compare prior and after scaling data……..……..…...11

Figure 14: Dendrogram plot ……..12

Figure 15: Truncated Dendrogram……....13

Figure 16: Bar graph of Hierarchical clustering …….....14

Figure 17: Scatter plot of Hierarchical clustering …….....14

Figure 18: Elbow curve for K-means clustering ……...15
Figure 19: Bar graph of K-means clustering …….…….....18
Figure 20: Scatter plot of K-means clustering …….....18
Figure 21: Scatter plot of Hierarchical clustering ……....18
Figure 22: Scatter plot of K-means clustering ……..…….....19
Figure 23: Boxplot & Displot of Age ……..……..23
Figure 24: Boxplot & Displot of Commission……..……..……23
Figure 25: Boxplot & Displot of Duration…… ……24
Figure 26: Boxplot & Displot of Sales……...24
Figure 27: Count plot of categorical variables……..24
Figure 28: Boxplot and swarm plot of Agency code vs Sales……...25
Figure 29: Boxplot and swarm plot of Type vs Sales …….…..26
Figure 30: Boxplot and swarm plot of Channel vs Sales…….…..26
Figure 31: Boxplot and swarm plot of Product Name vs Sales……..….27
Figure 32: Boxplot and swarm plot of Destination vs Sales…….……..27
Figure 33: Pair plot of problem 2 ……28
Figure 34: Heat map of problem 2 ……29
Figure 35: Feature importance plot of CART model……31
Figure 36: Feature importance plot of Random Forest model……33
Figure 37: Confusion matrix for CART Model……..36
Figure 38: ROC curve for CART Model……...36
Figure 39: Confusion matrix for Random Forest Model……37
Figure 40 : ROC curve for Random Forest Model……38
Figure **41**: Confusion matrix for Artificial Neural Network Model……...38
Figure 42 : ROC curve for Artificial Neural Network Model……...39
Figure 43 : Comparison of ROC curve and AUC score of all models for train data……...40
Figure 44 : Comparison of ROC curve and AUC score of all models for train data……...41

List of Tables:

Table 1: Inferences of Univariate Data visualization of clustering……..……...…7
Table 2: Proportions of Target variable – Claimed……...30
Table 3 : Model Performance for CART model……37
Table 4 : Model Performance for Random Forest model……38
Table 4 : Model Performance for Artificial Neural Network model……39

Problem 1: Clustering

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

Data Dictionary for Market Segmentation:

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)

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

Exploratory Data Analysis:

Read and view data

	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
5	12.70	13.41	0.8874	5.183	3.091	8.456	5.000
6	12.02	13.33	0.8503	5.350	2.810	4.271	5.308
7	13.74	14.05	0.8744	5.482	3.114	2.932	4.825
8	18.17	16.26	0.8637	6.271	3.512	2.853	6.273
9	11.23	12.88	0.8511	5.140	2.795	4.325	5.003

Checking for the information of features:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 210 entries, 0 to 209
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   spending                      210 non-null    float64
 1   advance_payments              210 non-null    float64
 2   probability_of_full_payment   210 non-null    float64
 3   current_balance               210 non-null    float64
 4   credit_limit                  210 non-null    float64
 5   min_payment_amt               210 non-null    float64
 6   max_spent_in_single_shopping  210 non-null    float64
dtypes: float64(7)
memory usage: 11.6 KB
```

Checking the Skewness and Kurtosis:

```
spending                      0.399889
advance_payments              0.386573
probability_of_full_payment  -0.537954
current_balance               0.525482
credit_limit                  0.134378
min_payment_amt               0.401667
max_spent_in_single_shopping  0.561897
dtype: float64
```

```
spending                      -1.084266
advance_payments              -1.106703
probability_of_full_payment   -0.140315
current_balance               -0.785645
credit_limit                  -1.097697
min_payment_amt               -0.066603
max_spent_in_single_shopping  -0.840792
dtype: float64
```

Checking the description of dataset:

	count	mean	std	min	25%	50%	75%	max
spending	210.0	14.847524	2.909699	10.5900	12.27000	14.35500	17.305000	21.1800
advance_payments	210.0	14.559286	1.305959	12.4100	13.45000	14.32000	15.715000	17.2500
probability_of_full_payment	210.0	0.870999	0.023629	0.8081	0.85690	0.87345	0.887775	0.9183
current_balance	210.0	5.628533	0.443063	4.8990	5.26225	5.52350	5.979750	6.6750
credit_limit	210.0	3.258605	0.377714	2.6300	2.94400	3.23700	3.561750	4.0330
min_payment_amt	210.0	3.700201	1.503557	0.7651	2.56150	3.59900	4.768750	8.4560
max_spent_in_single_shopping	210.0	5.408071	0.491480	4.5190	5.04500	5.22300	5.877000	6.5500

Checking for duplicates in this dataset:

```
# Are there any duplicates?
dups = df_clust.duplicated()
print('Number of duplicate rows = %d' % (dups.sum()))
df_clust[dups]
```

```
Number of duplicate rows = 0
```

Observations:

- Dataset has 7 columns and 210 rows
- The entire dataset is of float data type.
- The dataset is a sample that summarizes the activities of users during the past few months.
- There are no null values and duplicate rows in the dataset.
- Skewness and Kurtosis is also calculated for each column, Data with high skewness indicates lack of symmetry and high value of kurtosis indicates heavily tailed data.
- Based on summary descriptive, the data looks good, we see that for most of the variables the mean/medium are nearly equal.
- Standard Deviation is high for 'spending' variable.
- The minimum and maximum values of all features are not very far from the median. So, there would be very less or no outliers in most of the features.

Data Visualization:

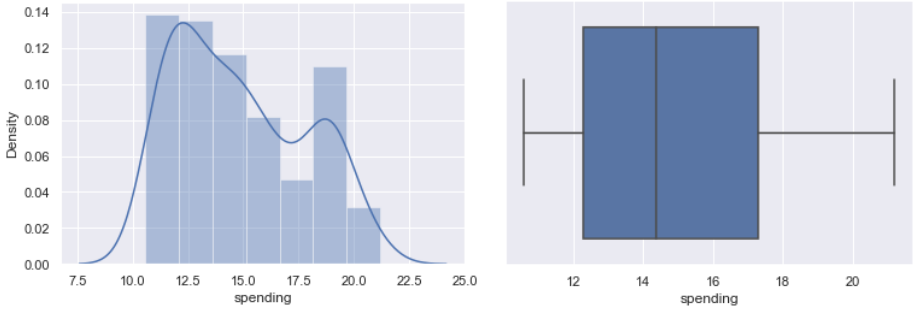
Univariate Analysis:

Let us define a function 'univariateAnalysis_numeric' to display information as part of univariate analysis of numeric variables. The function will accept column name and number of bins as arguments.

The function will display the statistical description of the numeric variable, histogram or distplot to view the distribution and the box plot to view 5-point summary and outliers if any.

Spending: Amount spent by the customer per month (in 1000s) *(Figure 1: Boxplot & Displot of Spending)*

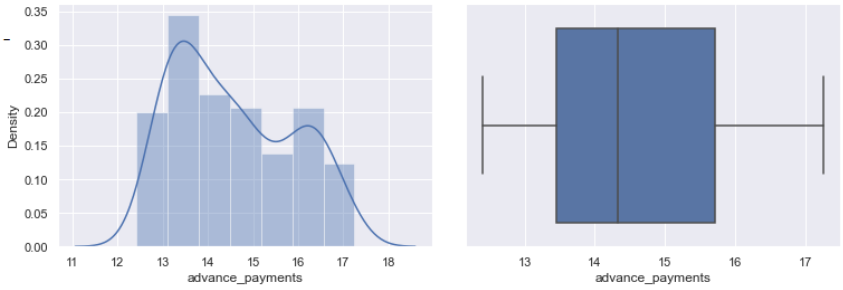
```
Description of spending
-----------------------
count    210.000000
mean      14.847524
std        2.909699
min       10.590000
25%       12.270000
50%       14.355000
75%       17.305000
max       21.180000
```



- From the above graphs, we can infer that mean spending of users is around 14.84K with the minimum of 10.59K and maximum of 21.18K
- The distribution of 'spending' is slightly right skewed with skewness value of 0.399
- The dist plot shows the distribution of data from 10 to 22 (1000's)
- The box plot of the spending variable shows no outliers.

Advance_payments: Amount paid by the customer in advance by cash (in 100s)
(Figure 2: Boxplot & Displot of Advance_payments)

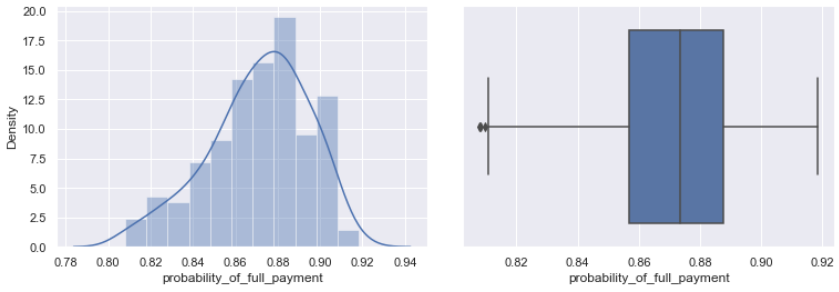
```
Description of advance_payments
-------------------------------
count    210.000000
mean      14.559286
std        1.305959
min       12.410000
25%       13.450000
50%       14.320000
75%       15.715000
max       17.250000
```



- From the above graphs, we can infer that mean of advance payments by users is around 1.455K with the minimum of 1.241K and maximum of 1.725K
- The distribution of 'advance_payments' is slightly right skewed with skewness value of 0.386
- The dist plot shows the distribution of data from 12 to 17 (100's)
- The box plot of the advance payment's variable shows no outliers.

Probability_of_full_payment: Probability of payment done in full by the customer to the bank
(Figure 3: Boxplot & Displot of Probability_of_full_payment)

```
Description of probability_
---------------------------
count    210.000000
mean       0.870999
std        0.023629
min        0.808100
25%        0.856900
50%        0.873450
75%        0.887775
max        0.918300
```

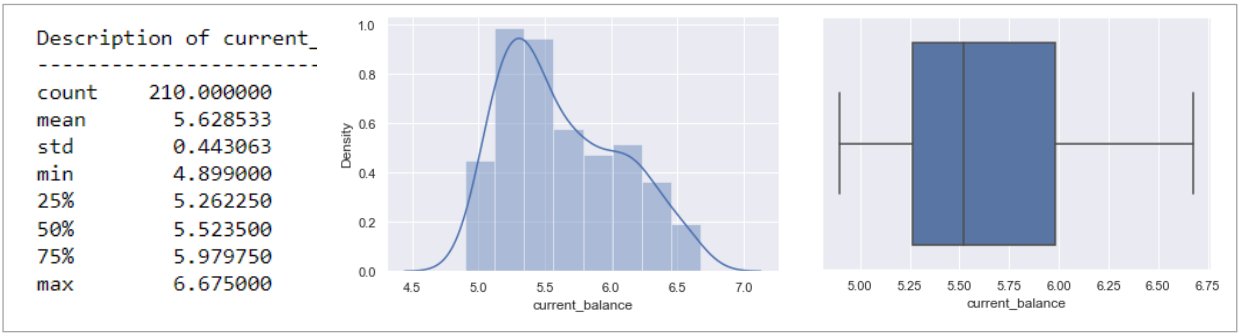


- From the above graphs, we can infer that mean probability of full payment by users is 0.87 with the minimum of 0.80 and maximum of 0.91.

- The distribution of 'probability_of_full_payment' is left skewed with skewness value of -0.537
- The dist plot shows the distribution of data from 0.80 to 0.92
- The box plot of the probability of full payment variable shows few outliers.
- The Probability values is good above 80%

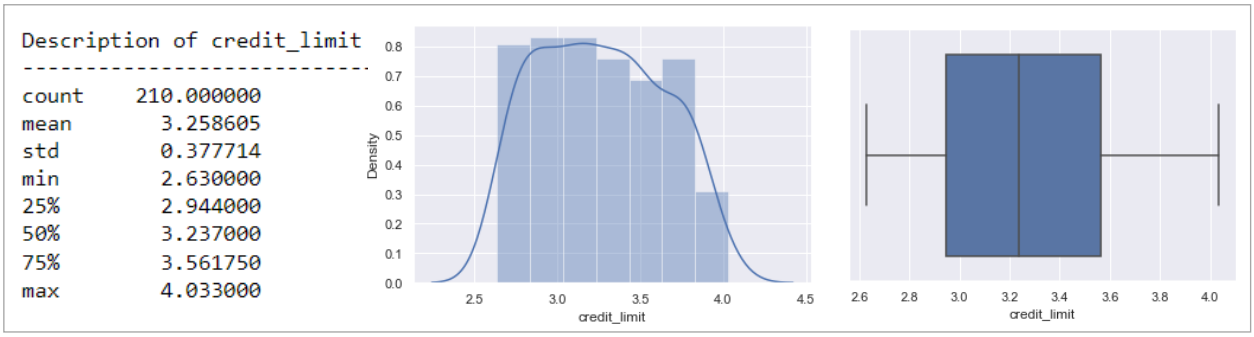
Current_balance: Balance amount left in the account to make purchases (in 1000s)

(Figure 4: Boxplot & Displot of Current_balance)



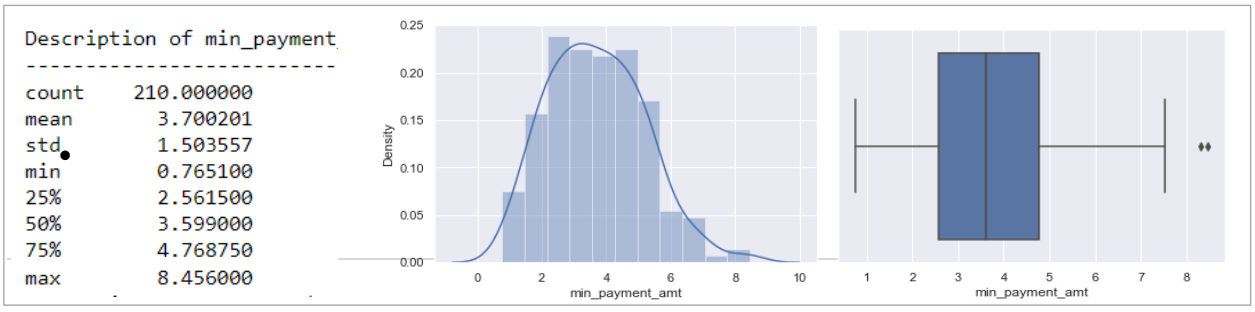
- From the above graphs, we can infer that mean current balance of users is 5.628K with the minimum of 4.899K and maximum of 6.675K
- The distribution of 'current_balance' is right skewed with skewness value of 0.525
- The dist plot shows the distribution of data from 5.0 to 6.5 (1000's)
- The box plot of the current balance variable shows no outliers.

Credit_limit: Limit of the amount in credit card (10000s) *(Figure 5: Boxplot & Displot of Credit_limit)*



- From the above graphs, we can infer that mean credit limit of users is 32.58K with the minimum of 26.3K and maximum of 40.33K
- The distribution of 'current_balance' is slightly right skewed with skewness value of 0.1343
- The dist plot shows the distribution of data from 2.5 to 4.0 (10,000's)
- The box plot of the credit limit variable shows no outliers.

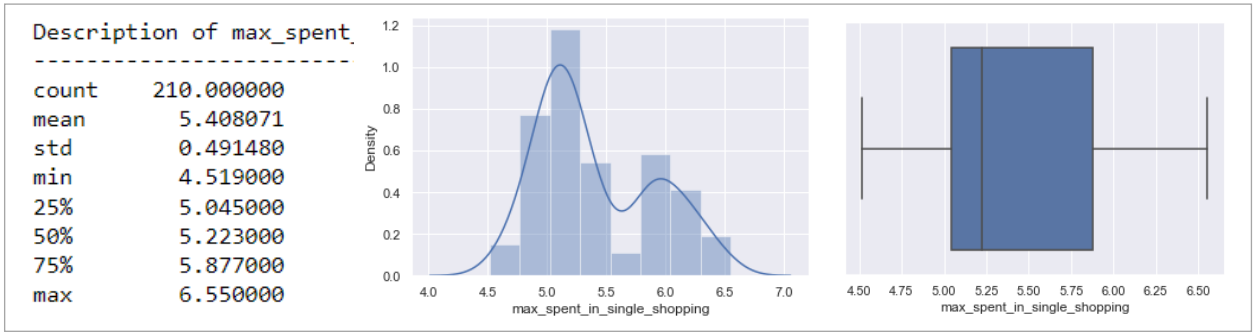
Min_payment_amt: minimum paid by the customer while making payments for purchases made monthly (in 100s) *(Figure 6: Boxplot & Displot of Min_payment_amt)*



- From the above graphs, we can infer that mean of minimum payment amount by the customer while making payments for purchases made monthly is 0.37K with the minimum of 0.0765K and maximum of 0.845K
- The distribution of 'min_payment_amt' is right skewed with skewness value of 0.4016
- The dist plot shows the distribution of data from 2 to 8 (100's)
- The box plot of the min payment amount variable shows few outliers.

Max_spent_in_single_shopping: Maximum amount spent in one purchase (in 1000s)

(Figure 7: Boxplot & Displot of Max_spent_in_single_shopping)



- From the above graphs, we can infer that mean of max spent in single shopping is 5.408K with the minimum of 4.519K and maximum of 6.550K
- The distribution of 'max_spent_in_single_shoppingt' is right skewed with skewness value of 0.5618
- The dist plot shows the distribution of data from 4.5 to 6.5(1000's)
- The box plot of the max spent in single shopping variable shows no outliers.

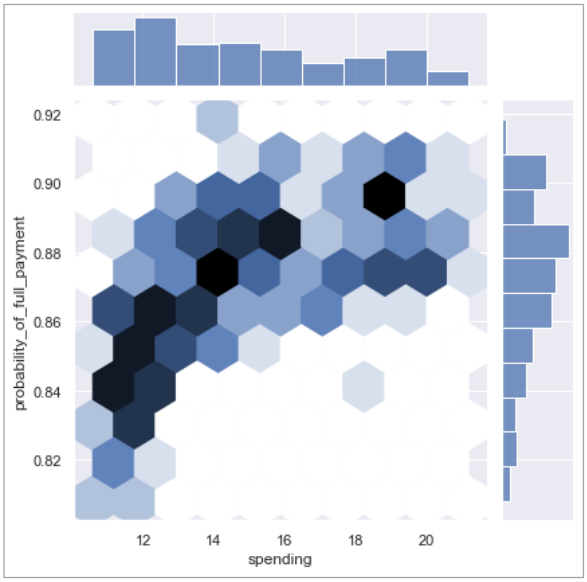
We can summarize the above graphs and descriptions into a table to observe type of distribution, skewness and outliers further. Find the table below for the same:

(Table 1: Inferences of Univariate Data visualization)

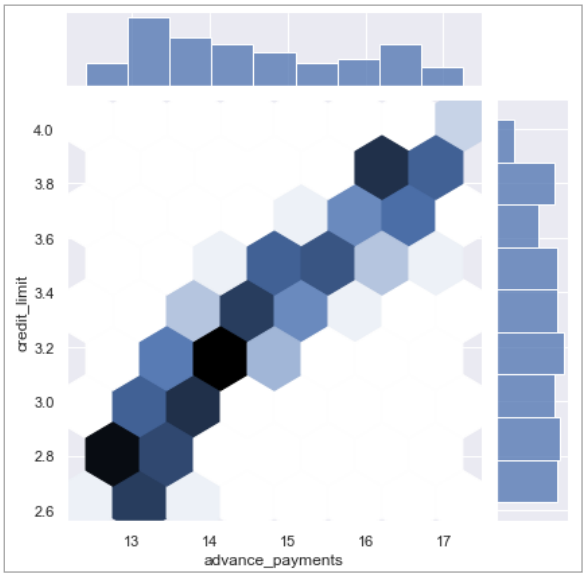
Sl.no	Features	Distribution	Skewness	Outliers
1	Spending	Right Skewed	+0.40	No
2	Advance Payments	Right Skewed	+0.39	No
3	Prob of full payment	Left Skewed	-0.54	Lower Outlier
4	Current Balance	Right Skewed	+0.53	No
5	Credit Limit	Almost Normal	+0.13	No
6	Min Payment Amount	Almost Normal	+0.40	Upper Outlier
7	Max spent in single shopping	Right Skewed	+0.56	No

- The preceding table indicates that each variable's distribution is approximately normal, and that the skewness is not excessive.
- There are outliers in 2 features only i.e. Prob of full payment and Min payment Amount.
- The proportion of outliers in the data is not particularly large, as seen by the boxplots. It's virtually insignificant.
- However, we will still be treating the outliers for a better analysis later as we go on.

Bivariate Analysis:



(Figure 8: Joint plot of spending vs probability of full pay)

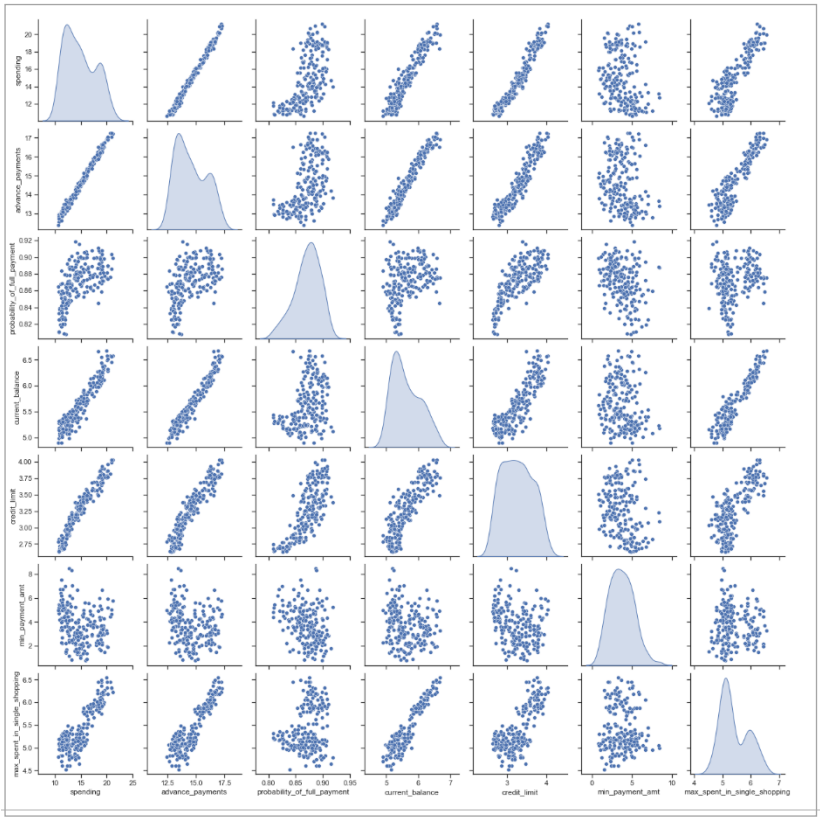


(Figure 9: Joint plot of credit limit vs advance payment)

- From above graph we can infer that, users who spend less in the range of 10K-12K, have high probability of full payment.
- The spending and probability of full payment are inversely proportional to each other, lesser the spending higher the probability of full payment and higher the spending the probability of full payment is very less.
- From the above graph we can infer that, the advance payments and credit limit are linearly proportional.

Pair plot

(Figure 10: Pair plot of clustering)

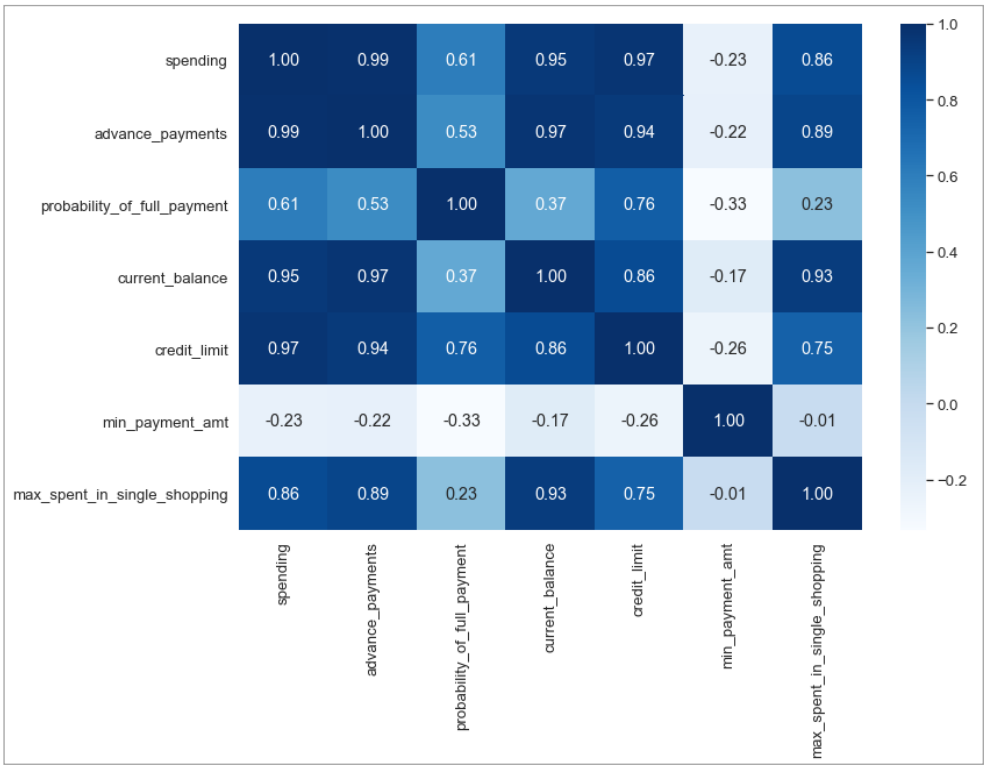


- A pair plot gives us correlation graphs between all numerical variables in the dataset. Thus, from the graphs we can identify the relationships between all numerical variables.
- The almost 45-degree inclined lines in above Pair plot suggest a very high correlation between the two variables and are having linear proportion between the variables.

Multivariate Analysis:

Heatmap:

A heatmap gives us the correlation between numerical variables. If the correlation value is tending to 1, the variables are highly positively correlated whereas if the correlation value is close to 0, the variables are not correlated. Also, if the value is negative, the correlation is negative. That means, higher the value of one variable, the lower is the value of another variable and vice-versa.



(Figure 11: Heatmap of clustering)

		correlation
spending	advance_payments	0.994341
advance_payments	current_balance	0.972422
credit_limit	spending	0.970771
spending	current_balance	0.949985
credit_limit	advance_payments	0.944829
max_spent_in_single_shopping	current_balance	0.932806
advance_payments	max_spent_in_single_shopping	0.890784
spending	max_spent_in_single_shopping	0.863693
current_balance	credit_limit	0.860415
probability_of_full_payment	credit_limit	0.761635
max_spent_in_single_shopping	credit_limit	0.749131
spending	probability_of_full_payment	0.608288
advance_payments	probability_of_full_payment	0.529244
current_balance	probability_of_full_payment	0.367915
probability_of_full_payment	min_payment_amt	0.331471

Observation
As per the Heat Map, we can conclude that the following variables are highly correlated:

- Spending and advance_payments, spending and current_balance, spending and credit_limit
- Advance_payment and current_balance, advance_payment and credit limit
- Current balance and max spent in single shopping

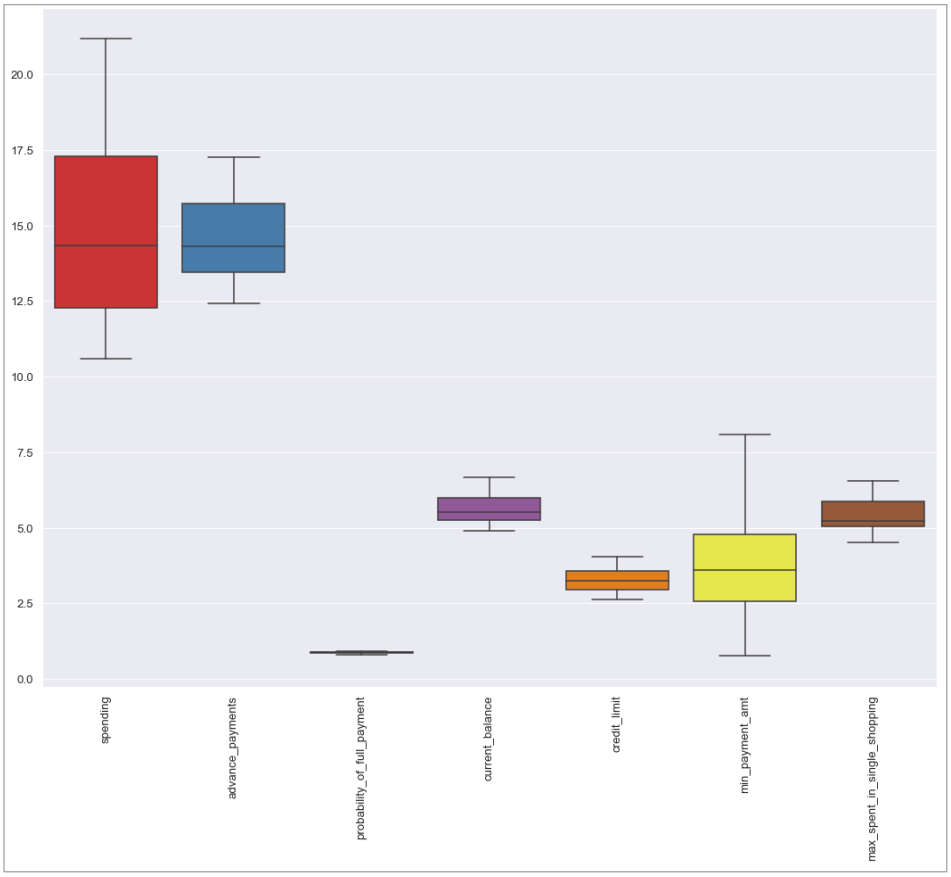
Inferences

- By this we can conclude that the customers who are spending very high have a higher current balance and high credit limit. Advance payments and maximum expenditure done in single shopping are done by majority of those customers who have high current balance in their bank accounts.
- Probability of full payments are higher for those customers who have a higher credit limit.
- Minimum payment amount is not correlated to any of the variables; hence, it is not affected by any changes in the current balance or credit limit of the customers.

Outlier Treatment *(Figure 12: Boxplot of clustering)*

Outlier's present should be treated as Clustering results are affected by the presence of outliers. There is very few outliers present in Prob_of_full_payment and Min_Payment_Amt features. So, we will treat the values by imputing the outliers to the upper range if it is an upper outlier and to lower range if it is a lower outlier.

Outliers are treated and **Box plot is plotted to cross check the treatment.**



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

Yes, scaling is necessary for clustering in this case.

The clustering model works on the distance-based calculations. Mostly Euclidean distance. If the data is unscaled, the distances calculated for each variable would be on different scales. Hence, the computation won't be making much sense.

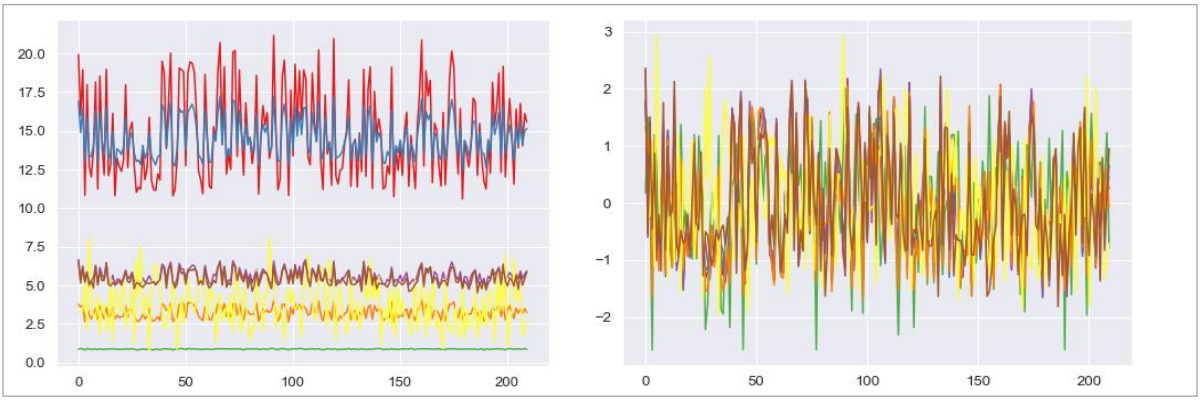
Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values as higher weightage and consider smaller values as the lower weightage, regardless of the unit of the values.

For the data given, scaling is required as all the variables are expressed in different units such as spending in 1000's, advance payments in 100's and credit limit in 10000's, whereas probability is expressed as fraction or decimal values.

Since the other values expressed in higher units will outweigh probabilities and can give varied results hence it is important to Scale the data using Standard Scaler and therefore normalise the values where the means will be 0 and standard deviation 1.

Scaling of data is done using importing a package called StandardScaler from sklearn. preprocessing. For further clustering of dataset, we will be using the scaled data.

Plots prior to scaling and after scaling dataset: *(Figure 13: Plots to compare prior and after scaling data)*



Before Scaling:
Before scaling our dataset is distributed as shown in the figure above. Some variables are plotted higher as they must be having higher absolute values while some are near the zero line as their absolute values are lower than the other variables.

Even if the values for these variables are lower, they play significant role in the dataset, we cannot ignore them. Therefore, scaling is necessary.

For this dataset, we would be performing Standard Scaler function to scale the dataset. StandardScaler follows Standard Normal Distribution (SND). Therefore, it makes mean = 0 and scales the data to unit variance.

After Scaling:
We can see that all variables are scaled now and the values are close to each other. If we now check the plot of the scaled dataset, we would find that all variables are distributed similar to each other and all variables would be significant.

We can see now that all variables are scaled to have a mean tending to 0 and standard deviation to one. Therefore, scaling is very important for this dataset.

Output of scaled dataset:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping
0	1.754355	1.811968	0.177628	2.367533	1.338579	-0.298625	2.328998
1	0.393582	0.253840	1.505071	-0.600744	0.858236	-0.242292	-0.538582
2	1.413300	1.428192	0.505234	1.401485	1.317348	-0.220832	1.509107
3	-1.384034	-1.227533	-2.571391	-0.793049	-1.639017	0.995699	-0.454961
4	1.082581	0.998364	1.198738	0.591544	1.155464	-1.092656	0.874813
5	-0.739821	-0.882135	0.696692	-1.007977	-0.444794	2.938945	-0.832274
6	-0.974080	-0.943539	-0.881773	-0.630155	-1.190520	0.384751	-0.204099
7	-0.381541	-0.390903	0.143591	-0.331518	-0.383756	-0.513228	-1.189192
8	1.144591	1.305384	-0.311654	1.453520	0.672468	-0.566208	1.764048
9	-1.246235	-1.288937	-0.847736	-1.105261	-1.230328	0.420965	-0.826156

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

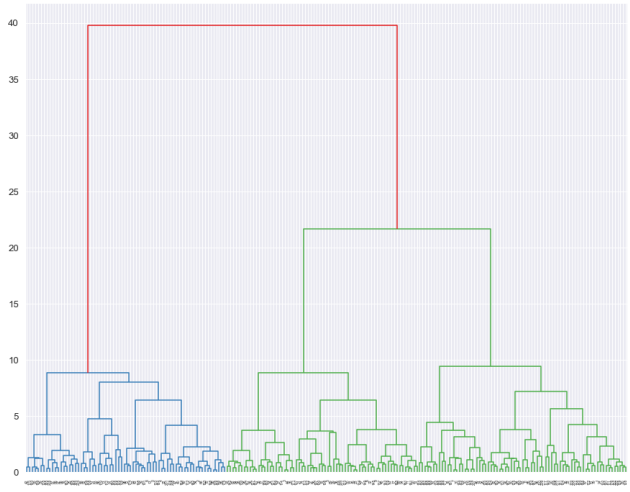
Hierarchical clustering is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other. There are two types of hierarchical clustering, Divisive and Agglomerative.

We now perform Hierarchical clustering on the scaled dataset using ward linkage method. Ward's linkage method joins records and clusters together progressively to produce larger and larger clusters. Ward's method makes sure to maintain minimum within cluster variation.

After linkage of the scaled data, we create a Dendrogram. A Dendrogram is a treelike diagram that summarizes the process of clustering. Similar records are joined by lines whose vertical length reflects the distance between the records. By choosing a cut-off distance on the y-axis, a set of clusters is created. Let us see the Dendrogram created for our linkage.

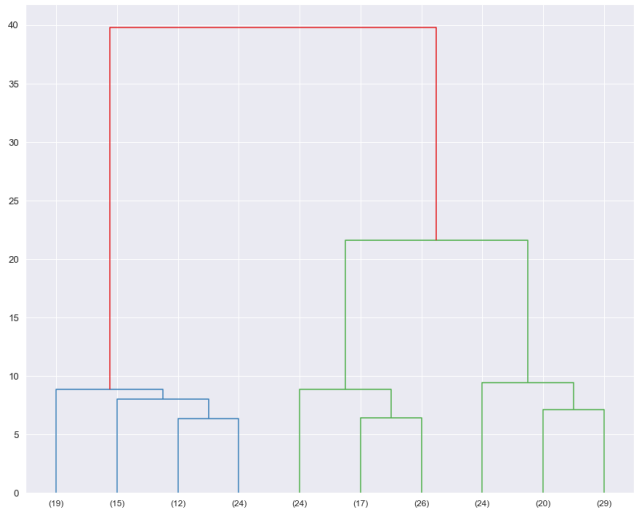
For the dataset in question, we will be using Agglomerative Hierarchical Clustering method to create optimum clusters and categorising the dataset on the basis of these clusters.

Dendrogram is created using the scaled data, and using the 'Ward' linkage method.



(Figure 14: Dendrogram)

- Dendrogram is created using the scaled data, and using the **'Ward' linkage method.**
- Firstly, imported the package dendrogram, linkage from scipy.cluster.hierarchy. Using this function, we have created a dendrogram.
- The above Dendrogram, the colour combination shows 2 clusters.
- Two clusters really do not make much business impact as it is kind of implicit. For example, for the dataset, it is imperative that there will be some high spenders and some low spenders.
- Further, the **Dendrogram is truncated to the last 10 linkage** to find the optimal number cluster through which we can solve our business objective.



- This truncated Dendrogram gives us a clearer picture that the optimum number of clusters for this dataset should be 3.
- Now, we can understand all the data points have clustered into 3 clusters which is the optimal number of clusters for business problem
- Next mapping these clusters to the dataset, we use **fcluster()** to find the optimal number of clusters. With Ward link method, and criterion as 'maxclust' with value as 3 we find the optimal number of clusters.
- Also, when you look at the dendrogram, it seems that 2 clusters would be optimal. But that is not the only way to interpret. With **distance criterion** and value as 20-25, we see that that the optimal number of clusters shall be 3.

(Figure 15: Truncated Dendrogram)

Mapping the Clusters to Dataset using fclusters, criterion='maxclust' and appending the respective clusters to data:

Now we map each record to the one of the three clusters by using the **fcluster()** function. We pass our linkage in fcluster and use the criterion of **'maxclust'** along with minimum threshold value = 3. We finally get an array with cluster for each record of dataset. We add this array to the original dataset and get the cluster assigned to each record. The final dataset looks as below:

	spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	clusters
0	19.94	16.92	0.875200	6.675	3.763	3.252000	6.550	1
1	15.99	14.89	0.906400	5.363	3.582	3.336000	5.144	3
2	18.95	16.42	0.882900	6.248	3.755	3.368000	6.148	1
3	10.83	12.96	0.810588	5.278	2.641	5.182000	5.185	2
4	17.99	15.86	0.899200	5.890	3.694	2.068000	5.837	1
5	12.70	13.41	0.887400	5.183	3.091	8.079625	5.000	2
6	12.02	13.33	0.850300	5.350	2.810	4.271000	5.308	2
7	13.74	14.05	0.874400	5.482	3.114	2.932000	4.825	3
8	18.17	16.26	0.863700	6.271	3.512	2.853000	6.273	1
9	11.23	12.88	0.851100	5.140	2.795	4.325000	5.003	2

Cluster Frequency and Profiling:

Cluster profiling is the main idea behind doing the process we have performed till now. For cluster profiling, we would group the dataset by the clusters column and calculate the mean of each variable. We can also calculate the frequency of each cluster and see how many records are grouped under each cluster.

The python output for the same is as below:

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	11.872388	13.257015	0.848155	5.238940	2.848537	4.940302	5.122209	67
3	14.199041	14.233562	0.879190	5.478233	3.226452	2.612181	5.086178	73

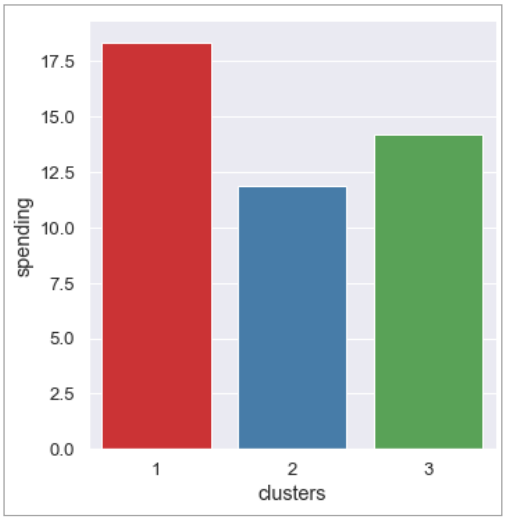
According to the table above, we get 3 clusters with almost equal number of records 3 cluster solution makes sense based on the spending pattern

Cluster 1 has 70 records under it with a mean spending of 18.37.

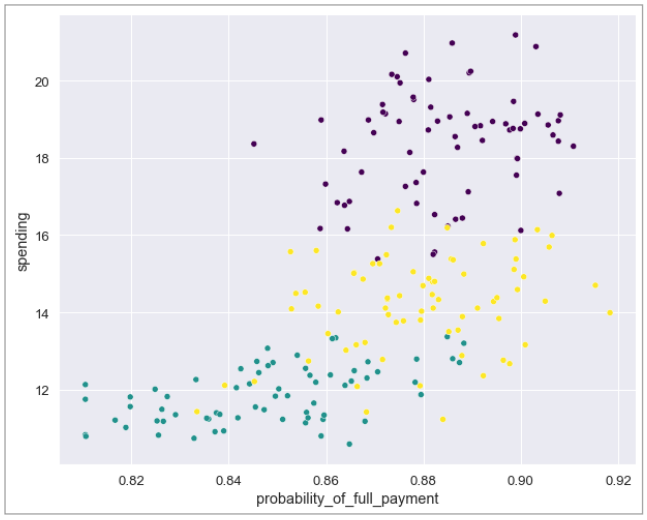
Cluster 2 has 67 records under it with a mean spending of 11.87.

Cluster 3 has 73 records under it with a mean spending of 14.19.

Visualizing the spending pattern of each cluster with the help of a bar chart and scatter plot:



(Figure 16: Bar graph of Hierarchical clustering)



(Figure 17: Scatterplot of Hierarchical clustering)

Based on the observations and the bar graph, we can profile these clusters as follows:

Cluster 1 – High Spending

Cluster 2 – Low Spending

Cluster 3 – Medium Spending

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 a non-hierarchical approach to forming good clusters where we pre-specify a desired number of clusters. In this technique, we randomly take centroids in our dataset and calculate the within-cluster distance. The datapoints are clustered in such a way that distance between the assigned centroid is least as compared to distance with other centroids.

The centroids are then changed accordingly to minimize the measure of dispersion within the clusters. This process is repeated till clusters with minimum dispersion within them are created.

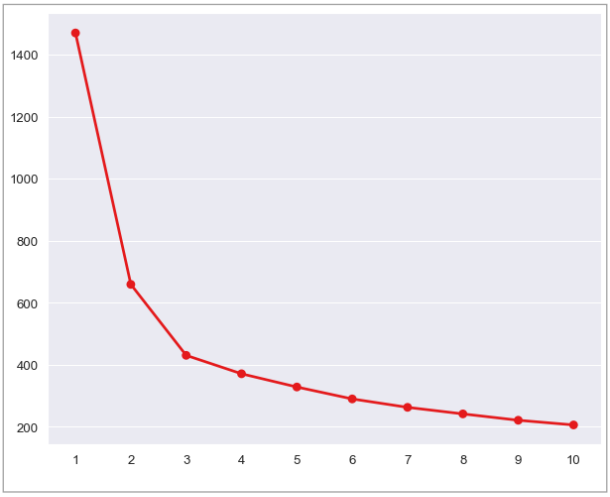
We start with taking different number of clusters in our python notebook and calculating within sum of square (WSS) values for each. Find the python output as below.

```
wss
[1469.999999999999,
 659.14740095485,
 430.298481751223,
 370.86859623942047,
 328.20399270811714,
 289.7476634411696,
 262.59831712590284,
 241.4383087594363,
 221.00431051620964,
 205.8191561657478]
```

Calculating within sum of squares variance for the clusters ranging from 1 to 11 for optimal K – value : Elbow Method

The above image shows the WSS values starting with 1 cluster till 10 clusters. The optimum number of clusters is obtained when the change in WSS score is not very significant. From the above numbers, we can feel that till 3 clusters the WSS change is significant after that it is not that much. So, 3 clusters might be optimum for this dataset.

Plotting the Elbow Curve *(Figure 18: Elbow curve for K-means clustering)*



The ideal number of clusters, according to the **elbow curve**, is where we can observe the last steep turn. We can observe that 3 is the final location in the plot with a significant cut. As a result of K-Means clustering, the total number of clusters for this dataset is 3.

Silhouette score

The silhouette score will be used to determine if the number of clusters taken into account is optimal. It's essentially a strategy for evaluating models indirectly.

In this technique, we calculate the Silhouette width for each record.

Sil_width = (b-a)/Max(a,b)

On calculating the Silhouette score for our dataset, it comes out to be **0.4**

And we also calculate all the Silhouette Widths in the dataset.

```
array([0.5732776 , 0.36556355, 0.63709249, 0.515595  , 0.36097201,
       0.22152508, 0.47529542, 0.36025848, 0.51938329, 0.53443903,
       0.46599399, 0.12839864, 0.39177784, 0.52379458, 0.11202082,
       0.22512083, 0.33760956, 0.5018087 , 0.03635503, 0.23801566,
       0.36177434, 0.3693663 , 0.43153403, 0.26364196, 0.47484293,
       0.06663956, 0.27151643, 0.50414367, 0.55487254, 0.43479958,
       0.37528473, 0.43006502, 0.39151526, 0.3943622 , 0.5362567 ,
       0.55717776, 0.50878421, 0.42617776, 0.50641159, 0.62170114,
       0.55929539, 0.48579454, 0.39864428, 0.61044051, 0.51398993,
       0.37791063, 0.30664315, 0.58154614, 0.48759463, 0.53302467,
       0.31693425, 0.49463828, 0.58531649, 0.59861082, 0.61892471,
       0.23370264, 0.44475373, 0.54060572, 0.57808265, 0.57623567,
       0.55297302, 0.51585343, 0.55579575, 0.27793624, 0.49524145,
       0.56405298, 0.57784591, 0.52274801, 0.6313322 , 0.08521853,
       0.44260057, 0.3237709 , 0.54096805, 0.5820664 , 0.29582948,
       0.58655026, 0.45231288, 0.46830265, 0.36000354, 0.47143411,
       0.35317476, 0.28265172, 0.47102324, 0.43236677, 0.54323936,
       0.10831225, 0.22108374, 0.00448511, 0.03036773, 0.16534823,
       0.20285151, 0.51765323, 0.48590397, 0.46259037, 0.12086046,
       0.47867023, 0.52337341, 0.1289478 , 0.5618949 , 0.50054917,
       0.07194211, 0.63868093, 0.35554133, 0.58968356, 0.43872323,
       0.56951742, 0.44700962, 0.27276241, 0.05068582, 0.57424805,
       0.13242666, 0.4626182 , 0.53707541, 0.37068256, 0.51956487,
       0.37098379, 0.45441607, 0.01914986, 0.56112586, 0.57214236,
       0.09120134, 0.4945929 , 0.31958252, 0.23703607, 0.45224539,
       0.47366877, 0.45916664, 0.5837774 , 0.51266421, 0.51798195,
       0.53490128, 0.49085079, 0.12532637, 0.54900399, 0.55651719,
       0.52279708, 0.46100705, 0.4770828 , 0.2945584 , 0.36502559,
       0.20876186, 0.51160405, 0.49320708, 0.36218263, 0.00606357,
       0.48178891, 0.50754259, 0.56282114, 0.46576776, 0.50057087,
       0.29104633, 0.34205027, 0.54985868, 0.11961635, 0.15510656,
       0.4370881 , 0.01156361, 0.58866461, 0.4920353 , 0.50917157,
       0.54981126, 0.16832273, 0.49074961, 0.40644768, 0.56222673,
       0.53028456, 0.08211016, 0.48792563, 0.28247582, 0.31468493,
       0.29983623, 0.55421689, 0.53187089, 0.48192209, 0.54055018,
       0.55091698, 0.45775599, 0.04751093, 0.08299646, 0.44235462,
       0.48260857, 0.07809599, 0.27491244, 0.40433159, 0.24770267,
       0.33999491, 0.04992993, 0.40361423, 0.36916642, 0.45685928,
       0.00276854, 0.36816915, 0.49743358, 0.54713282, 0.48730846,
       0.26508389, 0.59700311, 0.39850516, 0.61330409, 0.47290575,
       0.52337193, 0.09672676, 0.51720179, 0.5116529 , 0.0473538 ,
       0.30803559, 0.26742336, 0.5059218 , 0.25717369, 0.04206292])
```

For Silhouette score, we take the average of all observations.

If Silhouette score is positive and is tending to +1, cluster is well separated.

If Silhouette score is around 0, cluster is not well separated.

If Silhouette score is tending to -1, model has done a blunder.

- The silhouette scores and silhouette widths are calculated using silhouette_samples and silhouette_score package from sklearn. metrics. The average silhouettes score is coming to be **0.400** and minimum silhouette score is **0.002.**
- The silhouette score ranges from -1 to +1 and higher the silhouette score better the clustering.
- The min silwidth is computed which is **0.0027.** All positive silwidth means that all details are matched to the correct clusters.

Below is the Data frame where Clusters and Silhouette width are appended to the original dataset:

spending	advance_payments	probability_of_full_payment	current_balance	credit_limit	min_payment_amt	max_spent_in_single_shopping	kmeans_clusters	silh_wid
19.94	16.92	0.875200	6.675	3.763	3.252000	6.550	1	0.573278
15.99	14.89	0.906400	5.363	3.582	3.336000	5.144	2	0.365564
18.95	16.42	0.882900	6.248	3.755	3.368000	6.148	1	0.637092
10.83	12.96	0.810588	5.278	2.641	5.182000	5.185	0	0.515595
17.99	15.86	0.899200	5.890	3.694	2.068000	5.837	1	0.360972
12.70	13.41	0.887400	5.183	3.091	8.079625	5.000	0	0.221525
12.02	13.33	0.850300	5.350	2.810	4.271000	5.308	0	0.475295
13.74	14.05	0.874400	5.482	3.114	2.932000	4.825	2	0.360258
18.17	16.26	0.863700	6.271	3.512	2.853000	6.273	1	0.519383
11.23	12.88	0.851100	5.140	2.795	4.325000	5.003	0	0.534439

Cluster Frequency and Profiling:

Cluster profiling is the main idea behind doing the process we have performed till now. For cluster profiling, we would group the dataset by the clusters column and calculate the mean of each variable. We can also calculate the frequency of each cluster and see how many records are grouped under each cluster.

The python output for the same is as below:

kmeans_clusters	0	1	2
spending	11.856944	18.495373	14.437887
advance_payments	13.247778	16.203433	14.337746
probability_of_full_payment	0.848330	0.884210	0.881597
current_balance	5.231750	6.175687	5.514577
credit_limit	2.849542	3.697537	3.259225
min_payment_amt	4.733892	3.632373	2.707341
max_spent_in_single_shopping	5.101722	6.041701	5.120803
silh_wid	0.399556	0.468077	0.338593
Freq	72.000000	67.000000	71.000000

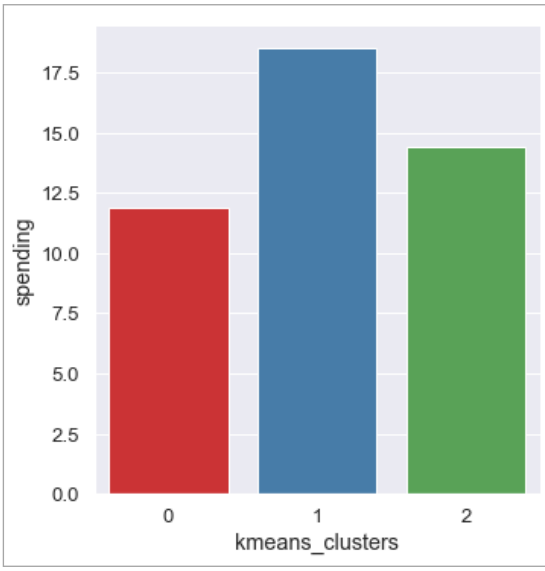
From the above table we can find out that the 3 clusters can be profiled as follows:

Cluster 0 – mean spending is 11.85, with 72 records under it

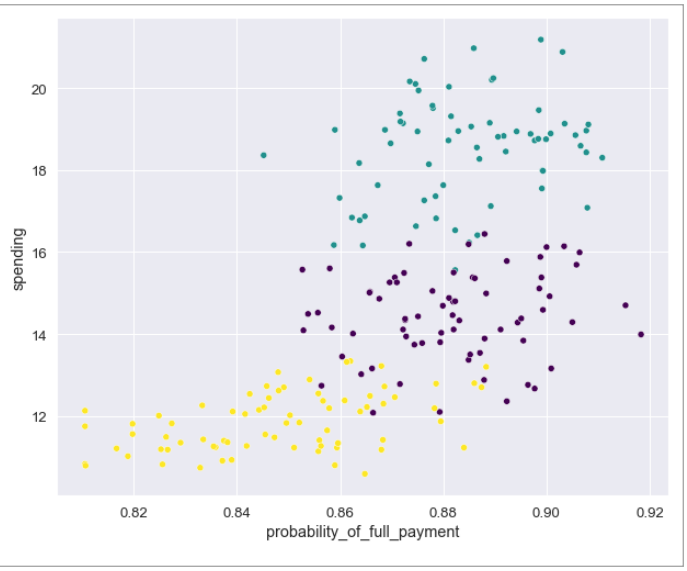
Cluster 1 – mean spending is 18.49, with 67 records under it

Cluster 2 – mean spending is 14.37, with 71 records under it.

Visualizing the spending pattern of each cluster with the help of a bar chart and scatter plot:



(Figure 19: Bar graph of K-means clustering)



(Figure 20: Scatterplot of K-means clustering)

Based on the observations and the bar graph, we can profile these clusters as follows:

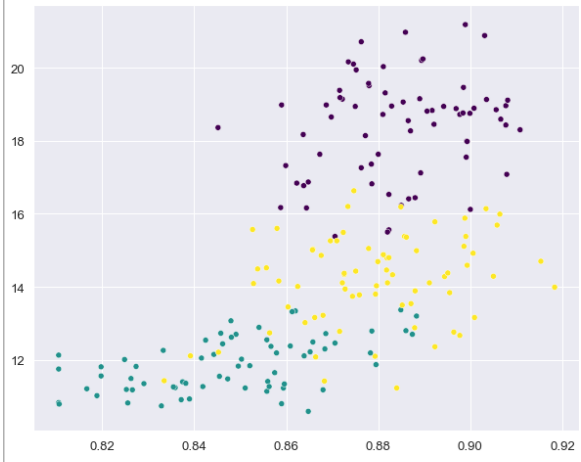
Cluster 0 – Low Spending

Cluster 1 – High Spending

Cluster 2 – Medium Spending

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

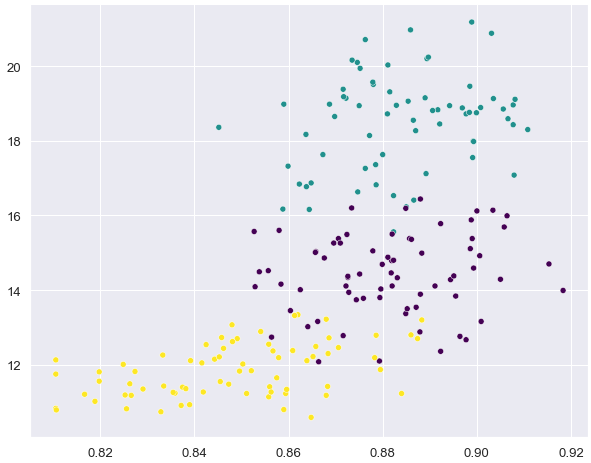
Visualizing the Clusters of Hierarchical clustering *(Figure 21: Scatter plot of Hierarchical clustering)*




```
aggdata=df_clust.groupby('clusters').mean()
aggdata['Freq']=df_clust.clusters.value_counts().sort
aggdata.T
```

clusters	1	2	3
spending	18.371429	11.872388	14.199041
advance_payments	16.145429	13.257015	14.233562
probability_of_full_payment	0.884400	0.848155	0.879190
current_balance	6.158171	5.238940	5.478233
credit_limit	3.684629	2.848537	3.226452
min_payment_amt	3.639157	4.940302	2.612181
max_spent_in_single_shopping	6.017371	5.122209	5.086178
Freq	70.000000	67.000000	73.000000

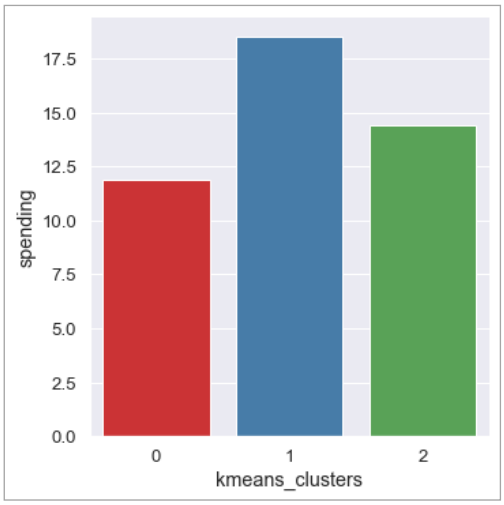
Visualizing the Clusters of K-means clustering *(Figure 22: Scatter plot of K- means clustering)*




```
aggdata=clust_k.groupby('kmeans_clusters').mean()
aggdata['Freq']=clust_k.kmeans_clusters.value_counts()
aggdata.T
```

kmeans_clusters	0	1	2
spending	11.856944	18.495373	14.437887
advance_payments	13.247778	16.203433	14.337746
probability_of_full_payment	0.848330	0.884210	0.881597
current_balance	5.231750	6.175687	5.514577
credit_limit	2.849542	3.697537	3.259225
min_payment_amt	4.733892	3.632373	2.707341
max_spent_in_single_shopping	5.101722	6.041701	5.120803
silh_wid	0.399556	0.468077	0.338593
Freq	72.000000	67.000000	71.000000

We can see that in both the graphs the 3 different clusters are represented with different colours



When we compare the frequencies both the methods, we can clear see that there is no much difference. Hence, will focus on the observations for K-means clusters:

Cluster 0 - Low spending - 72 frequency

Cluster 1 - High spending - 67 frequency

Cluster 2 - Medium spending - 71 frequency

For this question, we will consider cluster profiling which is obtained from K means technique and describing the cluster profiles.

Cluster 0 – Low Spending

This cluster people have the lowest spending with a mean of 11.85. Mean Probability of full payment is 0.84 which is also good. It has the lowest mean current balance out of the 3 clusters which is 5.23. Mean Credit limit is also the lowest with 2.84. The mean maximum spent in single shopping is approximately 5.10.

Cluster 1 – High Spending

This cluster people have a high spending with a mean of 18.49. Mean Probability of full payment is 0.8842 which is pretty good. It also has the highest mean current balance out of the 3 clusters which is 6.17. Mean Credit limit is also the highest with 3.69. This cluster also has the highest mean maximum spent in single shopping which is approximately 6.

Cluster 2 – Medium Spending

This cluster people have a medium spending with a mean of 14.43. Mean Probability of full payment is 0.88 which is pretty good. It has the mid mean current balance out of the 3 clusters which is 5.51. Mean Credit limit is also in the middle with 3.25. The mean maximum spent in single shopping is approximately 5.12.

Recommendations for different promotional strategies for different clusters based on K-means Clustering:

Cluster 0 - Low spending

- This segment has the lowest spending per month, lowest current balance and credit limit. This is the Financially Stressed Class with very low income on an average.
- This segment can be targeted with cards with offers such as zero annual charges and lurking them with benefits such as free coupons or tickets and waivers on a variety of places.
- To boost their payment rate, early payment offers/Rewards can be proposed.
- Minimum payment amount could be lowered and credit limit can be increased. This should be started with those customers whose payment rates are good and for others, offers should be given on early payments so that their payment rate becomes good.
- Increase their spending habits by forming partnerships with grocery stores, utilities, and other businesses (electricity, phone, gas, others)

Cluster 1 - High Spending

- This segment has higher spending per month, high current balance and credit limit. This is the Prosperous or Upper class with majorly higher income.
- They have high expenditure value in single shopping, so can be offered discounts on their next big transaction. This will also increase the spending of this group.
- Giving any reward points might increase their purchases.
- This group's maximum max spent in single shopping is large, thus they may be offered a discount/offer on subsequent purchases if they pay in full.
- Tie up with luxury brands, which will drive more one_time_maximun spending.
- This segment can be targeted using various offers such as cards with rewards and loyalty points for every spent.

Cluster 2 - Medium Spending

- This segment comprises of Customers making decent purchases, pay their bills on time, and have a solid credit score.
- They are potential target customers who are paying bills and doing purchases and maintaining comparatively good credit score. So, we can increase credit limit or can lower down interest rate.
- For these customers, upselling needs to be done. Give them some lucrative discounts so that they use premium accounts to increase transactions
- This segment can be targeted with cards that have lower interest rates so as to encourage more spending.
- Increase spending habits by trying with premium ecommerce sites

---

Problem 2: CART-RF-ANN

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

Attribute Information:
1. Target: Claim Status (Claimed)
2. Code of tour firm (Agency_Code)
3. Type of tour insurance firms (Type)
4. Distribution channel of tour insurance agencies (Channel)
5. Name of the tour insurance products (Product)
6. Duration of the tour (Duration in days)
7. Destination of the tour (Destination)
8. Amount worth of sales per customer in procuring tour insurance policies in rupees (in 100's)
9. The commission received for tour insurance firm (Commission is in percentage of sales)
10. Age of insured (Age)

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

Exploratory Data Analysis:

Read and view data

	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
5	45	JZI	Airlines	Yes	15.75	Online	8	45.00	Bronze Plan	ASIA
6	61	CWT	Travel Agency	No	35.64	Online	30	59.40	Customised Plan	Americas
7	36	EPX	Travel Agency	No	0.00	Online	16	80.00	Cancellation Plan	ASIA
8	36	EPX	Travel Agency	No	0.00	Online	19	14.00	Cancellation Plan	ASIA
9	36	EPX	Travel Agency	No	0.00	Online	42	43.00	Cancellation Plan	ASIA

Checking for the information of features:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Age           3000 non-null   int64
 1   Agency_Code   3000 non-null   object
 2   Type          3000 non-null   object
 3   Claimed       3000 non-null   object
 4   Commision     3000 non-null   float64
 5   Channel       3000 non-null   object
 6   Duration      3000 non-null   int64
 7   Sales         3000 non-null   float64
 8   Product Name  3000 non-null   object
 9   Destination   3000 non-null   object
```

Checking the Skewness and Kurtosis:

```
df.skew()
```

```
Age           1.149713
Commision      3.148858
Duration      13.784681
Sales          2.381148
dtype: float64
```

```
df.kurt()
```

```
Age            1.652124
Commision      13.984825
Duration      427.587926
Sales           6.155248
dtype: float64
```

Checking the description of dataset:

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Age	3000.0	NaN	NaN	NaN	38.091	10.463518	8.0	32.0	36.0	42.0	84.0
Agency_Code	3000	4	EPX	1365	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Type	3000	2	Travel Agency	1837	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Claimed	3000	2	No	2076	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Commision	3000.0	NaN	NaN	NaN	14.529203	25.481455	0.0	0.0	4.63	17.235	210.21
Channel	3000	2	Online	2954	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Duration	3000.0	NaN	NaN	NaN	70.001333	134.053313	-1.0	11.0	26.5	63.0	4580.0
Sales	3000.0	NaN	NaN	NaN	60.249913	70.733954	0.0	20.0	33.0	69.0	539.0
Product Name	3000	5	Customised Plan	1136	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Destination	3000	3	ASIA	2465	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Checking for duplicates in this dataset:

```
Number of duplicate rows = 139
```

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
63	30	C2B	Airlines	Yes	15.0	Online	27	60.0	Bronze Plan	ASIA
329	36	EPX	Travel Agency	No	0.0	Online	5	20.0	Customised Plan	ASIA
407	36	EPX	Travel Agency	No	0.0	Online	11	19.0	Cancellation Plan	ASIA
411	35	EPX	Travel Agency	No	0.0	Online	2	20.0	Customised Plan	ASIA
422	36	EPX	Travel Agency	No	0.0	Online	5	20.0	Customised Plan	ASIA
...	...	...	...	...	...	...	...	...	...	...
2940	36	EPX	Travel Agency	No	0.0	Online	8	10.0	Cancellation Plan	ASIA
2947	36	EPX	Travel Agency	No	0.0	Online	10	28.0	Customised Plan	ASIA
2952	36	EPX	Travel Agency	No	0.0	Online	2	10.0	Cancellation Plan	ASIA
2962	36	EPX	Travel Agency	No	0.0	Online	4	20.0	Customised Plan	ASIA
2984	36	EPX	Travel Agency	No	0.0	Online	1	20.0	Customised Plan	ASIA

139 rows × 10 columns

Observations

- Dataset has 10 columns and 3000 rows
- The dataset contains 10 variables where Age, Commission, Duration, Sales are numeric variable and rest are categorial variables.
- There are no null values in the dataset.
- There are 9 independent variable and one target/dependent variable - Claimed
- The minimum value of 'Duration' is negative value, which is possibly not right, It’s a wrong entry.
- The median and means of Commission & Sales varies significantly.
- Skewness and Kurtosis is also calculated for each column, Data with high skewness indicates lack of symmetry and high value of kurtosis indicates heavily tailed data.
- Based on summary descriptive, the data looks good at first glance.

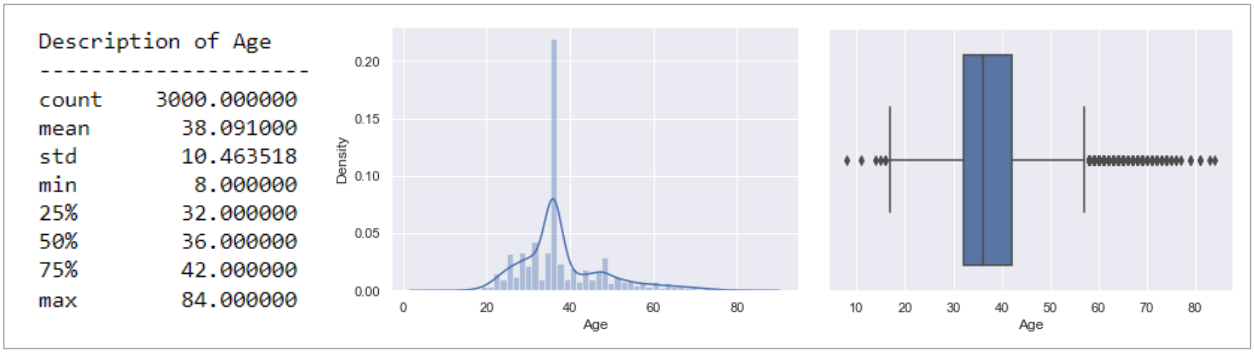
- There are 139 duplicate records, but it can be of different customers, since there is no customer ID or any unique identifier, so not dropping them.

Data Visualization:

Univariate Analysis of Numeric variables:
Let us define a function 'univariateAnalysis_numeric' to display information as part of univariate analysis of numeric variables. The function will accept column name and number of bins as arguments.

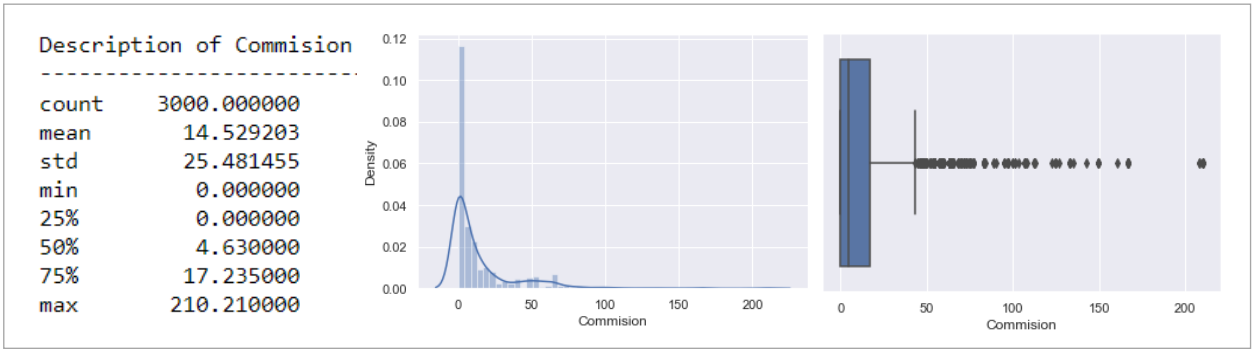
The function will display the statistical description of the numeric variable, histogram or distplot to view the distribution and the box plot to view 5-point summary and outliers if any.

Age of insured (Age) *(Figure 23: Boxplot & Displot of Age)*



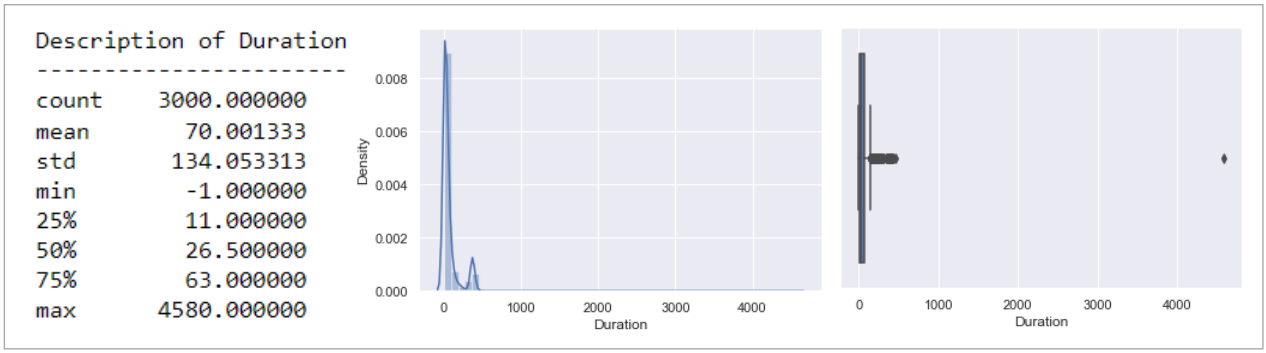
- From the above graphs, we can infer that mean Age of insured (Age) is 38years.
- The distribution of 'Age' is right skewed with skewness value of 1.149.
- The dist plot shows the distribution of data from 20-60 years.
- The box plot of 'Age' show lot of outliers.

The commission received for tour insurance firm (Commission is in percentage of sales)
(Figure 24: Boxplot & Commission)



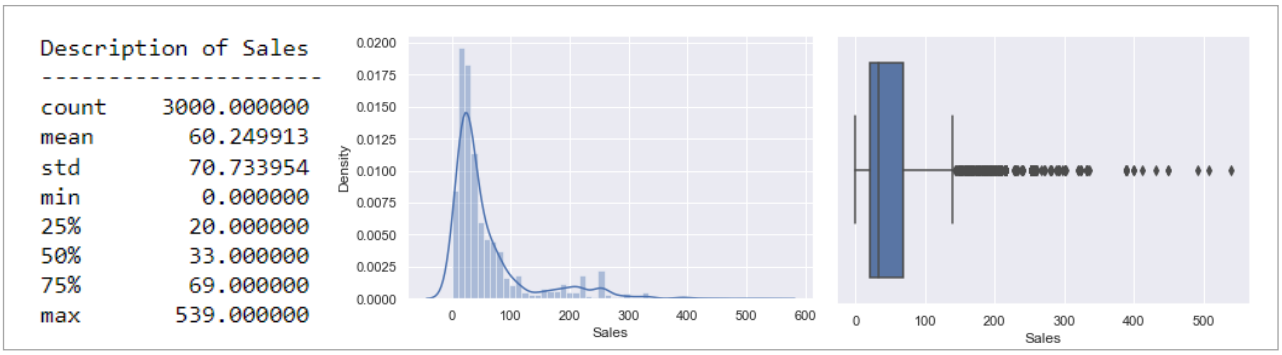
- From the above graphs, we can infer that mean the Commission received for tour insurance firm is 14.52%.
- The distribution of 'Commission' is right skewed with skewness value of 3.148.
- The dist plot shows the distribution of data from 0-50.
- The box plot of 'Commission' shows upper outliers.

Duration of the tour (Duration in days) *(Figure 25: Boxplot & Displot of Duration)*



- From the above graphs, we can infer that mean the Commission received for tour insurance firm is 14.52%.
- The distribution of 'Duration' is right skewed with skewness value of 13.78.
- The dist plot shows the distribution of data from 0-50.
- The box plot of ‘Commission’ shows upper outliers.

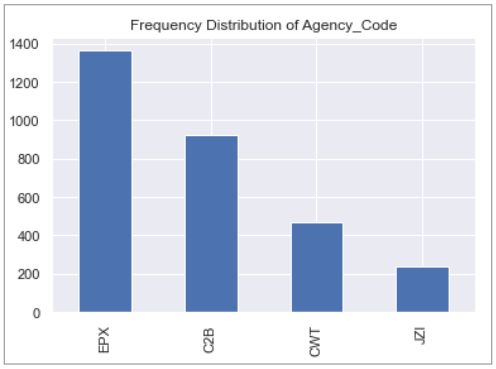
Sales *(Figure 26: Boxplot & Displot of Sales)*



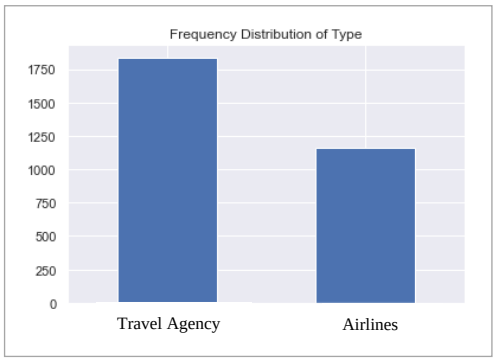
- From the above graphs, we can infer that mean of Sales is 60 with maximum of 539 (in 100’s)
- The distribution of 'Duration' is right skewed with skewness value of 2.38.
- The dist plot shows most of the distribution of data from 0 - 300.
- The box plot of ‘Sales’ shows upper outliers.

Univariate Analysis of Categorical variables: *(Figure 27: Count plot of categorical variables)*

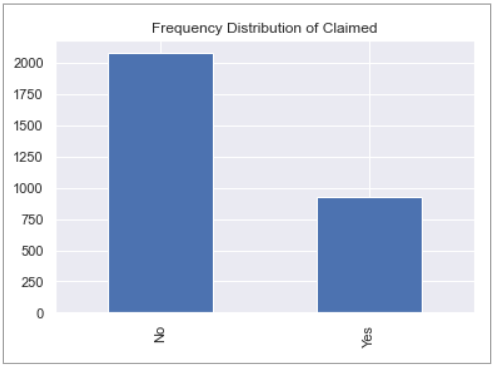
Code of tour firm (Agency_Code)



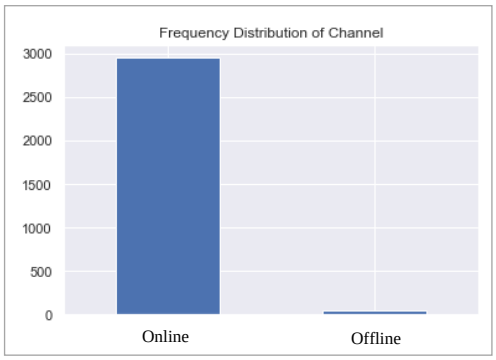
Type of tour insurance firms (Type)



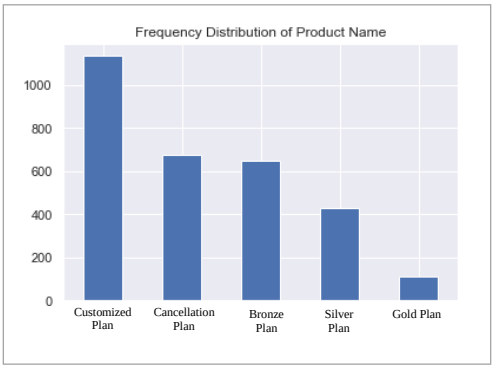
Claim Status (Claimed)



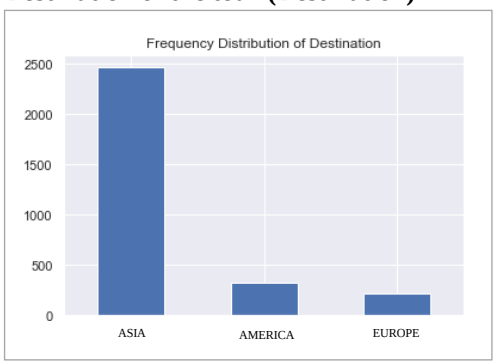
Channel of tour insurance agencies (Channel)



Product Name



Destination of the tour (Destination)

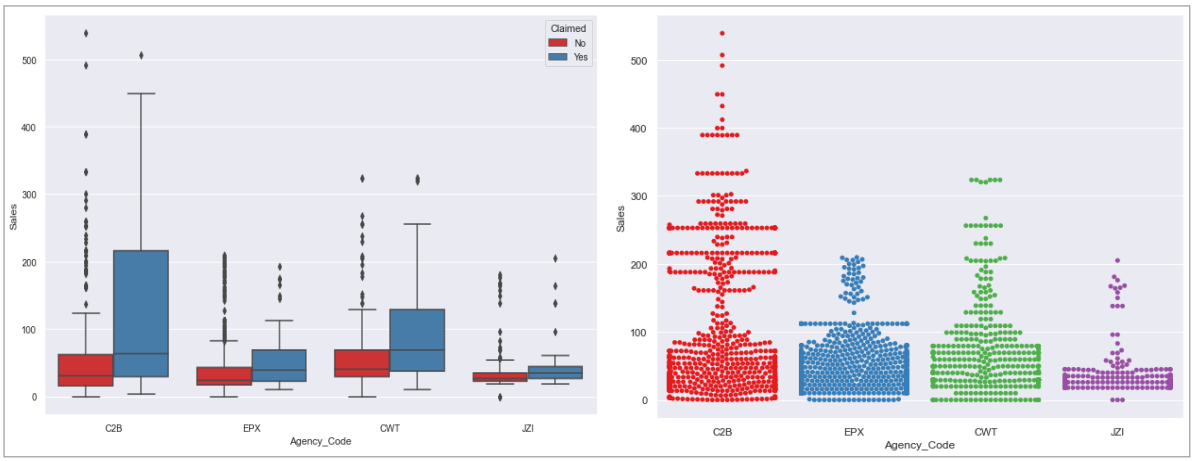


Observations

- The distribution of the agency code, shows us EPX with maximum frequency of 1365.
- The distribution of the Type of tour insurance firms, shows us Travel Agency with maximum claims of 1837.
- The distribution of the Claimed status, shows us 'No' as maximum frequency of 2076.
- The majority of customers have used online medium, very less with offline medium.
- Customized plan seems to be most liked plan by customers when compared to all other plans
- Asia is where customers choose when compared with other destination places

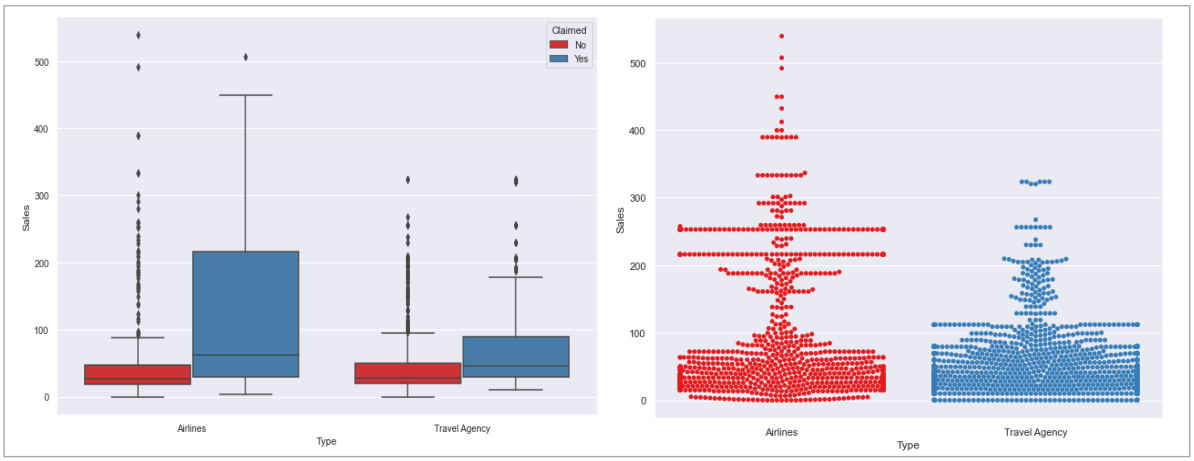
Bivariate Analysis:

Agency_code vs Sales with hue as Claimed ***(****Figure 28: Boxplot and swarm plot of Agency code vs Sales)*



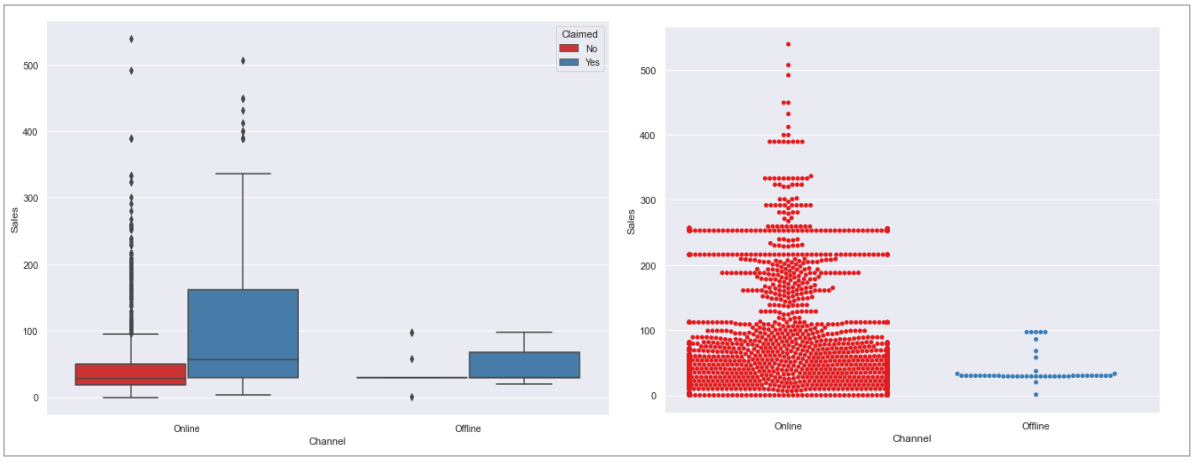
- The box plot shows the split of sales with different agency code and also hue having claimed column.
- It seems that C2B have claimed more claims than other agency
- From the above graph we can infer that, the C2B Agency sales are spread in wide range from 0-400, with few outliers.
- The least sales is from the JZI agency.

Type vs Sales with hue as Claimed *(Figure 29: Boxplot and swarm plot of Type vs Sales)*



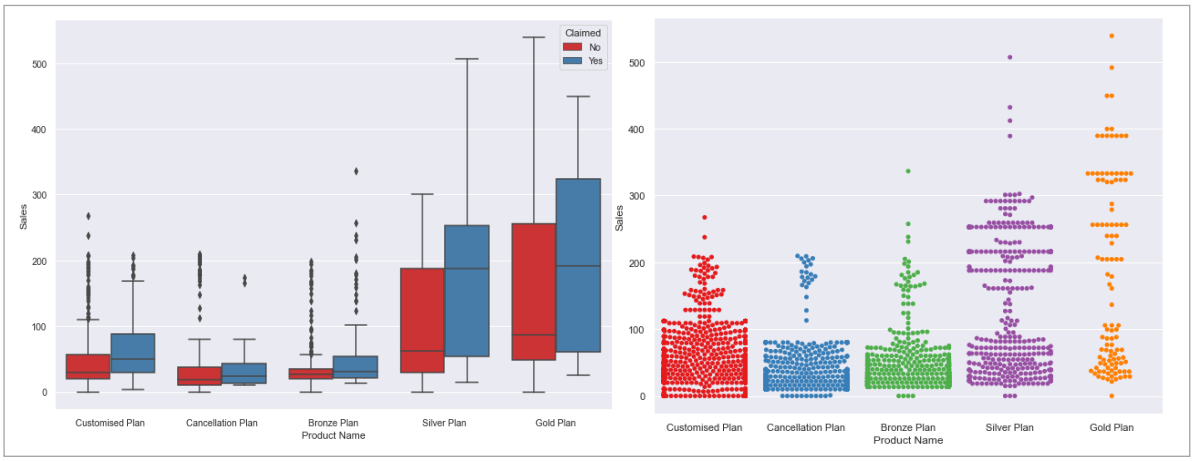
- The box plot shows the split of sales with different type and also hue having claimed column.
- We can see that airlines type has more claims.
- Travel agency type has maximum number of claims though the amount worth of sales per customer in procuring tour insurance policies in rupees (in 100's) ranges from 0-200 of most of data distribution, while the airlines sales is spread till the range of 300.

Channel vs Sales with hue as Claimed *(Figure 30: Boxplot and swarm plot of Channel vs Sales)*



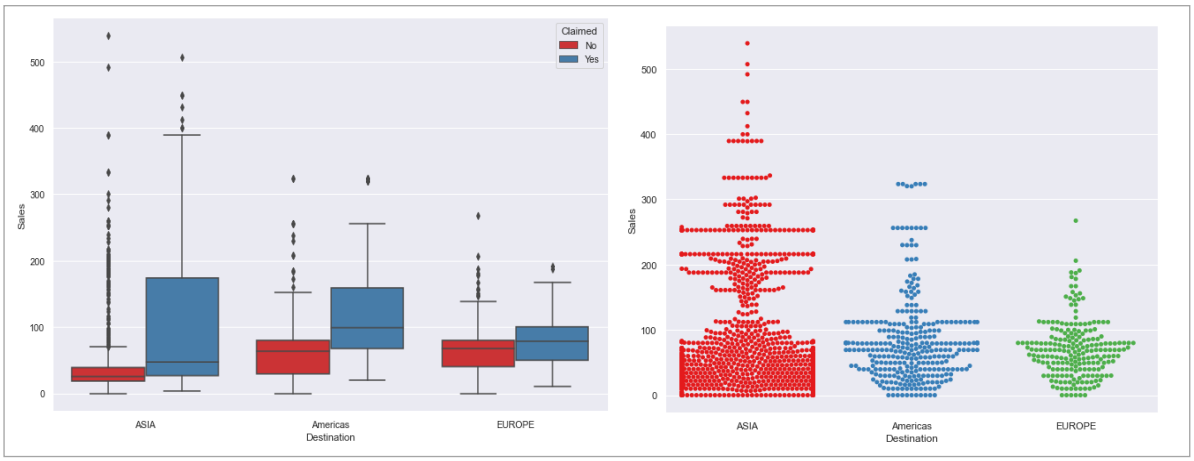
- The box plot shows the split of sales with different channel and also hue having claimed column.
- we can see that; the majority of customers have used online medium and very less with offline medium.

Product Name vs Sales with hue as Claimed *(Figure 31: Boxplot and swarm plot of Product Name vs Sales)*



- The box plot shows the split of sales with different product name and also hue having claimed column.
- We can infer from above plots that, the number of sales in customised plan are more, but the amount worth of sales per customer in procuring tour insurance policies for gold plan is more.

Destination vs Sales with hue as Claimed *(Figure 32: Boxplot and swarm plot of Destination vs Sales)*



- The box plot shows the split of sales with different destination and also hue having claimed column.
- Asia is where customers choose when compared with other destination places.
- Asia has the maximum amount worth of sales per customer for insurance.

Value counts of all the Categorical variables:

```
Details of Agency_Code
-------------------------
EPX     1365
C2B      924
CWT      472
JZI      239
```

```
Details of Destination
-------------------------
ASIA          2465
Americas       320
EUROPE         215
```

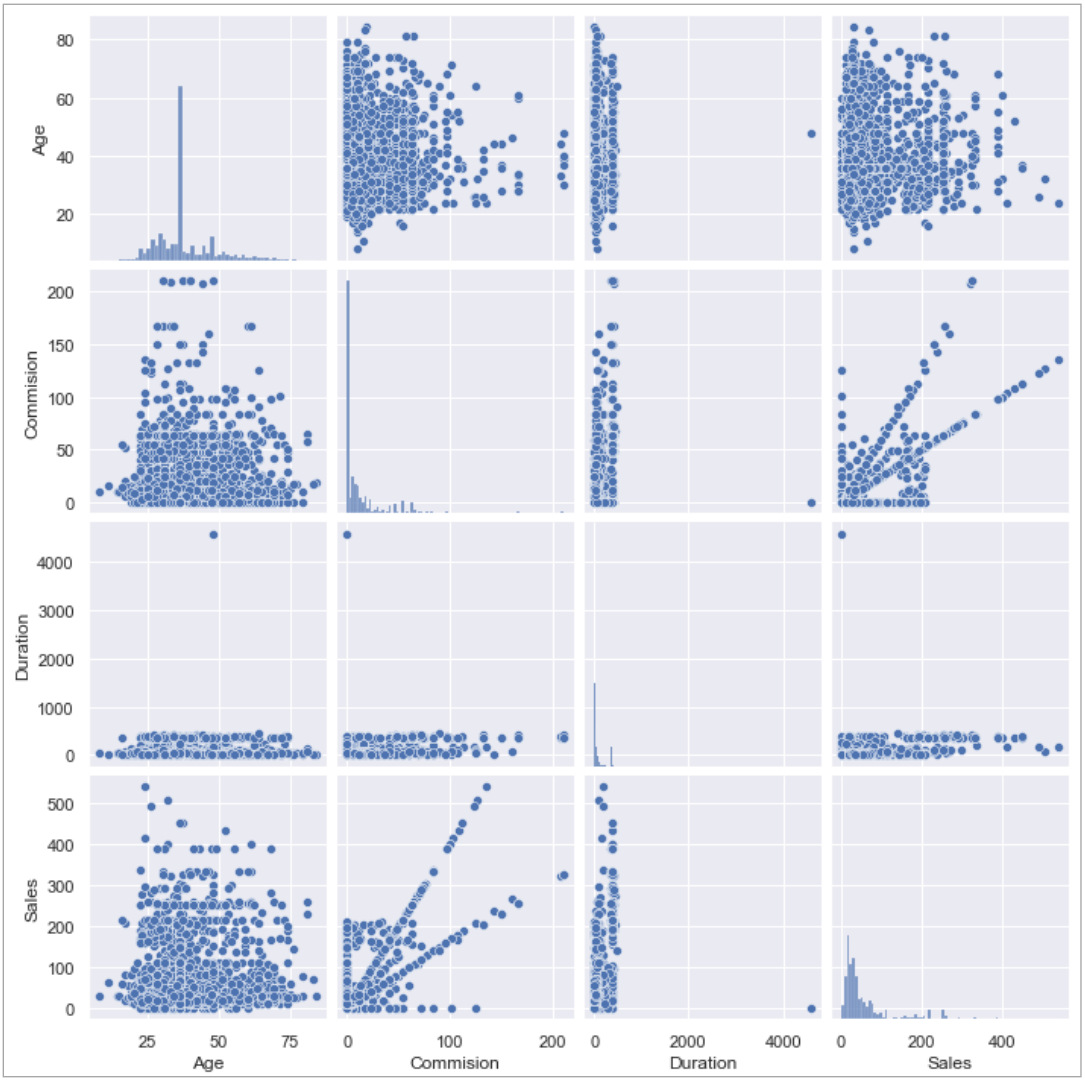
```
Details of Claimed
----------------------
No      2076
Yes      924
```

```
Details of Type
-------------------------
Travel Agency    1837
Airlines         1163
```

```
Details of Channel
-----------------------------
Online      2954
Offline       46
```

```
Details of Product Name
---------------------------
Customised Plan      1136
Cancellation Plan     678
Bronze Plan           650
Silver Plan           427
Gold Plan             109
```

Pair plot *(Figure 33: Pair plot of problem 2)*
A pair plot gives us correlation graphs between all numerical variables in the dataset. Thus, from the graphs we can identify the relationships between all numerical variables.

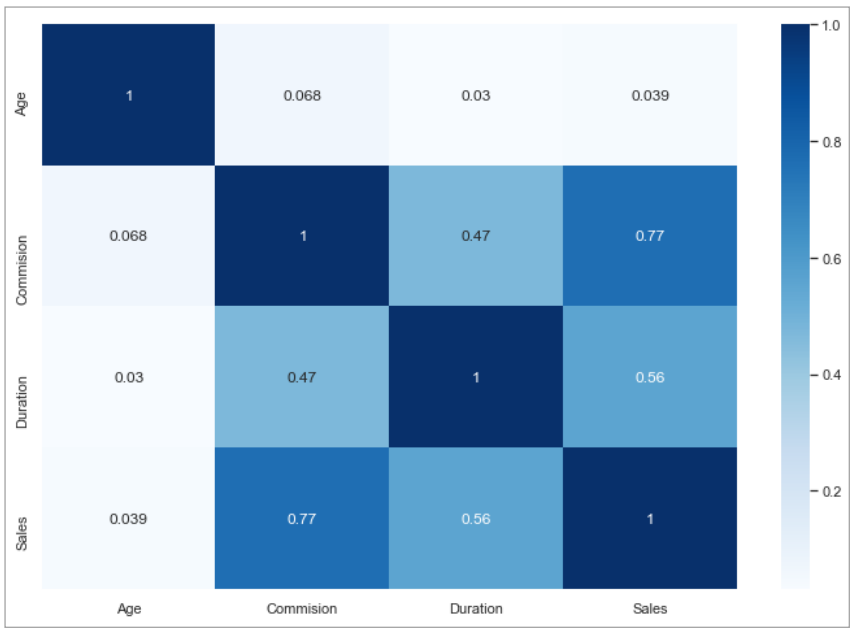


- We can see that the Sales and commission are directly proportional, as the amount of sales per customer for insurance increases the commission received for tour insurance firm also increase.

Multivariate Analysis

Heatmap

A heatmap gives us the correlation between numerical variables. If the correlation value is tending to 1, the variables are highly positively correlated whereas if the correlation value is close to 0, the variables are not correlated. Also, if the value is negative, the correlation is negative. That means, higher the value of one variable, the lower is the value of another variable and vice-versa.



(Figure 34: Heat map of problem 2)

Observations

- We can infer from above Heatmap that there is not much of multi collinearity.
- No negative correlation between the variables.
- we can see only positive correlation between the variables

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

After the extensive analysis of dataset, and understanding data set's underlying structure as well as trends, patterns, and linkages, next we go ahead to splitting the dataset, but before that we have to perform these below steps:

A) Feature Engineering:

Feature engineering for model building involves categorical encoding. It is the technique used to encode categorical features into numerical values which are usually simpler for an algorithm to understand. One hot encoding or Label encoding is a popularly used technique of categorical encoding. Here, categorical values are converted into simple numerical 1's and 0's without the loss of information, and in the case of Label encoding the categorical features are labelled with numeric values according to the alphabetical order.

Following is the output of head() of the dataset after feature engineering:

	Age	Agency_Code	Type	Claimed	Commision	Channel	Duration	Sales	Product Name	Destination
0	48	0	0	0	0.70	1	7	2.51	2	0
1	36	2	1	0	0.00	1	34	20.00	2	0
2	39	1	1	0	5.94	1	3	9.90	2	1
3	36	2	1	0	0.00	1	4	26.00	1	0
4	33	3	0	0	6.30	1	53	18.00	0	0

B) Checking the proportion of Observations of Target variable i.e. Claimed:

Identifying 'Claimed' column as our target variable. We have to train our dataset to identify if the insurance is claimed (Yes) or not (No).

Checking the proportion of Yes and No in Claimed, we get:

Claimed	Value counts	Percentage
Yes	2076	69.2%
No	924	30.8%

Table 2: Proportions of Target variable – Claimed

The proportion is good enough to train the dataset in it.

C) Capturing the Target variable into separate vectors for train and test data:

Train data includes independent variables and the test data includes the dependent variables in the dataset. Pop out the claimed variable from the dataset and keep it in 'y'. The rest of the dataset is kept in 'X'.

```
X = df.drop("Claimed" , axis=1)

y = df.pop("Claimed")
```

D) Scaling the Train data (X)

Scale the dataset X for keeping minimum variance between variables. Would help in better model performance. The train data is scaled by z-score method from Standard scaler.

E) Splitting data in to Train and Test

Now we split the transformed and scaled dataset into train and test. The split is done in a **70-30 ratio** where the train data is 70% and test data is taken to be 30%. The **random state** mentioned here is equal to **1**.

F) Checking the Value counts of Train and Test data

```
X_train (2100, 9)
X_test (900, 9)
train_labels (2100,)
test_labels (900,)
```

- **Train dataset has 2100 records (70% of the total dataset).**
- **Test dataset has 900 records (30% of the total dataset).**

Now we have our train and test data ready. We will start building our classification models one by one.

Models

1) CART Model

CART model is for a supervised learning algorithm which can be used for both classification and regression type of problems. We train the model on the training set and validate it on the testing set. The parent node gets split into child nodes and pruning is done to avoid overgrowing of sub-trees/branches.

Initially, we fit the train data and labels in the **CART model**, based on the model performance the model is tunned using **Grid search,** the best parameters are used and the model is re-built and model performance is calculated which includes Classification report of accuracy, recall, precision and F1 score for both train and test data.

Hyperparameter Tuning:

- *'max_depth': [4,5,6,7]*
- *'min_samples_leaf': [20,30,50]*
- *'min_samples_split': [60,90,150,200]*

- *Cross validation (cv) = 3*

To prune our decision tree to the best height, we take the values of max depth as 5,6 and 7 in our grid search to get optimum results.

Min samples leaf should be 1% to 3% of the total records. 1% of 3000 is 30. We take 20, 30 and 50 in our case to see which fits perfectly. Also, min samples split is approximately 3 times of Min samples life. So we take 60, 90 and 150 as our inputs for grid search. Also, cross validation (CV) given for this is equal to 3.

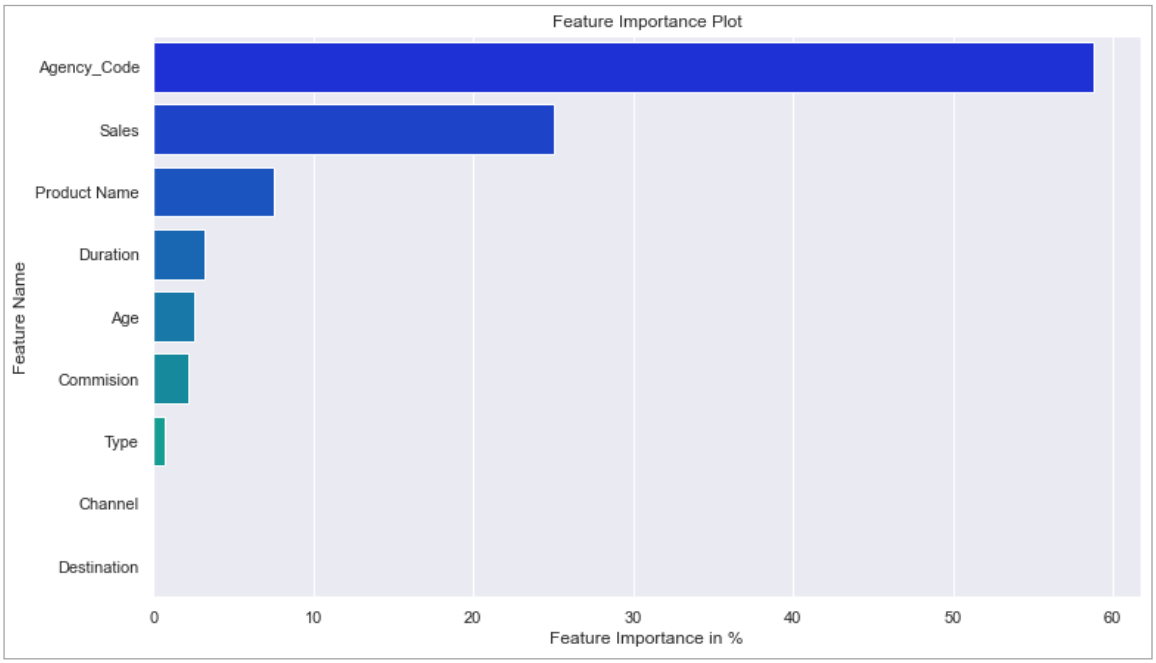
We fit the grid on the train data set. Once the grid is fit, we check the best parameters that are given to us from the grid search algorithm.

The final best parameters are:

- *Max depth is equal to 5*
- *Min samples leaf is 30*
- *Min_samples spit is 150.*

Feature importance:

Feature importance is also checked for the following to understand the importance of variables and how the Decision tree is split into branches.



(Figure 35: Feature importance plot of CART model)

The Agency code variable has the highest importance with approximately 58% followed by Sales with 25%. Destination and Channel have the least importance with almost 0%.

2) Random Forest Model

Random Forest is based on the concept of ensemble learning, which is a process of combining multiple models to solve a complex problem and to improve the performance of the model. It is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the majority to improve the classification accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Initially, we fit the train data and labels in the **Random Forest model**, based on the model performance the model is tunned using **Grid search,** the best parameters are used and the model is re-built and model performance is calculated which includes Classification report of accuracy, recall, precision and F1 score for both train and test data.

Hyperparameter Tuning:

- *'max_depth': [4,5,6,7,8]*
- *'max_features': [3,4,5, 6,7]*
- *'min_samples_leaf': [20,30, 40,50,60]*
- *'min_samples_split': [60,150, 120,150,180]*
- *'n_estimators': [301, 501]*
- *Cross validation (cv) = 3*

To hyper tune the random forest trees to the best height, we take the values of max depth as 4,5,6, 7 and 8 in our grid search to get optimum results. Max depth represents the depth of each tree in the forest.

Max_features in the algorithm is the maximum number of features random forest model is allowed to try in an individual tree. There are many ways to take max_features. SQRT is one of them. Here we have 10 variables and taking 2 as max_features doesn't make much sense. For grid search we keep 3,4,5,6 and 7. Let us see what we get as our best parameter.

Min samples leaf should be 1% to 3% of the total records. 1% of 3000 is 30. We take 20, 30,40,50 and 60 in our case to see which fits perfectly. Also, min samples split is approximately 3 times of Min samples life. So we take 60, 120,150 and 180 as our inputs for grid search. Also, cross validation (CV) given for this is equal to 3.

N_estimators is the number of trees you want to build before taking the maximum voting or averages of predictions. Higher number of trees give you better performance but makes your code slower. We check for 301 and 501 in our grid search. Let's check for the best out of these in our model.

We fit the grid on the train data set. Once the grid is fit, we check the best parameters that are given to us from the grid search algorithm.

The final best parameters are:

- *Max depth is equal to 5*
- *Max_features is equal to 7*
- *Min samples leaf is 20*
- *Min_samples spit is 60*
- *n_estimators is 501*

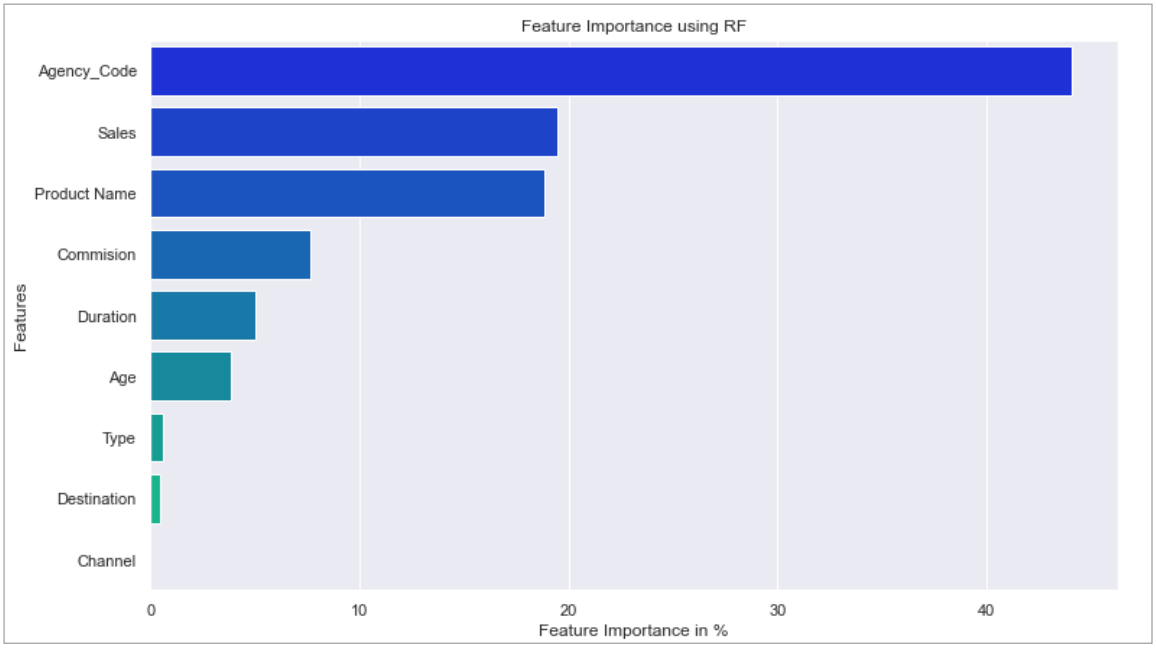
Feature importance:

We can check for the importance of each variable in our model built obtained from the grid search best parameters

Agency code is having the highest importance same as CART model but the difference is that in CART it had approx. 60% importance, in Random Forest the importance for Agency code is approx. 40% only.

Sales and Product name have almost equal importance Sales followed by product name which is around 20% approx.

Destination and Channel are still at the least same as CART but in Random Forest, the importance of Destination has slightly increased as compared to CART.



(Figure 36: Feature importance plot of Random Forest model)

3) Artificial Neural Network

Neural networks are composed of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

We used "Relu" as the activation function in the input and hidden layers, and "adam" for the output layer as this is a multiclass classification problem. We used "categorical accuracy" as the scoring metric.

Initially, we fit the train data and labels in the **Neural Network model**, based on the model performance the model is tunned using **Grid search,** the best parameters are used and the model is re-built and model performance is calculated which includes Classification report of accuracy, recall, precision and F1 score for both train and test data.

Hyperparameter Tuning:

- *hidden_layer_sizes': [50,100,200],*
- *'max_iter': [4000,5000],*
- *'solver': ['sgd','adam'],*
- *'activation': ['logistic', 'relu'],*
- *'tol': [0.01,0.001]*
- *Cross validation (cv) = 5*

A hidden layer is positioned between the algorithm's input and output in neural networks, and it applies weights to the inputs and directs them through an activation function as the output. The number of hidden neurons should be proportional to the size of the input and output layers. In our grid search, we use 50, 100, and 200 to get the optimal parameter for our model.

There is no set criteria for maximum iterations. The solver will run the model till it reaches convergence or till the max iterations you have provided. In this case we have given 4000 and 5000 as inputs. We will see which fits better.

The solver is the process that runs for optimization of the weights in the model. We have given sgd and adam as our inputs. Sgd is Stochastic gradient descent. Adam works very well on large training sets. Where we have training sets of thousands. So more or less adam would be chosen by our best grid. We will get to know that later.

Activation decides the which activation function would be used for the hidden layer. We have given logistic and relu. Logistic is the sigmoid function **[f(x) = 1/(1+exp(-x))]**. Relu is the rectified linear unit function **[f(x) = max(0, x)].**

Tol is the tolerance of optimization. When the training loss is not improved by at least the given tol on consecutive iterations, convergence is considered to be reached and the training stops. We will be checking for tolerance of 0.01 and 0.001.

The final best parameters are:

- *Activation - Relu*
- *hidden_layer_sizes - 200*
- *max_iter - 4000*
- *Solver - adam*
- *Tol – 0.001*

Our new model, which is based on the grid search algorithm's best parameters, is then saved in a distinct variable (best grid ann), and the model's performance is tested using these parameters. For the Artificial Neural Network model, there is no feature importance parameter.

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.

Experiments and Evaluation

Hyperparameter Tuning:
One of the most critical components of a machine learning workflow is hyperparameter tuning. The improper values for the hyperparameters can lead to incorrect findings and a model that performs poorly. Hyperparameters are model parameters that have their values set before the training. The process of obtaining the correct settings for the model's hyperparameters is known as hyperparameter tuning. We used Grid Search for the hyperparameter tuning.

Grid Search:
Grid search divides the hyperparameter domain into distinct grids. Then, using cross-validation, it attempts every possible combination of values in this grid, computing some performance measures. The ideal combination of values for the hyperparameters is the point on the grid that maximizes the average value in cross-validation. Grid search is a comprehensive technique that considers all possible combinations in order to locate the best point in the domain.

Model performance helps to understand how good the model that we have trained using the dataset is so that we have confidence in the performance of the model for future predictions.

We evaluate our models' performance on train and test datasets once they've been constructed. We try to determine if the model is underfitting or overfitting by checking for accuracy, precision, and other factors. We have specific scores and matrices for our model's performance. Following are the methods used to evaluate the model performance:

- **Confusion Matrix**
- **Classification Report**
  - **Accuracy**
  - **Precision**
  - **Recall**
  - **F1 Score**
- **ROC curve**
- **AUC score**

1) Confusion Matrix:

This gives us how many zeros (0s) i.e. (class = No claim) and ones (1s) i.e. (class = Yes claim) were correctly predicted by our model and how many were wrongly predicted.

	Predicted Class		
Actual class		Class = No	Class = Yes
	Class = No	True Negative	False Positive
	Class = yes	False Negative	True Positive

True positive and true negatives are the observations that are correctly predicted and therefore shown in green. We want to minimize false positives and false negatives so they are shown in red color.

2) Accuracy :

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

3) Precision:

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

Precision = TP/(TP + FP)

4) Recall (Sensitivity):

Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

Recall = TP/(TP + FN)

5) F1 Score:

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. That is, a good F1 score means that you have low false positives and low false negatives, so you're correctly identifying real threats and you are not disturbed by false alarms. An F1 score is considered perfect when it's 1 , while the model is a total failure when it's 0

F1 score = 2 x [(Precision x Recall) / (Precision + Recall)]

6) ROC Curve:

ROC curve is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate. False Positive Rate.

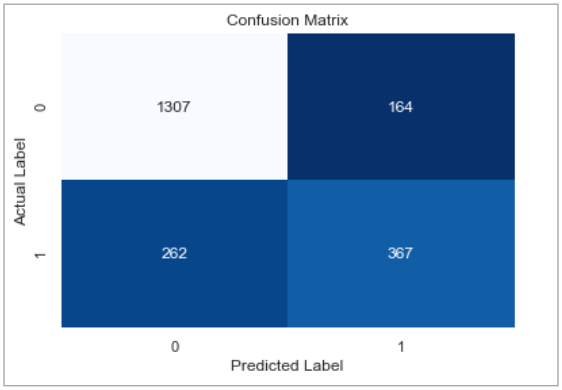
7) AUC Score:

AUC score gives the area under the ROC curve built. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative.

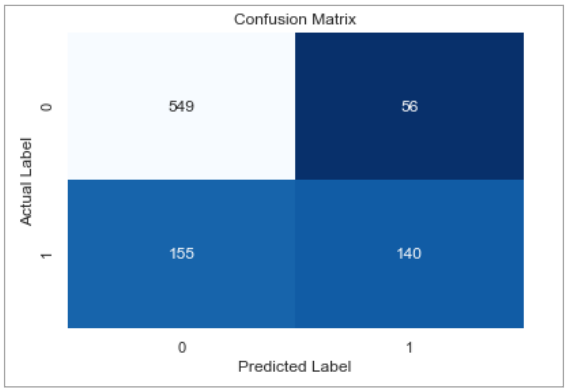
Checking the Model performance of each model one by one:

1. CART Model

a. Confusion Matrix *(Figure 37: Confusion matrix for CART Model)*



Confusion Matrix of Train data



Confusion Matrix of Test data

b. Classification Report

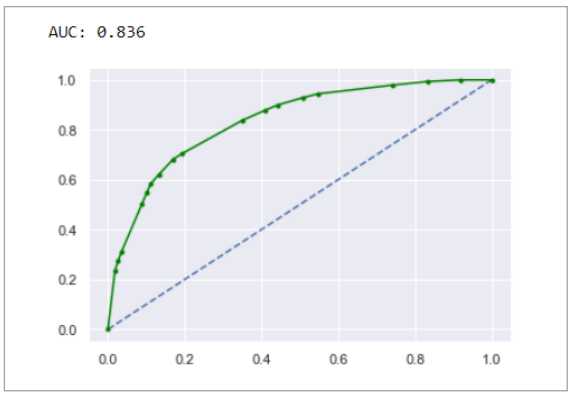
	precision	recall	f1-score	support
0	0.83	0.89	0.86	1471
1	0.69	0.58	0.63	629
accuracy			0.80	2100
macro avg	0.76	0.74	0.75	2100
weighted avg	0.79	0.80	0.79	2100

Classification report of Train data

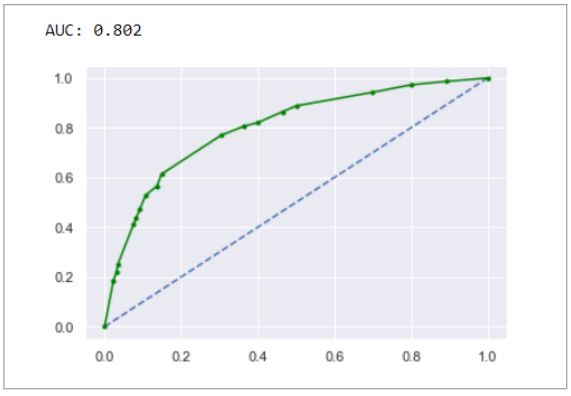
	precision	recall	f1-score	support
0	0.78	0.91	0.84	605
1	0.71	0.47	0.57	295
accuracy			0.77	900
macro avg	0.75	0.69	0.70	900
weighted avg	0.76	0.77	0.75	900

Classification report of Test data

c. ROC Curve and ROC_AUC score *(Figure 38: ROC curve for CART Model)*



ROC Curve and AUC for Train data



ROC Curve and AUC for Test data

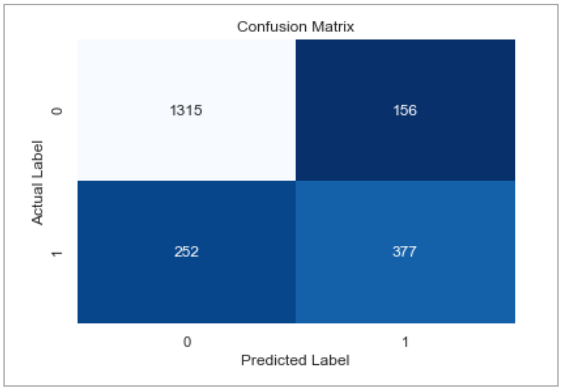
CART Model			
Sl. No		Train Data	Test Data
1.	True Positive	367	140
2.	True Negative	1307	549
3.	False Positive	164	56
4.	False Negative	262	155
5.	Accuracy	80%	77%
6.	Precision	69%	71%
7.	Recall	58%	47%
8.	F1 score	63%	57%
9.	AUC score	83.6%	80.2%

Table 3: Model performance for CART model

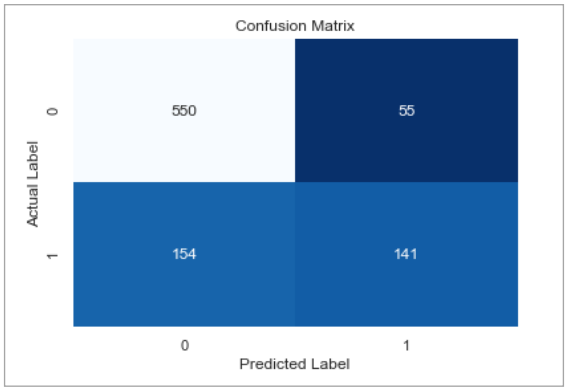
- Test data Accuracy, AUC, precision, and recall are nearly identical to training data.
- This shows that there was neither overfitting or underfitting, and that the model is a good classification model overall.
- Overall, the metrics are high and well-matched.

2. Random Forest Model

a. Confusion Matrix *(Figure 39: Confusion matrix for Random forest Model)*



Confusion Matrix of Train data



Confusion Matrix of Test data

b. Classification Report

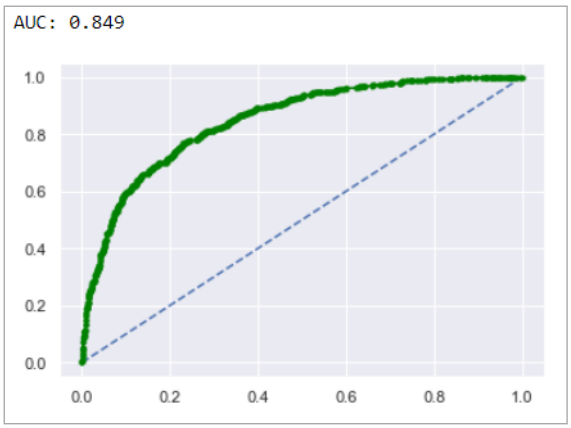
	precision	recall	f1-score	support
0	0.84	0.89	0.87	1471
1	0.71	0.60	0.65	629
accuracy			0.81	2100
macro avg	0.77	0.75	0.76	2100
weighted avg	0.80	0.81	0.80	2100

Classification report of Train data

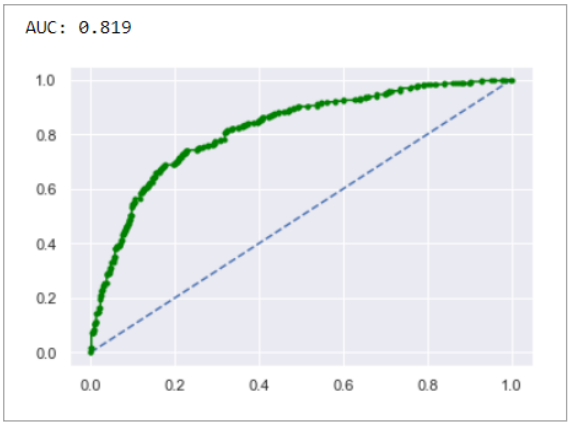
	precision	recall	f1-score	support
0	0.78	0.91	0.84	605
1	0.72	0.48	0.57	295
accuracy			0.77	900
macro avg	0.75	0.69	0.71	900
weighted avg	0.76	0.77	0.75	900

Classification report of Test data

c. ROC Curve and ROC_AUC score *(Figure 40 : ROC curve for Random Forest Model)*



ROC Curve and AUC for Train data



ROC Curve and AUC for Test data

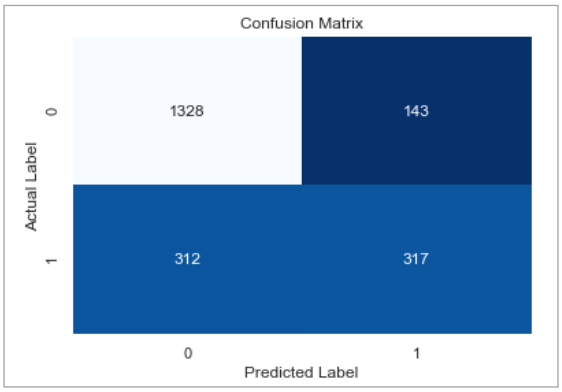
Random Forest Model			
Sl. No		Train Data	Test Data
1.	True Positive	377	141
2.	True Negative	1315	550
3.	False Positive	156	55
4.	False Negative	252	154
5.	Accuracy	81%	77%
6.	Precision	71%	72%
7.	Recall	60%	48%
8.	F1 score	65%	57%
9.	AUC score	85%	82%

Table 4: Model performance for Random Forest model

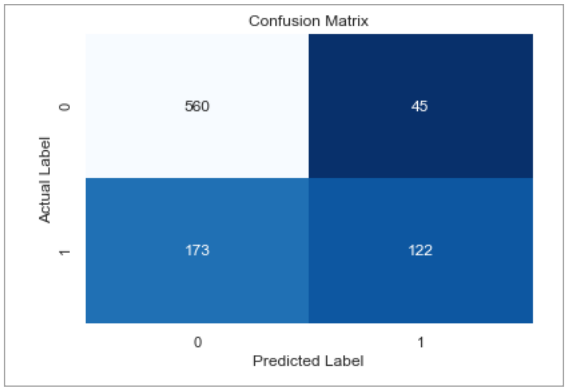
- Test data Accuracy, AUC, precision, and recall are nearly identical to training data.
- This shows that there was neither overfitting or underfitting, and that the model is a good classification model overall.
- Overall, the metrics are high and good fit.

3. Artificial Neural Network Model

a. Confusion Matrix *(Figure 41: Confusion matrix for Artificial Neural Network Model)*



Confusion Matrix of Train data



Confusion Matrix of Test data

b. Classification Report

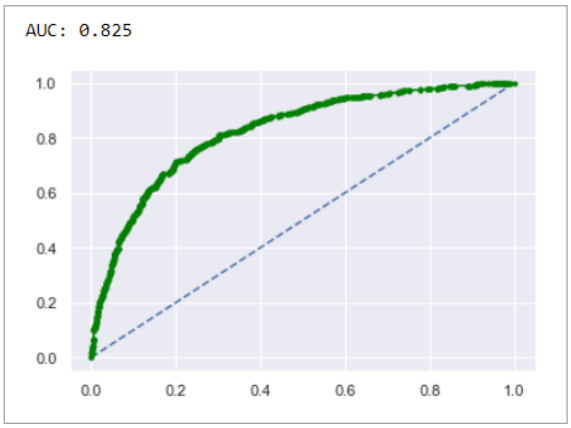
	precision	recall	f1-score	support
0	0.81	0.90	0.85	1471
1	0.69	0.50	0.58	629
accuracy			0.78	2100
macro avg	0.75	0.70	0.72	2100
weighted avg	0.77	0.78	0.77	2100

Classification report of Train data

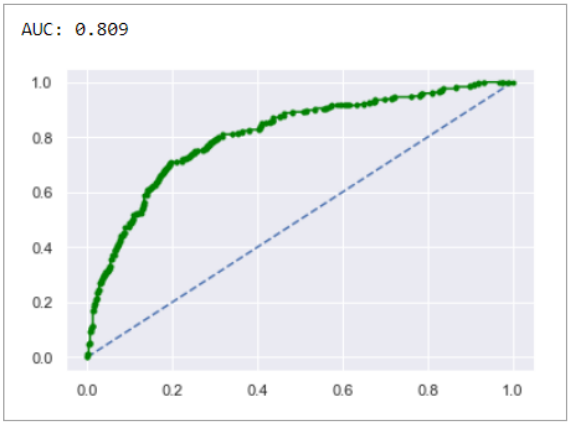
	precision	recall	f1-score	support
0	0.76	0.93	0.84	605
1	0.73	0.41	0.53	295
accuracy			0.76	900
macro avg	0.75	0.67	0.68	900
weighted avg	0.75	0.76	0.74	900

Classification report of Test data

c. ROC Curve and ROC_AUC score *(Figure 42 : ROC curve for Artificial Neural Network Model)*



ROC Curve and AUC for Train data



ROC Curve and AUC for Test data

Artificial Neural Network Model			
Sl. No		Train Data	Test Data
1.	True Positive	317	122
2.	True Negative	1328	560
3.	False Positive	143	45
4.	False Negative	312	173
5.	Accuracy	78%	76%
6.	Precision	69%	73%
7.	Recall	50%	41%
8.	F1 score	58%	53%
9.	AUC score	82.5%	80.9%

Table 5: Model performance for Artificial Neural Network model

- Test data Accuracy, AUC, precision, and recall are nearly identical to training data.
- This shows that there was neither overfitting or underfitting, and that the model is a good classification model overall.
- Overall, the metrics are high and good fit.

Inferences:

We must comprehend the meaning of False Positives and False Negatives as stated in the issue description. False positives are those people who actually did not claim for the insurance but the algorithm predicted that they would claim. False Negatives are those people who actually claimed for the insurance but the model predicted that won't claim.

As a result, we can see that false positives won't have a big impact on the insurance firm, but false negatives will. As a result, Sensitivity or recall will be more important in this instance.

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

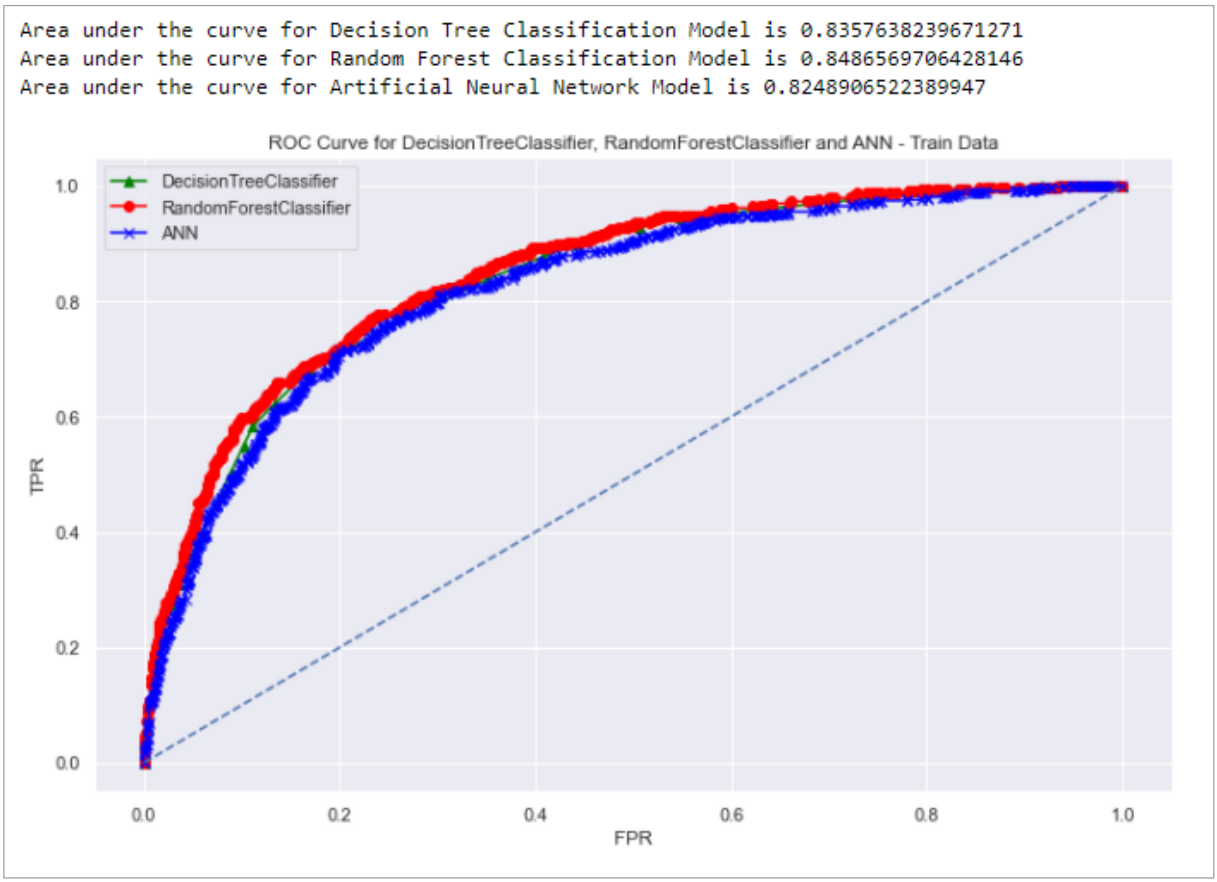
We have created three models so far: CART, Random Forest, and Artificial neural network. On training and test data sets, we evaluated the performance of all three models. Now we'll compare the results of these three models on training and test datasets to see which one is best for making predictions.

Comparison of all models on the basis of the performance metrics:

	Model Performance	CART -Train	CART - Test	RF - Train	RF - Test	ANN - Train	ANN - Test
0	Accuracy	0.805	0.770	0.810	0.77	0.780	0.760
1	AUC	0.836	0.802	0.850	0.82	0.825	0.809
2	Precision	0.690	0.710	0.710	0.72	0.690	0.730
3	Recall	0.580	0.470	0.601	0.48	0.501	0.410
4	F1 score	0.630	0.570	0.650	0.57	0.580	0.530

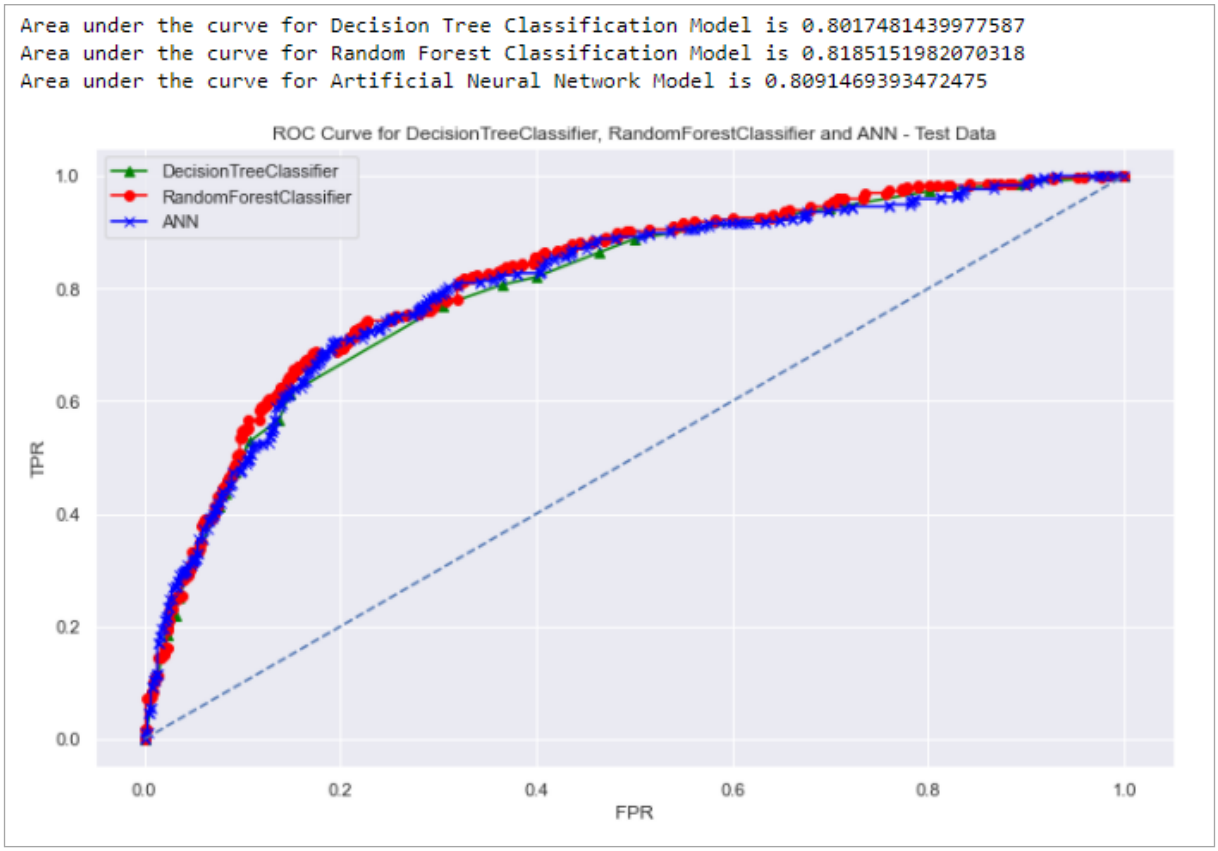
Apart from the table, we can compare the ROC curves of all models on training and test dataset:

Comparison of ROC Curve and AUC scores of all models for Training data:



(Figure 43 : Comparison of ROC curve and AUC score of all models for train data)

Comparison of ROC Curve and AUC scores of all models for Testing data:



(Figure 44 : Comparison of ROC curve and AUC score of all models for test data)

Inferences of Comparison of Model performance:
we have built the 3 models namely CART, random forest and artificial neural network. We have checked the performances of all 3 models on training and test data sets.

- In the problem statement, we should understand the meaning of False positives and False Negatives. False positives are those people who actually did not claim for the insurance but the algorithm predicted that they would claim. False Negatives are those people who actually claimed for the insurance but the model predicted that won't claim.
- As a result, we can see that false positives will not have a significant impact on our firm, however false negatives would. As a result, in this scenario, **sensitivity or recall will be more crucial.**
- All the 3 models are performing good and there is no under fitting or over fitting for and of the models performed.
- The training and test values aren't that far apart for all three values, thus there seems to be no concern of overfitting or underfitting.
- When comparing the ROC curves of the three models, Random Forest has the best graph of the three and also covers the maximum area.
- Clearly, we can infer from the above table that, the **Random Forest model** has the highest values for **Accuracy, AUC score, Precision, Recall and F1 score** when training data is considered.

Finally, the Random Forest model is the most optimised model, and we would choose to build our final model.

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

According to the problem statement, the Insurance firm providing tour insurance is facing higher claim frequency. This means that people are filing insurance claims more frequently. The company wants to go into historical data to figure out why the frequency is so high and come up with a strategy to reduce it by using the model performances.

In our extensive analysis so far, we have thoroughly examined historical data and developed a model that predicts claim status based on the characteristics in our dataset. Let us now look at the key points in our past data first and try to find out some recommendations for the firm.

Following are the insights and recommendations to help the management solve the business objective:

Insights from the Graphs and Analysis from EDA:

- **Claimed:** Although the number of insurances claimed recorded was less than half that of insurances not claimed, the sales for both claimed and unclaimed insurances were almost equal. As a result, sales of claimed insurances were significantly higher than those of non-claimed insurances.

- **Agency:**
  - JZI agency has the lowest number of records as well as the least sales for both claimed and unclaimed insurances.
  - For C2B agency, claimed insurances are more than the unclaimed ones; both in terms of number and sales.
  - EPX agency has the highest unclaimed to claimed ratio. The sales are also good and the claimed insurances are very less as well.

- **Product:**
  - Customised plan and Cancellation plan are the best products when it comes to insurance claims. Their Unclaimed to claimed insurance ratio is very high for these.
  - Silver plan is having claimed insurance bookings way more. The sales of these bookings which have claimed insurances are too high as well.
  - Gold plan is the least taken plan.

- **Booking Type:**
  - The people who booked airlines had equal number for both who claimed and did not claim for insurance. However, the sales for those who claimed were way higher than who did not claim.
  - It can be said that higher sales value bookings opted for insurance claim more. Travel Agency performed good. Unclaimed to claimed ratio was considerably less both in terms of numbers and sales.

- **Channel:**
  - Offline bookings are negligible as compared to Online bookings.
  - Number of Online bookings getting claimed for insurance is lower but sales value is higher.

- **Destination:**
  - Bookings for ASIA is the highest. AMERICA and EUROPE have very low bookings as compared to ASIA. In ASIA, number of bookings that got claimed is lower but the sales of bookings which got claimed was higher.

Recommendations:

- Records having a greater sales value should be prioritised since they are more likely to be claimed for insurance.
- JZI agency needs to equip resources to pick up sales as they fall through the cracks, promotional marketing campaign can be launched, or assess whether the insurance firm need to tie up with another alternate agency.
- The business should promote bookings from Travel Agency
- As a result, insurance firm requires customers to buy airline tickets, as well as cross-sell insurance based on claim data patterns.
- Customers benefited from streamlining online experiences, which resulted in a rise in conversions and, as a result, subsequently raises profits. Online reservations with a higher value should be prioritised.
- The gold plan should be restructured so that more customers choose to use it. The Silver plan should review its procedures and strive to reduce the number of insurance claims.
- ASIA's high sales bookings should be monitored. To attract more clients, several marketing plans for America and Europe tours should be implemented. Perhaps some reductions or special offers can be made.

Key performance indicators of insurance claims:

- Increase customer satisfaction which in fact will give more revenue
- Combat fraud transactions, deploy measures to avoid fraudulent transactions at earliest
- Optimize claims recovery method
- Reduce claim handling costs
