

DM PROJECT REPORT-2022

 DSBA

**Great Learnings:**

**PGP in Data Science and Analytics.**

**PGPDSBA LTC May 2022**

**Authored by: Sugandha Darshi**

|  |  |
| --- | --- |
| **CONTENTS**Problem 11.1 Read the data and do exploratory data analysis (3 pts). Describe the data briefly. Interpret the inferences for each (3 pts). Initial steps like head() .info(), Data Types, etc . Null value check. Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Summary stats, Skewness, Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct. |  |
| 1.2 Do you think scaling is necessary for clustering in this case? Justify The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for example std dev, variance, etc. Should justify whether there is a necessity for scaling and which method is he/she using to do the scaling. Can also comment on how that method works. |  |
| 1.3 Apply hierarchical clustering to scaled data (3 pts). Identify the number of optimum clusters using Dendrogram and briefly describe them (4). Students are expected to apply hierarchical clustering. It can be obtained via Fclusters or Agglomerative Clustering. Report should talk about the used criterion, affinity and linkage. Report must contain a Dendrogram and a logical reason behind choosing the optimum number of clusters and Inferences on the dendrogram. Customer segmentation can be visualized using limited features or whole data but it should be clear, correct and logical. Use appropriate plots to visualize the clusters. |  |
| 1.4 Apply K-Means clustering on scaled data and determine optimum clusters (2 pts). Apply elbow curve (3 pts). Interpret the inferences from the model (2.5 pts). K-means clustering code application with different number of clusters. Calculation of WSS(inertia for each value of k) Elbow Method must be applied and visualized with different values of K. Reasoning behind the selection of the optimal value of K must be explained properly. Report must contain logical and correct explanations for choosing the optimum clusters using the elbow method. Append cluster labels obtained from K-means clustering into the original data frame. Customer Segmentation can be visualized using appropriate graphs. |  |
| 1.5 Describe cluster profiles for the clusters defined (2.5 pts). Recommend different promotional strategies for different clusters in context to the business problem in-hand (2.5 pts ). After adding the final clusters to the original dataframe, do the cluster profiling. Divide the data in the finalyzed groups and check their means. Explain each of the group briefly. There should be at least 3-4 Recommendations. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks will only be allotted if the recommendations are correct and business specific. variable means. Students to explain the profiles and suggest a mechanism to approach each cluster. Any logical explanation is acceptable. |  |
| 2.1 Read the data and do exploratory data analysis (4 pts). Describe the data briefly. Interpret the inferences for each (2 pts). Initial steps like head() .info(), Data Types, etc . Null value check. Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot. Summary stats, Skewness, Outliers proportion should be discussed, and inferences from above used plots should be there. There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct. |  |
| 2.2 Data Split: Split the data into test and train(0.5 pts), build classification model CART (2.5 pts), Random Forest (2.5 pts). Object data should be converted into categorical/numerical data to fit in the models. (pd.categorical().codes(), pd.get\_dummies(drop\_first=True)) Data split, ratio defined for the split, train-test split should be discussed. Any reasonable split is acceptable. Use of random state is mandatory. Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Apply grid search for each model and make models on best\_params. Feature importance for each model. |  |
| 2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy (1 pts), Confusion Matrix (2 pts), Plot ROC curve and get ROC\_AUC score for each model (2 pts), Make classification reports for each model. Write inferences on each model (2 pts). Calculate Train and Test Accuracies for each model. Comment on the validness of models (overfitting or underfitting) Build confusion matrix for each model. Comment on the positive class in hand. Must clearly show obs/pred in row/col Plot roc\_curve for each model. Calculate roc\_auc\_score for each model. Comment on the above calculated scores and plots. Build classification reports for each model. Comment on f1 score, precision and recall, which one is important here. |  |
| 2.4 Final Model - Compare all models on the basis of the performance metrics in a structured tabular manner (2.5 pts). Describe on which model is best/optimized (1.5 pts ). A table containing all the values of accuracies, precision, recall, auc\_roc\_score, f1 score. Comparison between the different models(final) on the basis of above table values. After comparison which model suits the best for the problem in hand on the basis of different measures. Comment on the final model. |  |
| 2.5 Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective. There should be at least 3-4 Recommendations and insights in total. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks should only be allotted if the recommendations are correct and business specific. |  |

**Problem 1**

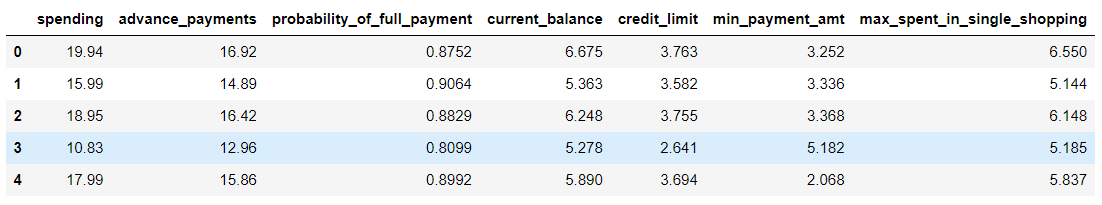
A leading bank wants to develop a customer segmentation to give promotional offers to its customers. They collected a sample that summarizes the activities of users during the past few months. You are given the task to identify the segments based on credit card usage.

1.1

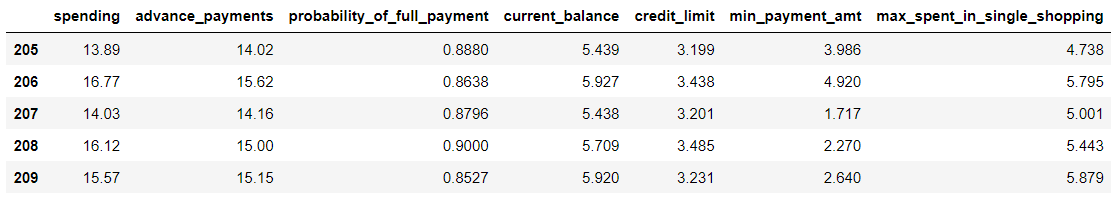
* Read the data and do exploratory data analysis (3 pts).
* Describe the data briefly. Interpret the inferences for each (3 pts).
* Initial steps like head() .info(), Data Types, etc .
* Null value check.
* Distribution plots(histogram) or similar plots for the continuous columns.
* Box plots, Correlation plots. Appropriate plots for categorical variables. Inferences on each plot.
* Summary stats, Skewness, Outliers proportion should be discussed, and inferences from above used plots should be there.
* There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.

**Soln 1.1**

We start with looking at the first 5 and last 5 rows of the data as seen in Figure 1 and 2.

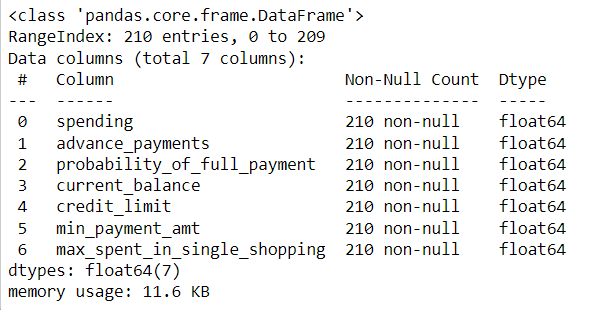


*Figure 1 First 5 rows of data of Problem 1*



*Figure 2 Last 5 rows of data of problem 1*

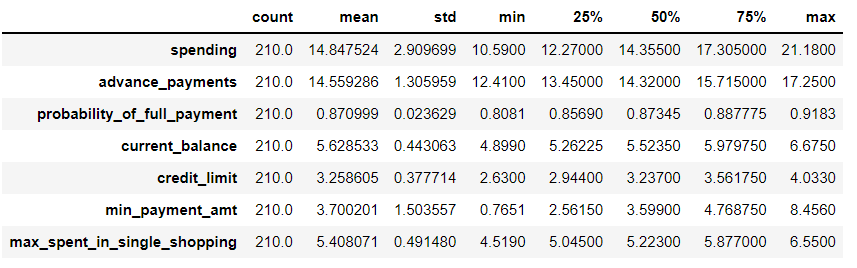
Then we look at the info and check for null values as shown in figure 3.



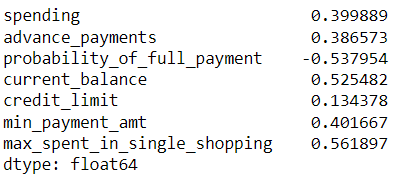
*Figure 3 Info of all columns of the data*

As can be seen from figure 3, there are 210 rows in the data and 7 columns. There are no null values in the data. It can be inferred that all the columns are numerical columns with float values. It can be inferred that all the columns contain numerical continuous entities.

The description of data is shown in figure 4 below.



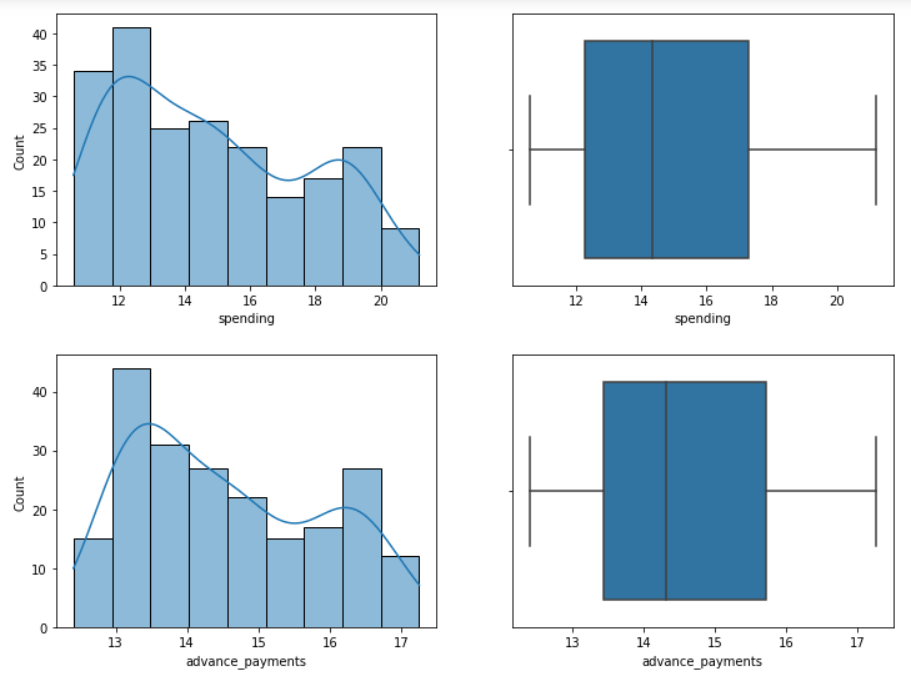
*Figure 4 Description of data*

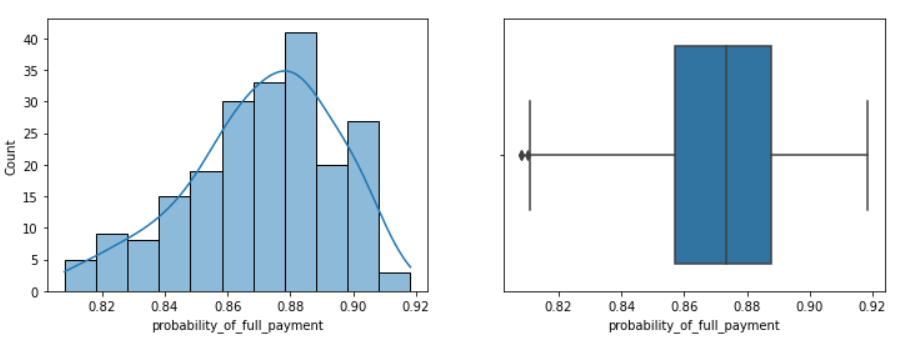


*Figure 5 Skewness of all the variables of data*

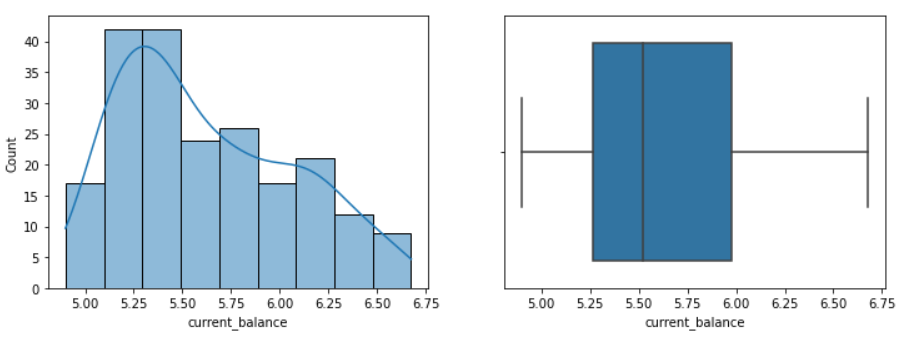
* All variables except probability of full payment are right, positively skewed.
* Probability of full payment is negatively skewed.
* However, the skew values are all less than 0.6 for all the variables which signifies low intensity of skewness.
* **All the variables have different scale with higher scales for spending and advance payments.**
* For the column 'spending', mean is approximately equal to median.
* For the column 'advance payments', mean is approximately equal to median. Low skewness is observed.

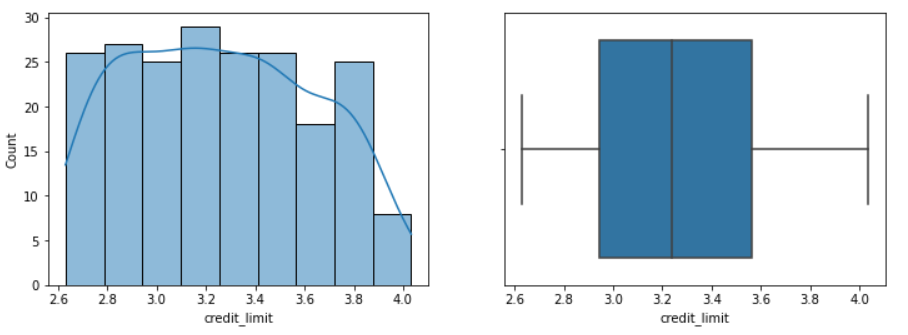
**Univariate Analysis**

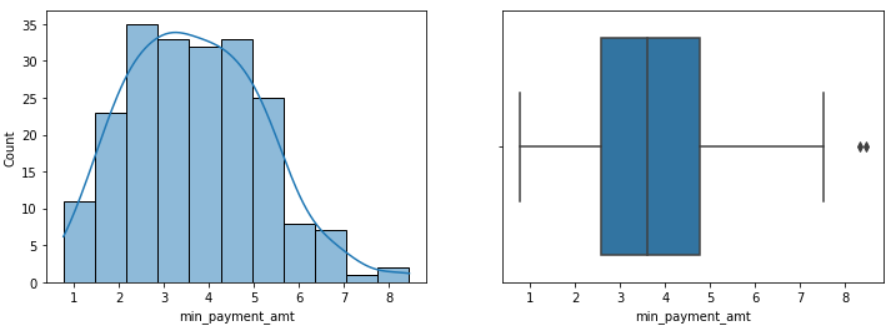


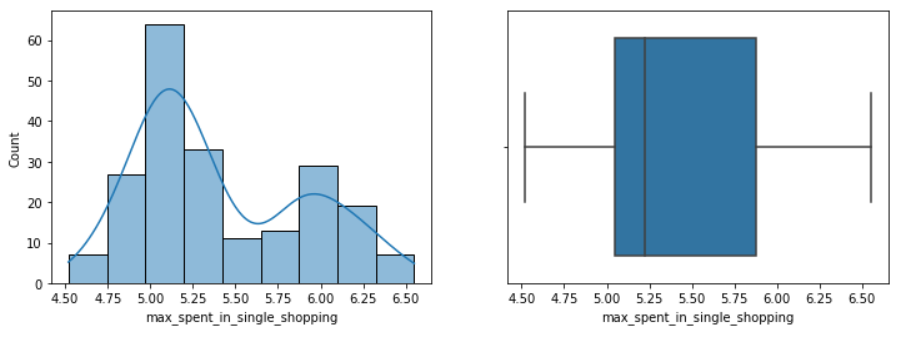


*Figure 6 histogram and box plot of variables*





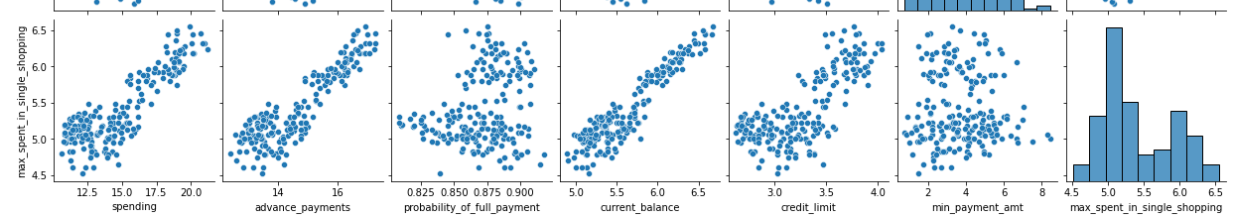
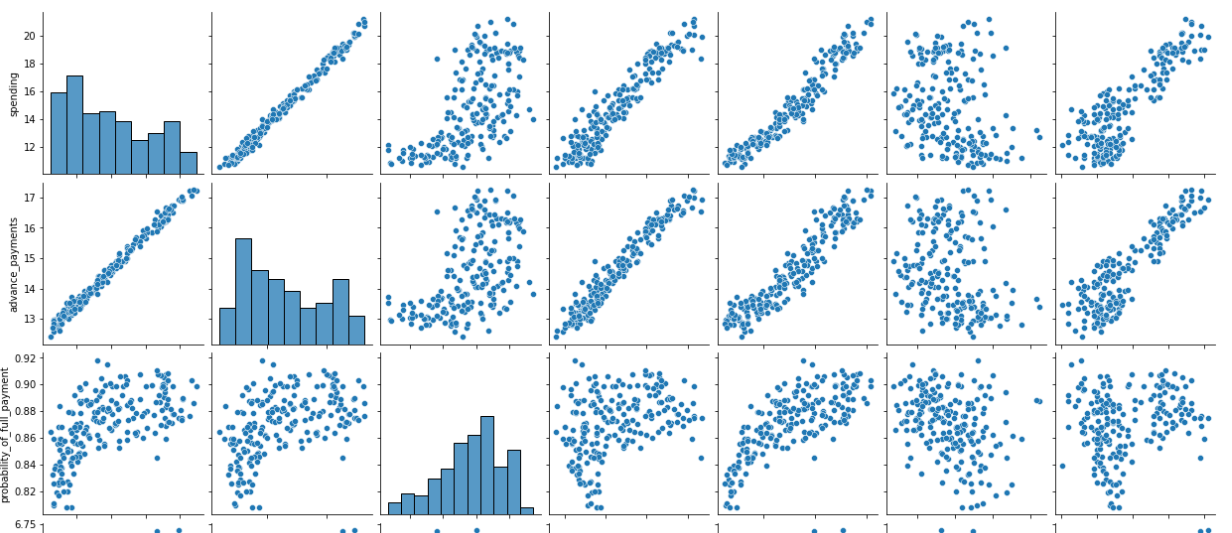
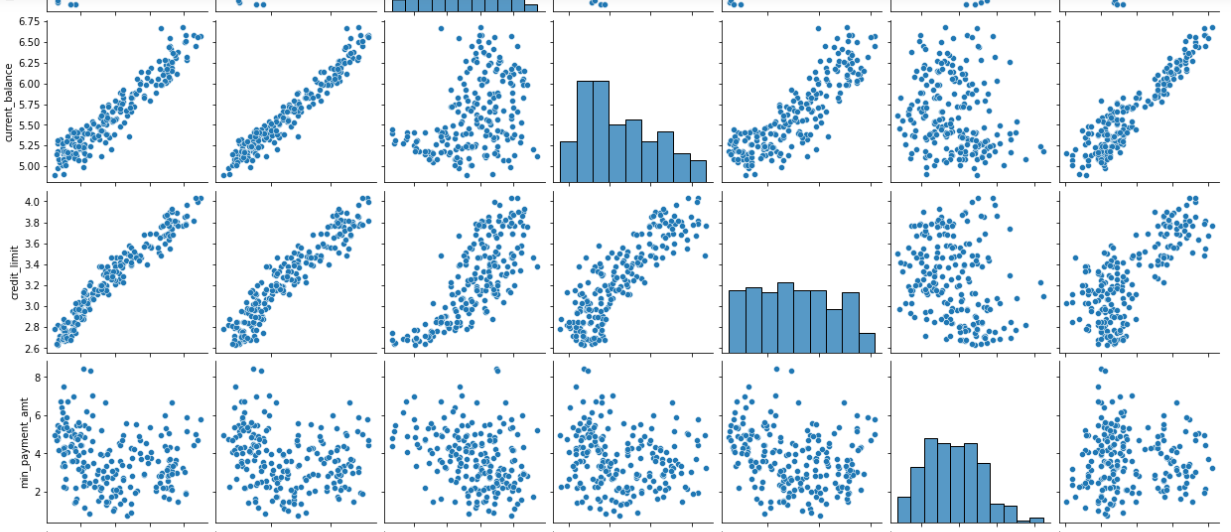




*Figure 7 histogram and box plot of variable*

* Probability of full payment and min payment amount has outliers.
* From the boxplots and skewness values, it can be inferred that data of all the column is moderately skewed towards .

**Bivariate Analysis**



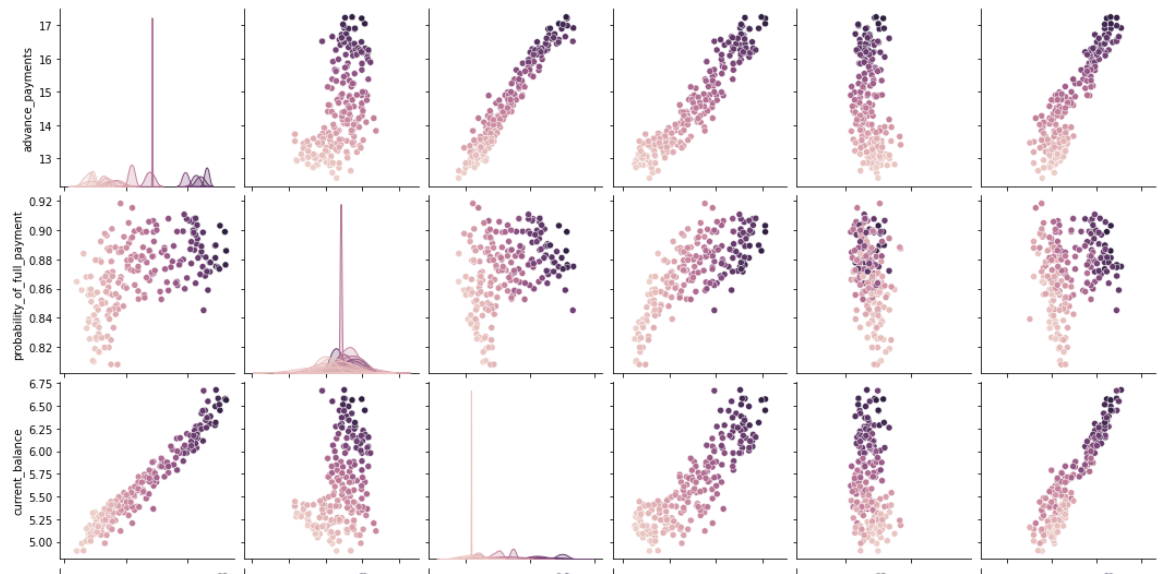
*Figure 8 Scatterplot of all the variable*



*Figure 9 Heat map of all the variables*

From the scatterplot and heat map we can infer that:

* spending has a strong correlation with all the variables except min payment amount.
* Advance payment has a strong correlation with all the variables except min payment amount.
* Min payment amount has a week correlation with all the variables.
* All the variables except min payment amount have strong correlation with each other.

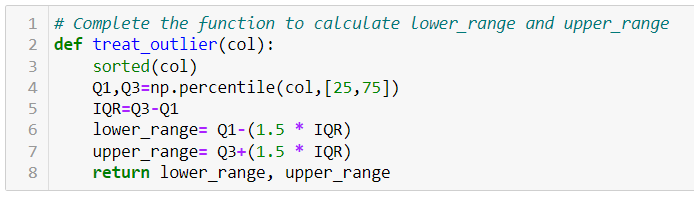


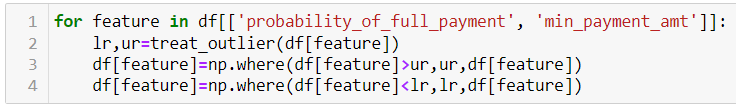
*Figure 10 Heat map of all the variables, spending as a hue*

From the heat map we can infer that:

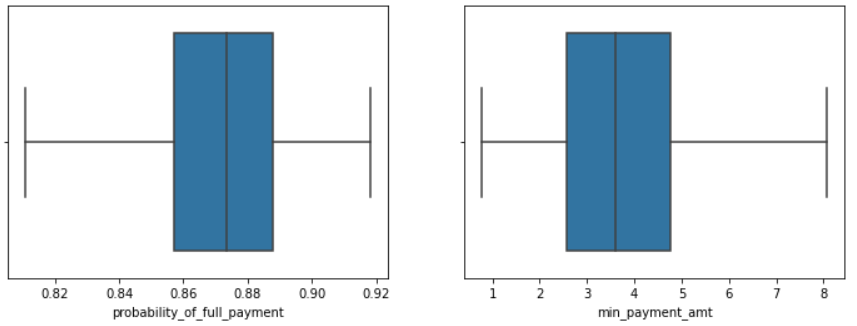
* People with higher current balance, higher credit limit has higher spending power.
* People with higher spending power do higher no of advance payment.
* People with higher spending power do higher no of max spent in single shopping.
* Min payment amount has a week correlation with all the variables.

**Outlier treatment:** Using interquartile range to treat outliers.





**Boxplot of variables after outlier treatment:**



*Figure 11 box plot of variable after outlier treatment*

1.2 Do you think scaling is necessary for clustering in this case? Justify The learner is expected to check and comment about the difference in scale of different features on the bases of appropriate measure for example

std dev, variance, etc. Should justify whether there is a necessity for scaling and which method is he/she

using to do the scaling. Can also comment on how that method works.

**Soln 1.2**

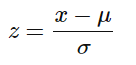
Yes, we will be scaling the variables as we can see from descriptive analysis that all the variables are having different scales.

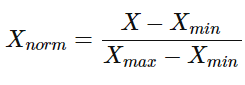
* Clustering is essentially “grouping close things together and distant things separate”.
* If you don’t normalize your features, you will end up giving more weight to some features than others.

**Distance-based algorithms** like clustering, K-means are most affected by the range of features. This is because behind the scenes they are using distances between data points to determine their similarity and hence perform the task at hand.

Therefore, we scale our data before employing a distance-based algorithm so that all the features contribute equally to the result

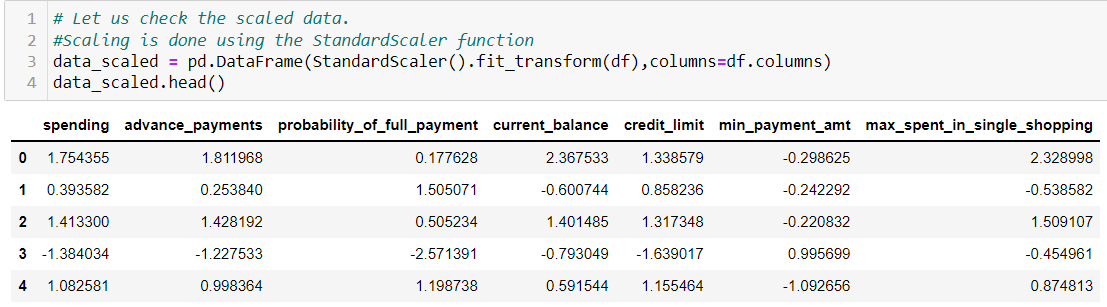
There are several ways to do feature scaling. Below mentioned is the 2 of the most commonly used feature scaling techniques:

* **Z-score normalization:** The result of **standardization**is that the features will be rescaled so that they’ll have the properties of a standard normal distribution with μ=0μ=0 and σ=1.

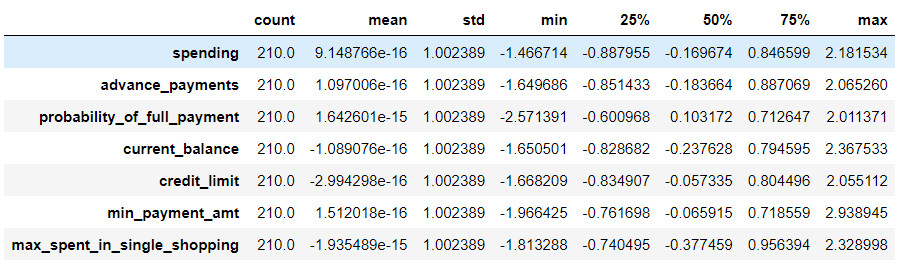
* Min-Max Scaling: In this approach, the data is scaled to a fixed range - usually 0 to 1.

We are using **Z-score normalization to scale the variables.**





**Descriptive analysis of scaled data:**



* After scaling the standard deviation become 1 for all the variables and mean become closer to 0.
* Now all the variables are on same scale.

1.3 Apply hierarchical clustering to scaled data (3 pts). Identify the number of optimum clusters using Dendrogram and briefly describe them (4). Students are expected to apply hierarchical clustering. It can be obtained via Fclusters or Agglomerative Clustering. Report should talk about the used criterion, affinity and linkage. Report must contain a Dendrogram and a logical reason behind choosing the optimum number of clusters and Inferences on the dendrogram. Customer segmentation can be visualized using limited features or whole data but it should be clear, correct and logical. Use appropriate plots to visualize the clusters.

**Soln 1.3**

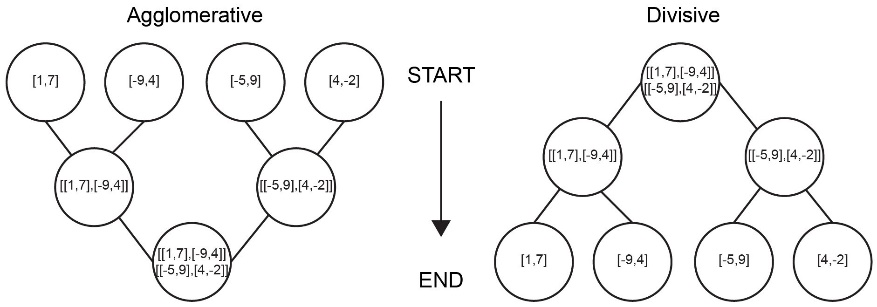
Clustering is the method of dividing objects into sets that are similar, and dissimilar to the objects belonging to another set. There are two different types of clustering, each divisible into two subsets

* Hierarchical clustering
  1. Agglomerative
  2. Divisive
* Partial clustering
  1. K-means
  2. Fuzzy c-means

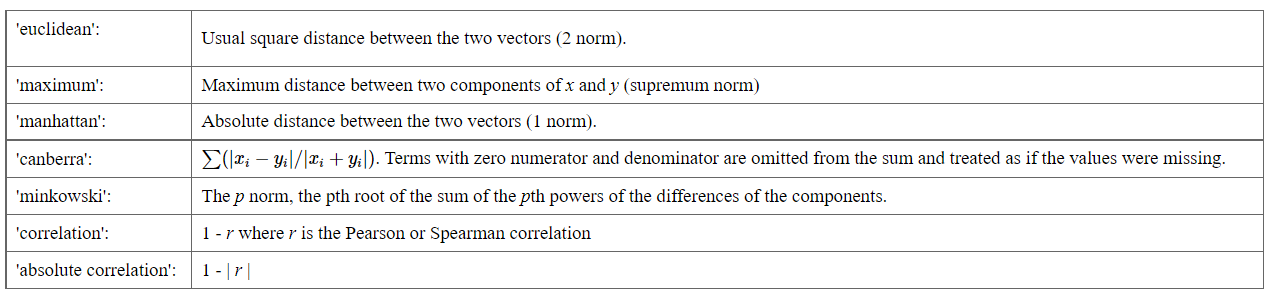
**Types of Hierarchical Clustering**

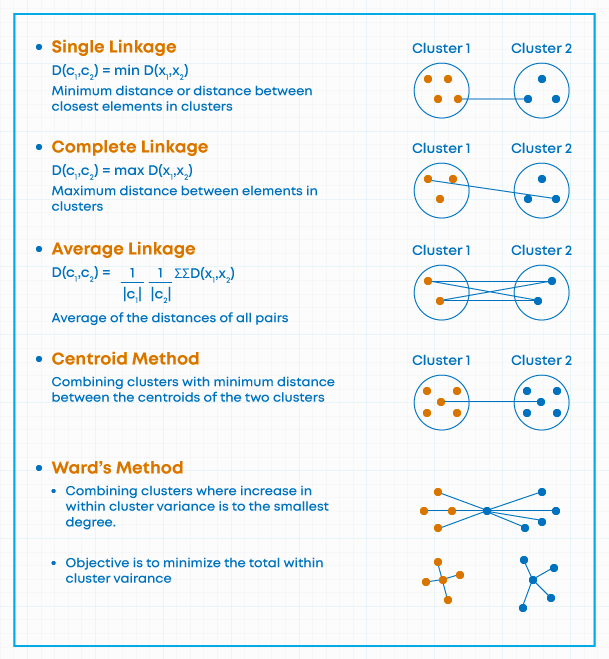
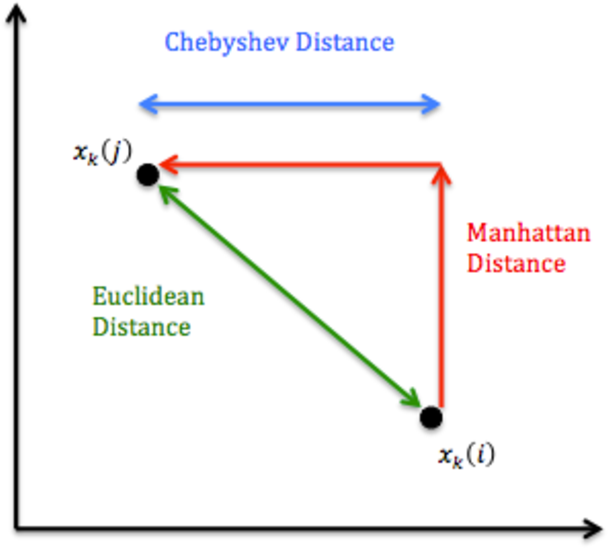
## Agglomerative hierarchical clustering: Initially consider every data point as an **individual** Cluster and at every step, **merge** the nearest pairs of the cluster. (It is a bottom-up method).

## Divisive Hierarchical clustering: We can say that the Divisive Hierarchical clustering is precisely the **opposite** of the Agglomerative Hierarchical clustering.

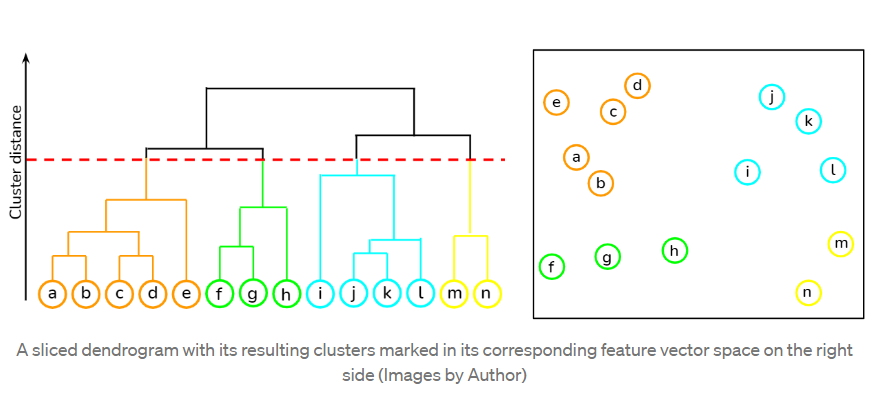


**For hierarchical clustering the distance measures between two data points and linkages function tells you to measure the distance between clusters:**

For most common hierarchical clustering software, the default distance measure is the Euclidean distance

**Dendrogram** is a type of [tree diagram](https://www.statisticshowto.com/how-to-use-a-probability-tree-for-probability-questions/) showing hierarchical clustering — relationships between similar sets of data. They are frequently used in biology to show clustering between genes or samples, but they can represent any type of grouped data.

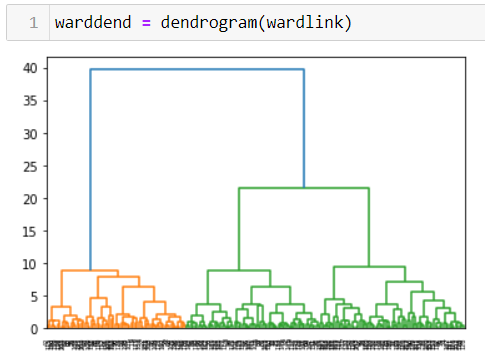


Performing Hierarchical Clustering with the 'scipy' package



Let us now try to cluster the data with the Euclidean distance and Ward's method for linkage.





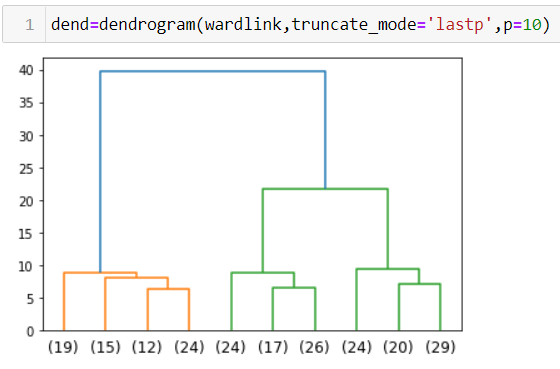
*Figure 12 Dendrogram*

Summarizing:

* horizontal lines are cluster merges
* vertical lines tell you which clusters/labels were part of merge forming that new cluster
* heights of the horizontal lines tell you about the distance that needed to be "bridged" to form the new cluster.

### **Dendrogram Truncation:**

* Truncated\_mode=lastp -show only the last p merged clusters
* P=10 – show only the last p merged cluster.

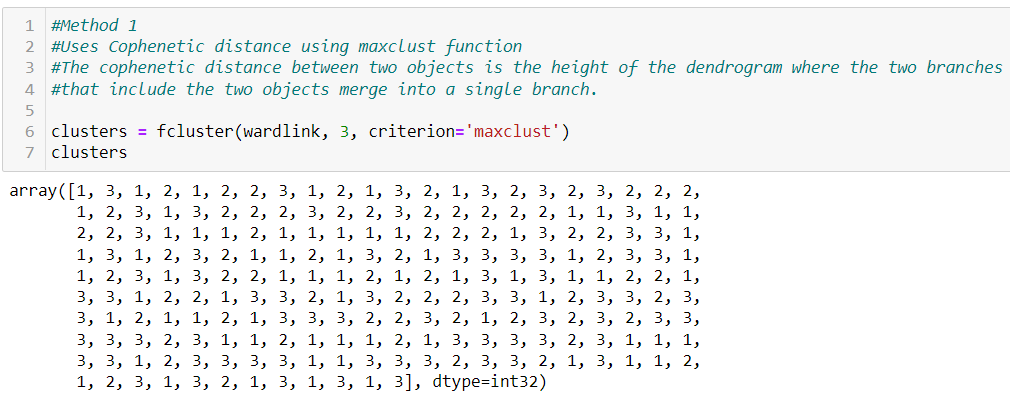


*Figure 13 Dendrogram after truncation*

Now that we have visualized the number of clusters, we need to cluster the data according to their similarity metrics

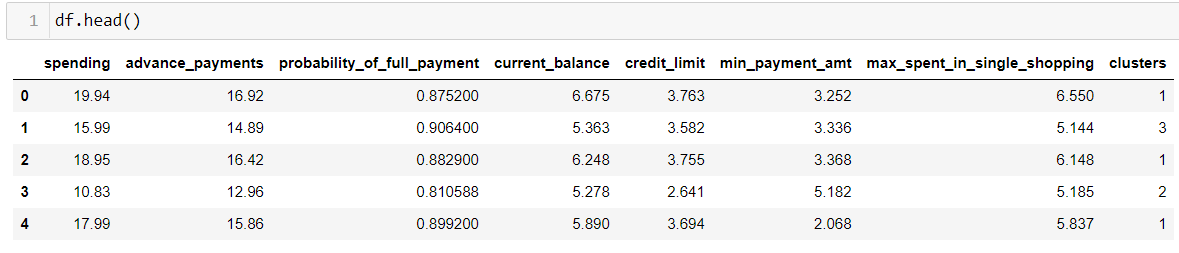
From the dendrogram, we see that 2 clusters are optimum. Thus, we are going to form 3 clusters based on the 'maxclust' criterion in the fcluster package.



We will use fcluster function in python and use maxclust as 3 to form 3 clusters post which we create a new column in the original data frame which assigns cluster values to each of the records. This will give us the records that belong to each of the 3 clusters.



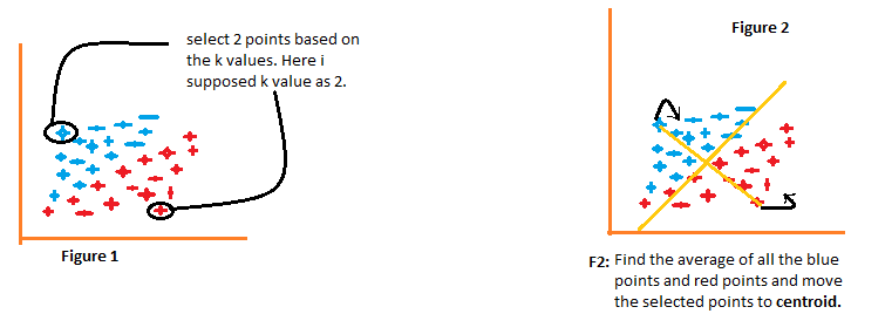
**Viewing the dataset after clustering:**

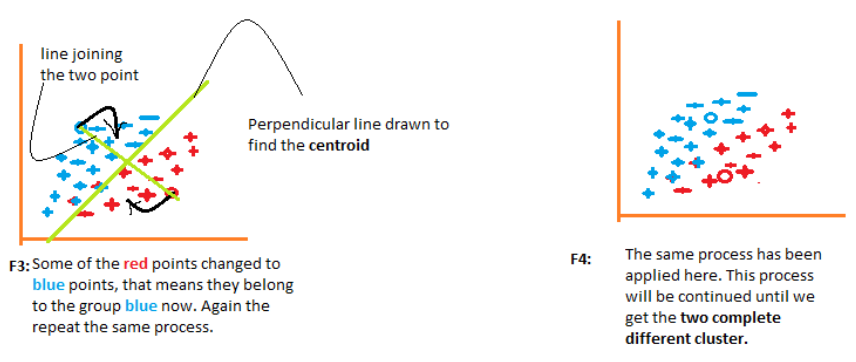


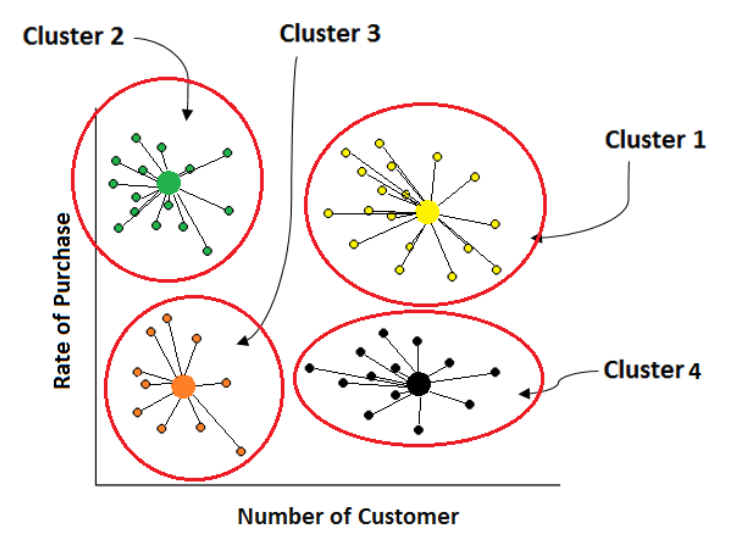
1.4 Apply K-Means clustering on scaled data and determine optimum clusters (2 pts). Apply elbow curve (3 pts). Interpret the inferences from the model (2.5 pts). K-means clustering code application with different number of clusters. Calculation of WSS(inertia for each value of k) Elbow Method must be applied and visualized with different values of K. Reasoning behind the selection of the optimal value of K must be explained properly. Report must contain logical and correct explanations for choosing the optimum clusters using the elbow method. Append cluster labels obtained from K-means clustering into the original data frame. Customer Segmentation can be visualized using appropriate graphs.

**Soln 1.4**

K-Means Clustering is an [Unsupervised Learning algorithm](https://www.javatpoint.com/unsupervised-machine-learning), which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.





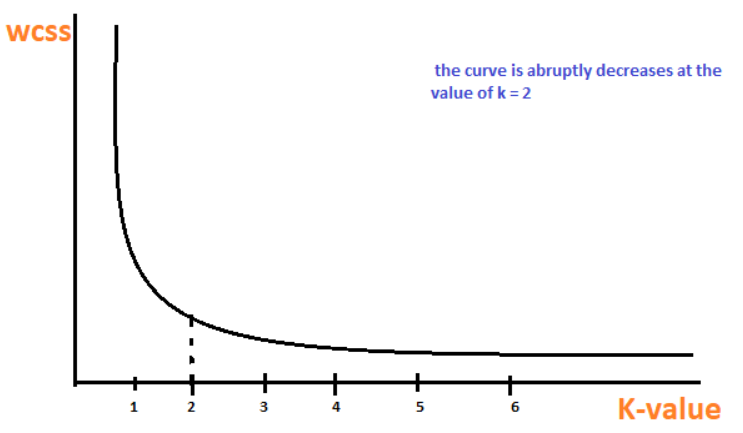
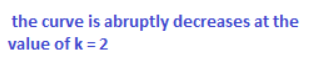
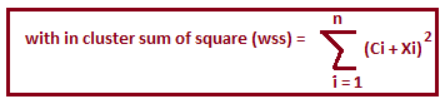


**The working of the K-Means algorithm is explained in the below steps:**

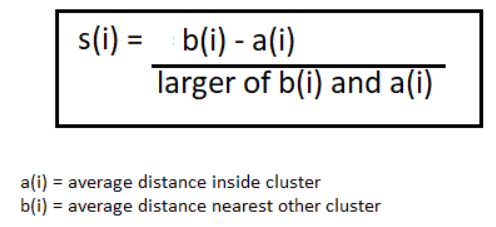
* **Step-1:** Select the number K to decide the number of clusters.
* **Step-2:** Select random K points or centroids. (It can be other from the input dataset).
* **Step-3:** Assign each data point to their closest centroid, which will form the predefined K clusters.
* **Step-4:** Calculate the variance and place a new centroid of each cluster.
* **Step-5:** Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.
* **Step-6:** If any reassignment occurs, then go to step-4 else go to FINISH.
* **Step-7**: The model is ready.

**How to choose no of clusters:**

* Elbow Method. -  It calculates the sum of the square of the points and calculates the average distance.



* Silhouette Method.- It calculates the silhouette coefficient of every point. It calculates the average distance of points within its cluster a (i) and the average distance of the points to its next closest cluster called b (i).



**To find the optimal value of clusters, the elbow method follows the below steps:**

* It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
* For each value of K, calculates the WSS value.
* Plots a curve between calculated WSS values and the number of clusters K.
* The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

**Problem 2: CART-RF**

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 and

compare the models' performances in train and test sets.

**2.1** Read the data, do the necessary initial steps, and exploratory data analysis (Univariate,

Bi-variate, and multivariate analysis).  
**2.2** Data Split: Split the data into test and train, build classification model CART, Random Forest  
**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.   
**2.4** Final Model: Compare all the models and write an inference which model is best/optimized.  
**2.5** Inference: Based on the whole Analysis, what are the business insights and recommendations

**Attribute** **Information:**

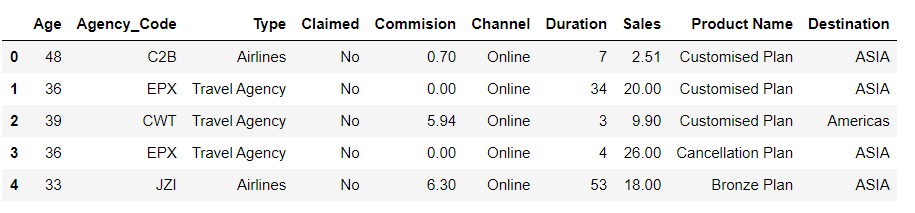
1. Target: Claim Status (Claimed)  
2. Code of tour firm (Agency Code)  
3. Type of tour insurance firms (Type)  
4. Distribution channel of tour insurance agencies (Channel)  
5. Name of the tour insurance products (Product)  
6. Duration of the tour (Duration in days)  
7. Destination of the tour (Destination)  
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

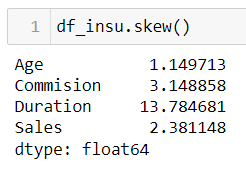
* Read the data and do exploratory data analysis (4 pts).
* Describe the data briefly. Interpret the inferences for each (2 pts). Initial steps like head () .info(), Data Types, etc . Null value check.
* Distribution plots(histogram) or similar plots for the continuous columns. Box plots, Correlation plots.
* Appropriate plots for categorical variables.
* Inferences on each plot. Summary stats, Skewness, Outliers proportion should be discussed, and inferences from above used plots should be there.
* There is no restriction on how the learner wishes to implement this but the code should be able to represent the correct output and inferences should be logical and correct.

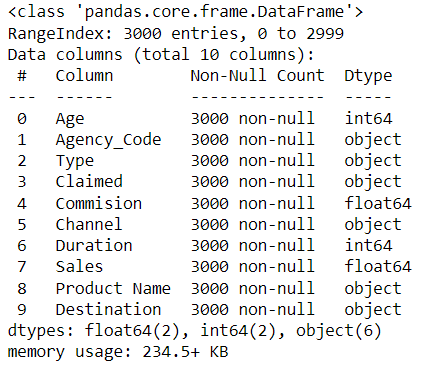
**Soln 2.1**

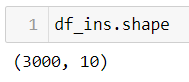
We start with looking at the first 5 and last 5 rows of the data as seen in Figure 1 and 2.

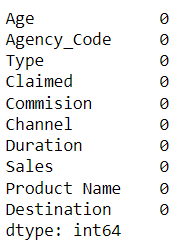






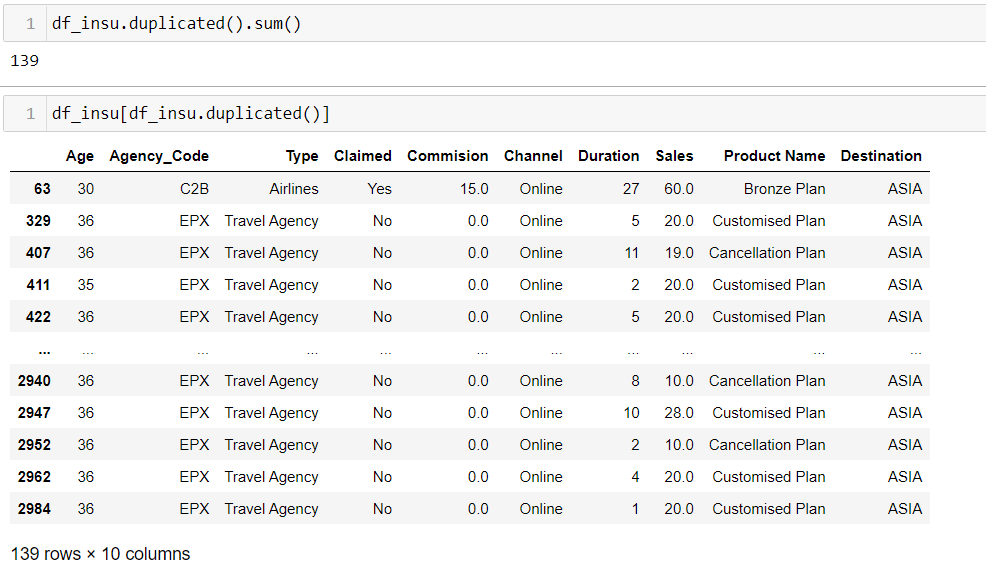






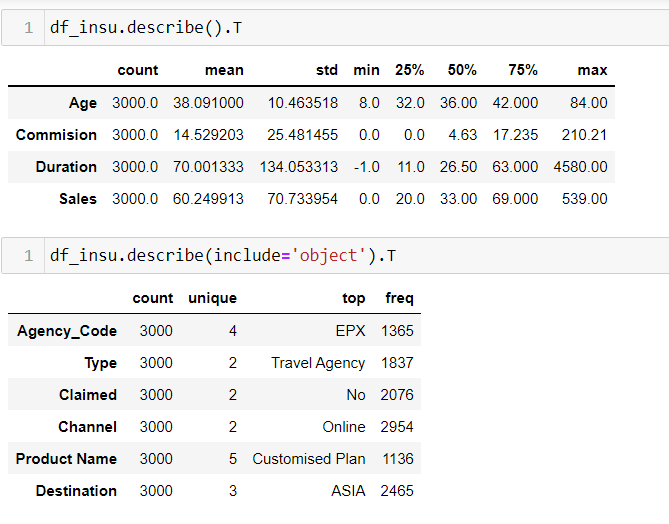
* The dataset has total 3000 rows and 10 columns.
* There are no null values in the data set.
* Numerical columns:
  + Age,
  + Commission,
  + Duration,
  + Sales.
* Categorical (Object type variable)
  + Agency code
  + Type
  + Claimed
  + Channel
  + Product name
  + Destination

**Duplicate records in the dataset:**



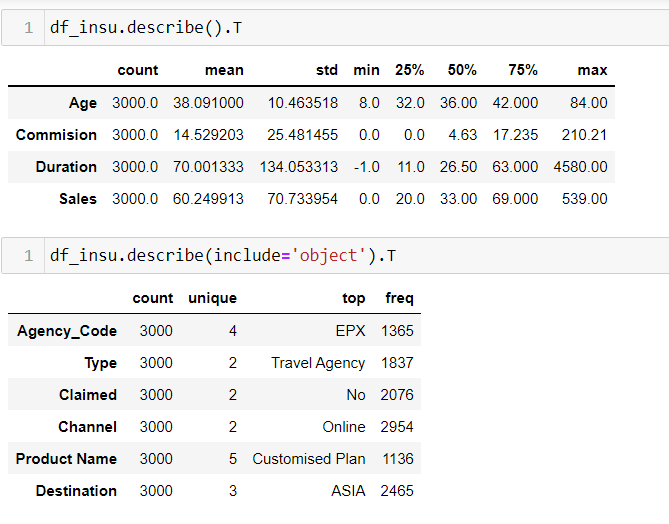
* There are 139 duplicate records in the data set.
* We are not treating the duplicate records as it could be possible that the travel agency has sold same travel insurance plan for the same insured age going for the same destination.

**Descriptive summary of the dataset for numerical variables:**



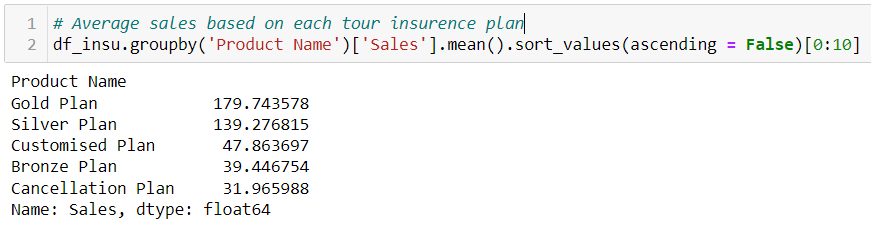
* Duration of trip is the most skewed distribution (min=-1 and max=4580). There is a possibility that people with customised plan have opted higher duration.
* Commision and sales are also right skewed.

**Descriptive summary of the dataset for categorical variables:**



* Most visited destination for which these insurance products were bought was Asia.
* Most people went for a customised plan when it came to the insurance product.
* Channel 'online' was used the most when it came to distribution of insurance products.
* The most used agency had a code 'EPX’, and the most used type of tour insurance firm was 'Travel Agency'.

**Amount worth of average sales per customer in procuring tour insurance policies in rupees based on each tour insurence plan**

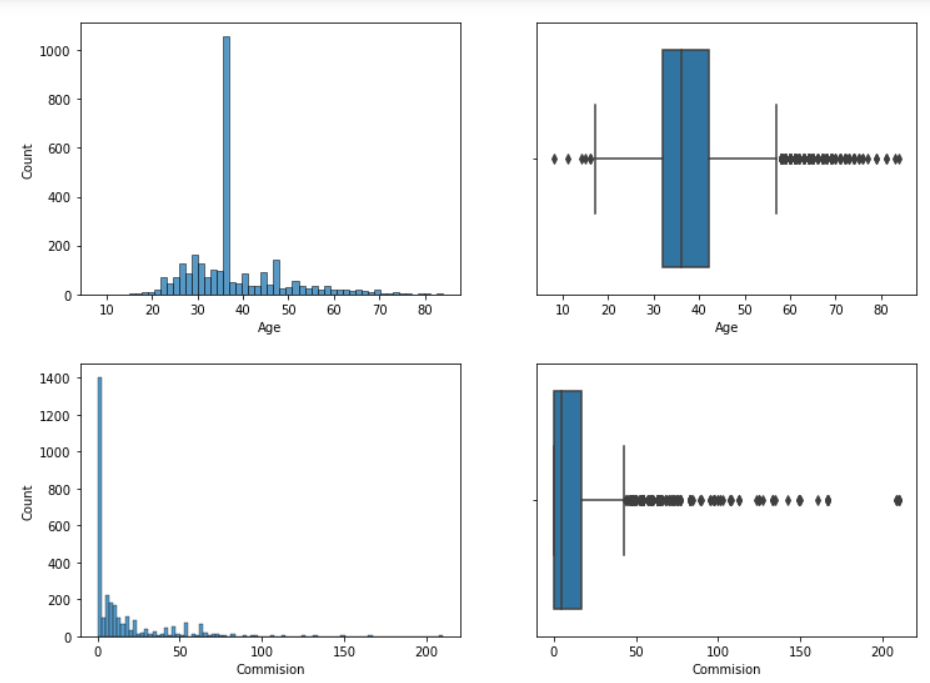


**The average commission received for tour insurance firm based on each tour insurence plan (Commission is in percentage of sales)**

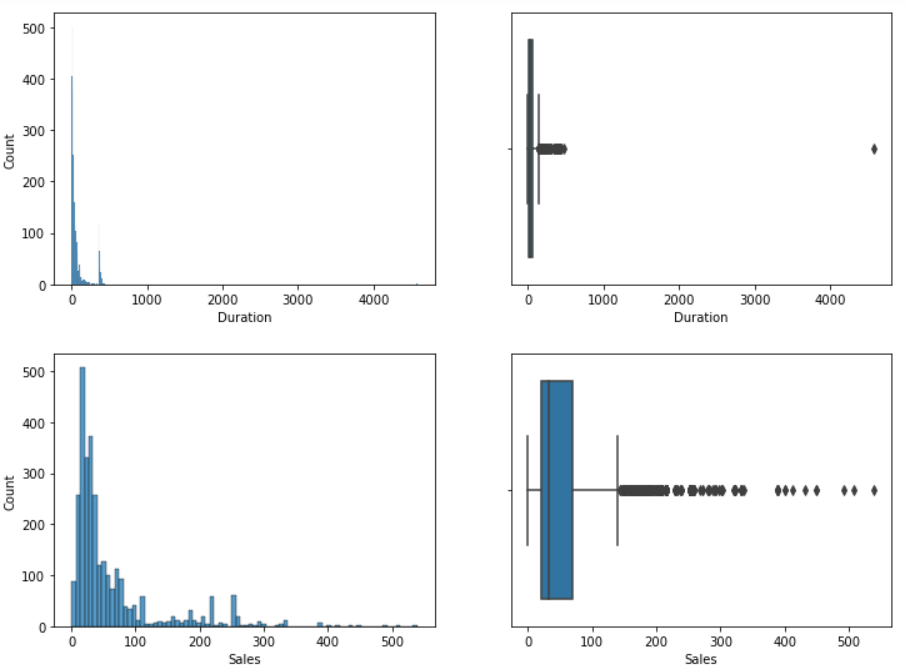


• We can see an order wherein the sales amount per customer and the commission decreases from Gold to Silver to Bronze plans with both being least for cancellation plans.

**Univariate Analysis**



*Figure 1 Histogram and boxplot of numerical variables*

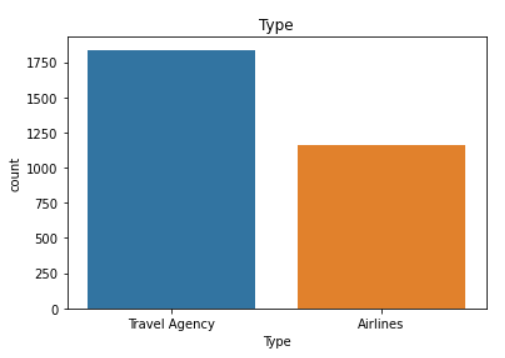


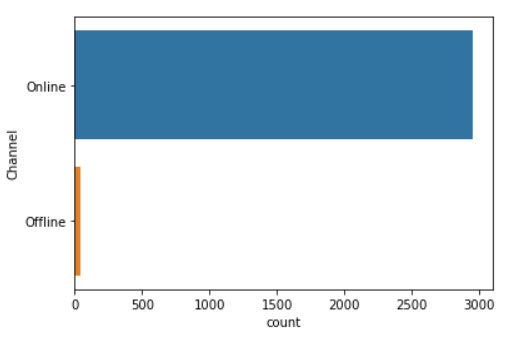
*Figure 2 Histogram and boxplot of numerical variables*

* Age of insured, Commission, Duration of trip and Sales are right skewed with significant higher no of records on the right side of boxplot.
* As we know if the duration of trip increased the insurance amount will also increase hence sales and commission will also increase.

**Count plot for different categorical variables:**

**Type of tour insurance firm and channel:**

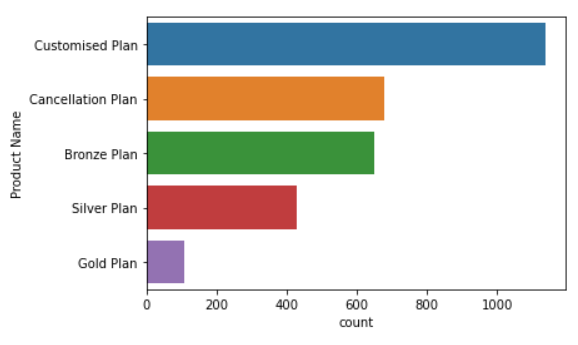
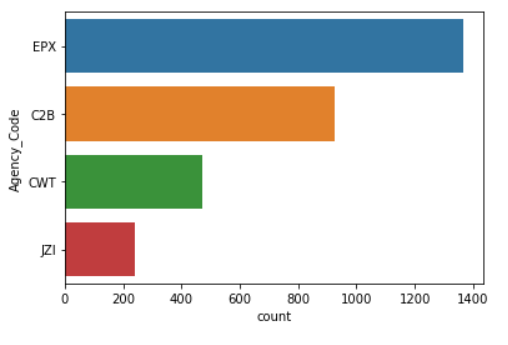




*Figure 3 and Figure 4 count plot for tour insurance firm and channel*

* Through travel agency more no of insurance records has taken as compared to airline.
* Through online channel mostly people have taken travel insurance as compared to offline channel.

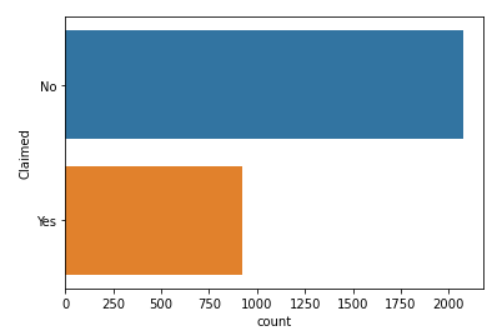
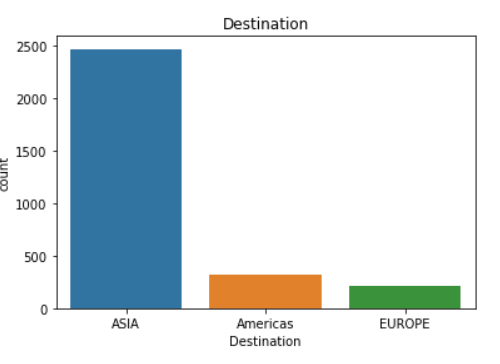
**Type of agency code by which tour insurance has booked and count of insurance product plans:**

*Figure 5 and Figure 6 count plot for agency code product name*

* Agency code EPX has higher no of insurance records as compared to other agency code.

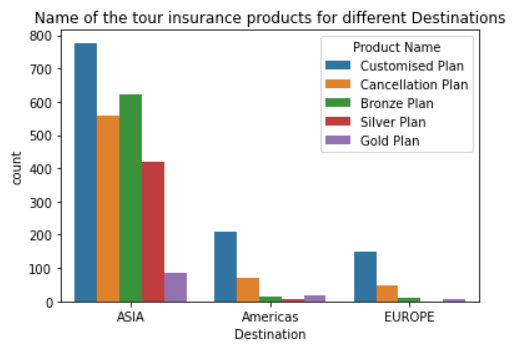
**Count of claimed and non-claimed insurance and different tour destinations:**



*Figure 6 and Figure 7 count plot for claimed and destination*

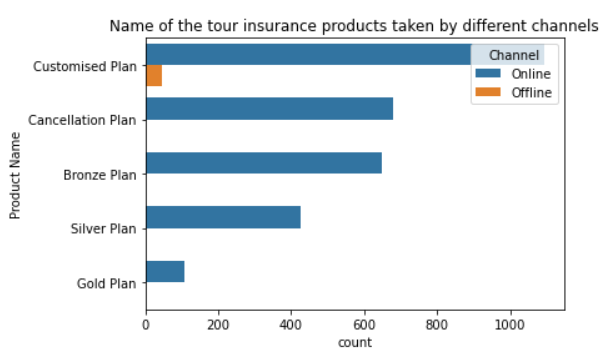
* The records have lesser values for insurance claimed as compare to non-claimed. Data is imbalanced.
* Destination Asia is the most popular destination with higher no of insurance plans sold.

**Bivariate Analysis**



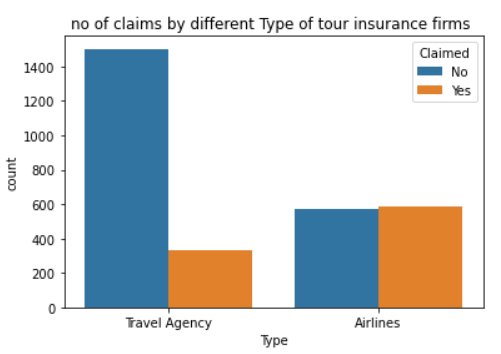
*Figure 8 Skewness of all the variables of data*

* Destination asia has the highest no of records for all tour insurance products.



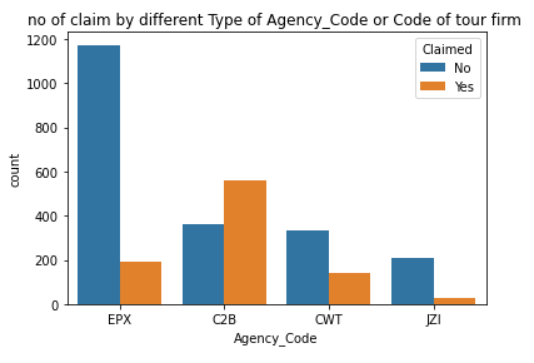
*Figure 9 Skewness of all the variables of data*

* Only customized plan has few insurances booking through offline channel.



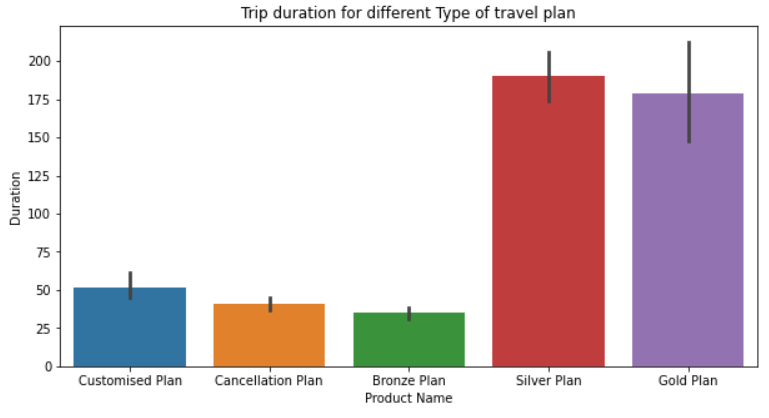
*Figure 10 Skewness of all the variables of data*

* Airlines insurance firms has significantly equal amount of claimed and non-claimed records.
* Through airline insurance firm more no of claimed records are there as compare to travel agency. Though a greater number of insurance sales through travel agencies as compared to airline companies.



*Figure 11 Skewness of all the variables of data*

* Agency code C2B has highest no of claimed records.
* Agency code EPX has higher no of records but comparative very less no of claimed insurance is there.



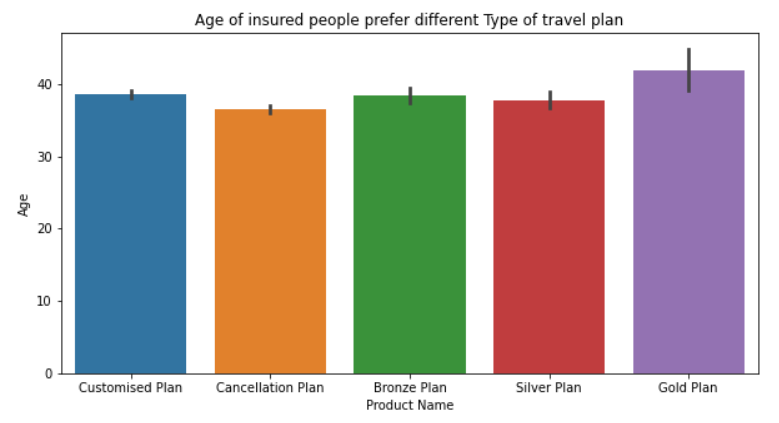
*Figure 12 Skewness of all the variables of data*

* Silver and golden tour insurance plan has higher trip duration as compare to other plans.



*Figure 13 Skewness of all the variables of data*

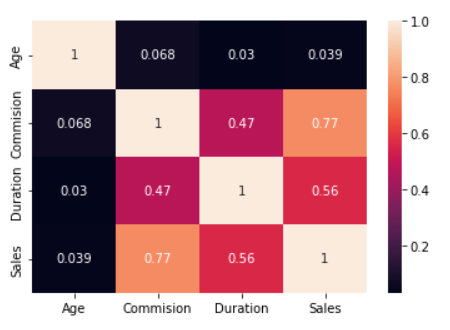
* Silver and golden tour insurance plan has higher sales as compare to other plans. Hence these insurance plans are more in demand than other insurance plans.



*Figure 14 Skewness of all the variables of data*

* All the Tour insurance plan has same age of insured.

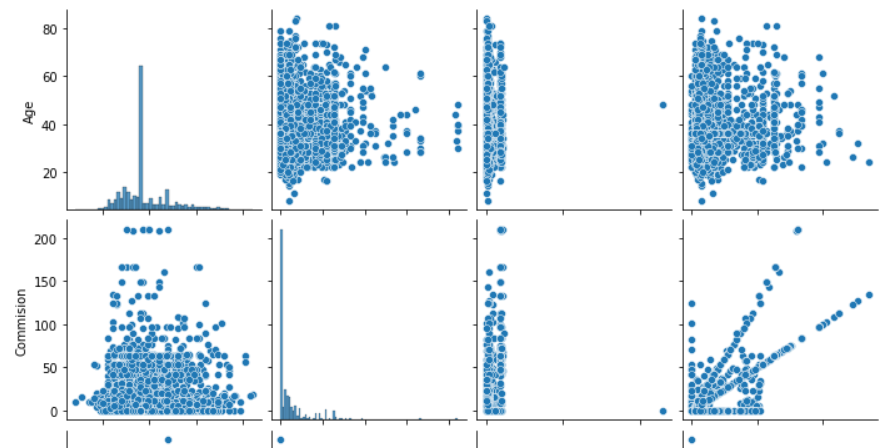
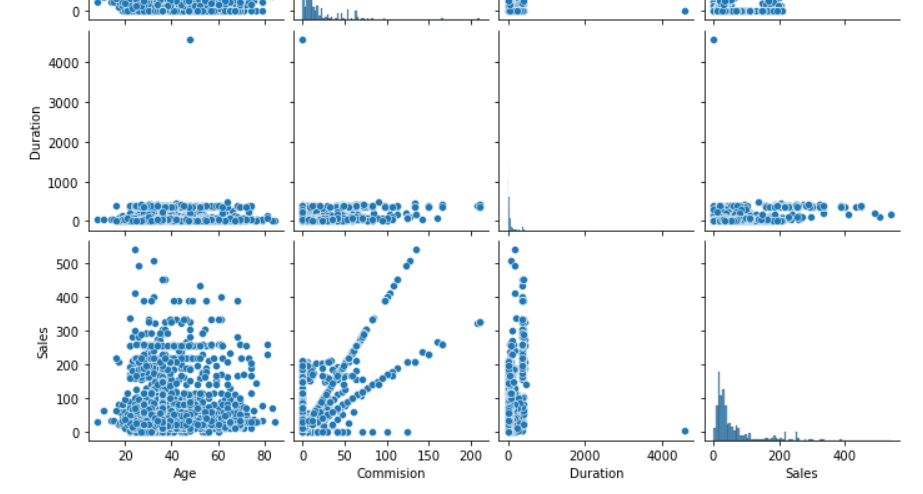
**Heat map of all the numerical variables:**



*Figure 15 Skewness of all the variables of data*

* From the heat map Sales spent per customer and Commission received on sale of insurance products have a strong positive correlation 0.77.
* Age has negligible correlation with the other numerical variables.
* Commission received on sale of insurance products and duration of trip also has a positive correlation of 0.56.
* Positive values denote positive linear correlation; • Negative values denote negative linear correlation; • A value of 0 denotes no linear correlation; • **The closer the value is to 1 or –1, the stronger the linear correlation**.

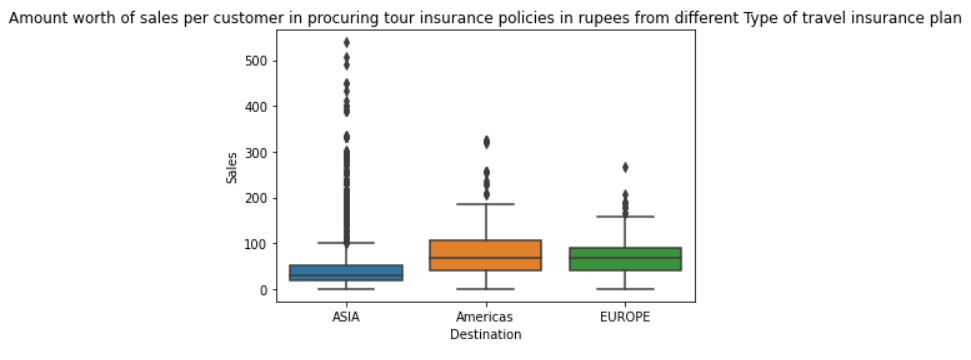
**Pair plot of all the numerical variables:**



*Figure 16 Skewness of all the variables of data*

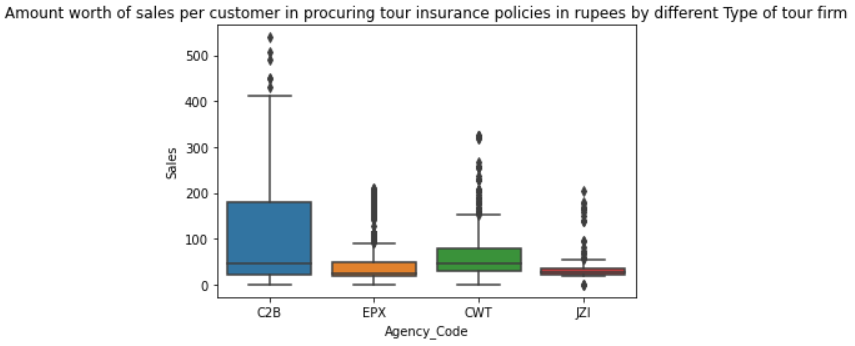
* From the scatterplots Sales spent per customer and Commission received on sale of insurance products look to have a strong positive correlation.
* Age has negligible correlation with the other numerical variables.

**Box plot of the numerical-categorical variables:**



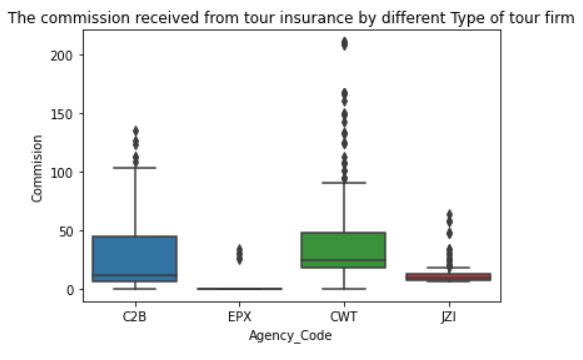
*Figure 17 Skewness of all the variables of data*

* Destination Asia has the higher values of amount worth of sales per customer in procuring tour insurance policies in rupees as compared to the mean sale.



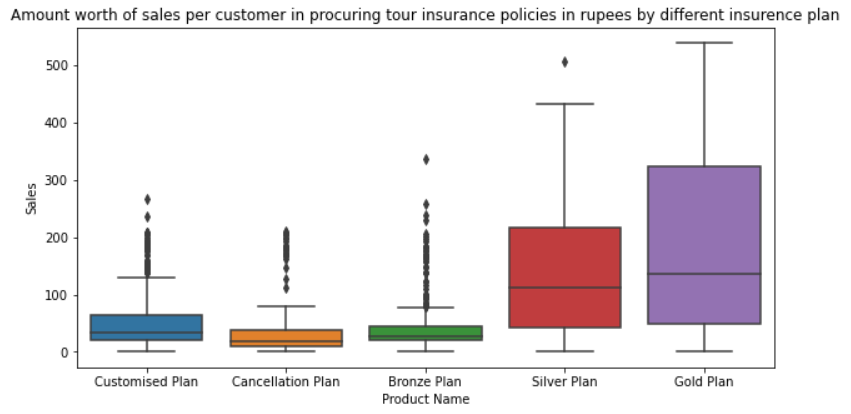
*Figure 18 Skewness of all the variables of data*

* Agency code C2B has the higher values of amount worth of sales per customer in procuring tour insurance policies in rupees as compared to the mean sale.
* Average sale value for all the agency code is significantly same.



*Figure 19 Skewness of all the variables of data*

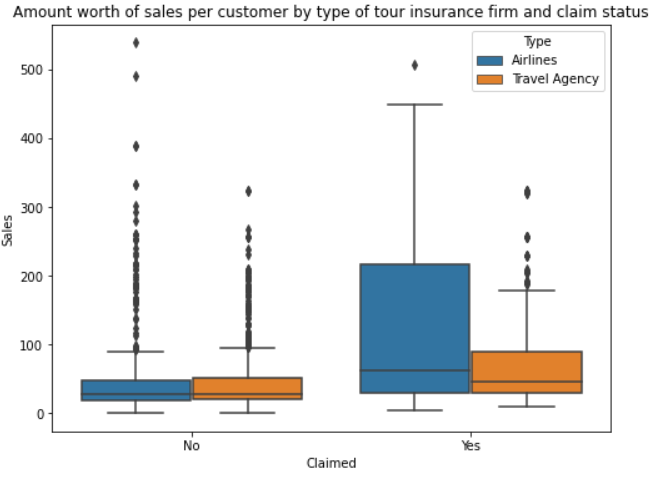
* Commission received from tour insurance is higher for agency CWT.
* Though sales per customer in procuring tour insurance policies in rupees is higher for agency C2B but their commission received from tour insurance is lower than agency CWT.



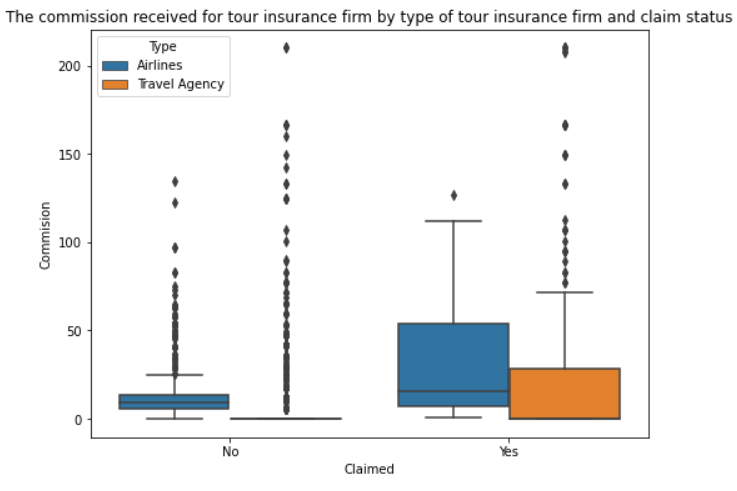
*Figure 20 Skewness of all the variables of data*

* Average sales per customer in procuring tour insurance policies in rupees is higher for silver and gold plan.

**Multivariate Analysis**

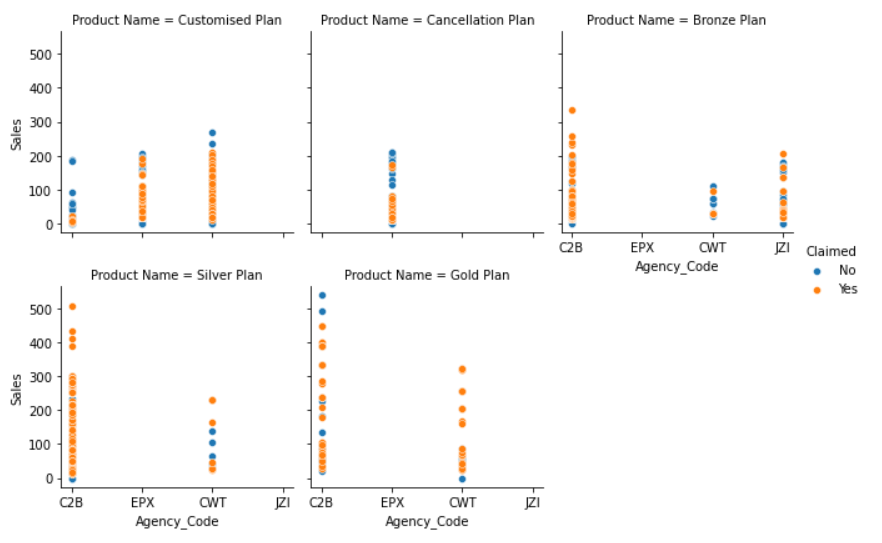


*Figure 21Skewness of all the variables of data*



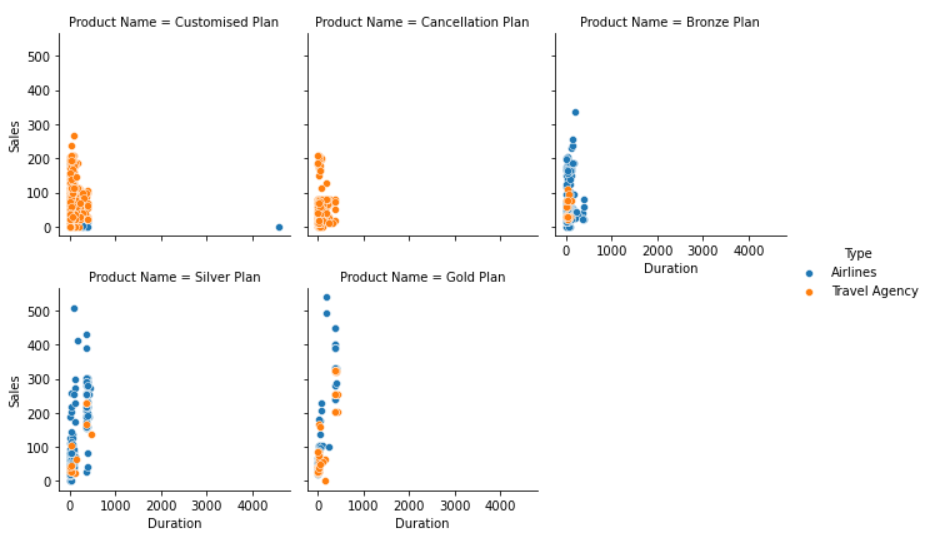
*Figure 22 Skewness of all the variables of data*

* The insurance products sold by airline agencies got a higher amount worth of sales per customer and commission received for tour insurance. Airline agencies also have higher claimed cases.



*Figure 23 Skewness of all the variables of data*

* Agency EPX only has the cancellation plan and customised plan for tour insurance.
* Agency C2B and CWT has all the tour insurance plan except cancellation plan.
* From the above plot we can see we are having lesser no of non-claimed records.
* Sales are higher for silver and gold plan with higher insurance claimed values.



*Figure 24 Skewness of all the variables of data*

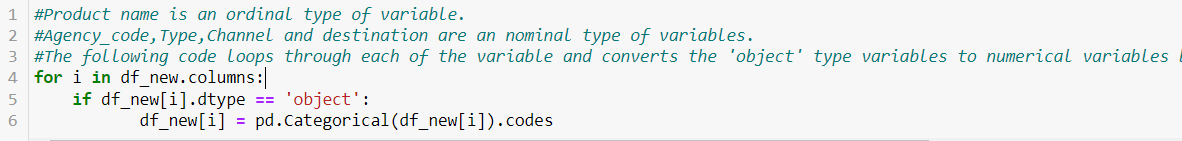
* Mostly all the customized and cancellation insurance plan are booked by travel agency.
* Airline has higher sale value for silver and gold plan.

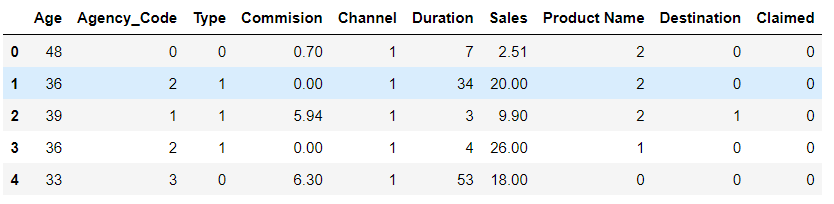
2.2 Data Split: Split the data into test and train(0.5 pts), build classification model CART (2.5 pts), Random Forest (2.5 pts). Object data should be converted into categorical/numerical data to fit in the models. (pd.categorical().codes(), pd.get\_dummies(drop\_first=True)) Data split, ratio defined for the split, train-test split should be discussed. Any reasonable split is acceptable. Use of random state is mandatory. Successful implementation of each model. Logical reason behind the selection of different values for the parameters involved in each model. Apply grid search for each model and make models on best\_params. Feature importance for each model.

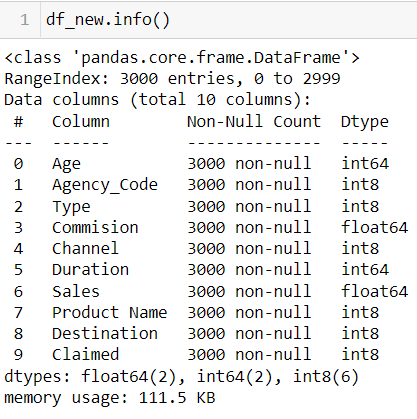
**Soln 2.2**

**Step-1:**

* Before splitting the data in to train and test, we will convert all the categorical variable into numerical by performing level encoding as model cannot take categorical values.
* The following code loops through each of the variable and converts the 'object' type variables to numerical variables by assigning ranks/numbers to each category.

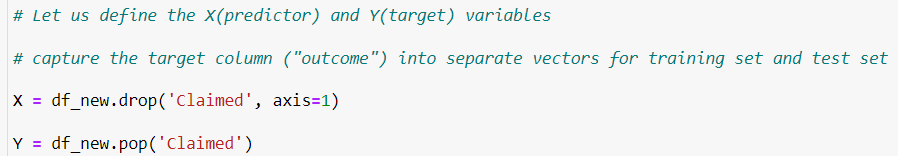






**Step-2:**

* We are prepping the data by segregating them into Target and independent variables to run this model going forward.
* Let us define the X(predictor) and Y(target) variables.
* Capture the target column ("Claimed") into separate vectors for training set and test set.



we have made a copy of the data frame as the 'pop' function removes that particular variable from the data frame and stores in another variable.

X=Stores all independent variable except dependent variable.

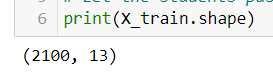
Y= stores only dependent variable (Claimed)

**Step-3:**

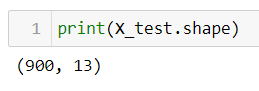
* After converting required categorical columns into numerical ones and before building the model, we should split the data into Train and Test.
* We will thus build a model on the training data and use this model to predict on the test data.
* **We will be doing a 70:30 split. 70% of the whole data will be used to train the data and then 30% of the data will be used for testing the model thus built.**

**We will split the data into following category:**

1. **X\_train** - This includes your all independent variables,these will be used to train the model, also as we have specified the test\_size = 0.3, this means 70% of observations from our complete data will be used to train/fit the model and rest 30% will be used to test the model.



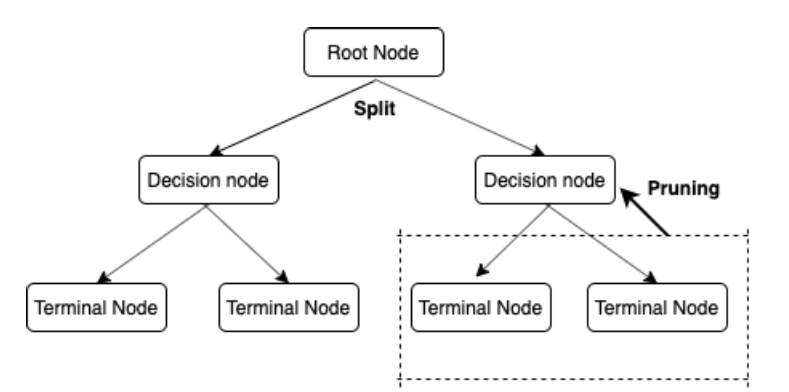
1. **X\_test** - This is remaining 30% portion of the independent variables from the data which will not be used in the training phase and will be used to make predictions to test the accuracy of the model.



1. **y\_train** - This is our dependent variable **‘Claimed’** which needs to be predicted by this model, this includes category labels against your independent variables, we need to specify our dependent variable while training/fitting the model.
2. **y\_test** - This data has category labels for our test data, these labels will be used to test the accuracy between actual and predicted categories.

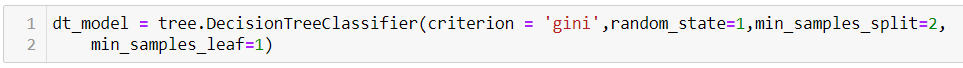
**1.Decision Tree**

**Decision Trees (DTs)** are a non-parametric supervised learning method used for [classification](https://scikit-learn.org/stable/modules/tree.html#tree-classification) and [regression](https://scikit-learn.org/stable/modules/tree.html#tree-regression). The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.



*Figure 25 Skewness of all the variables of data*

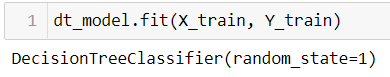
**We will start by building a very basic Decision Tree model.**



In the above code snippet, we have defined a Decision Tree (which is to be used for classification problems) with the splitting criteria for each node as 'gini'. The 'random\_state' parameter ensures that each time we run the code snippet the values remain the same.

In the above code snippet default values of 'min\_samples\_split' and 'min\_samples\_leaf' is taken as 2 and 1 respectively.

Now, that we have defined a Decision Tree, let us go ahead and build the model on the training data.

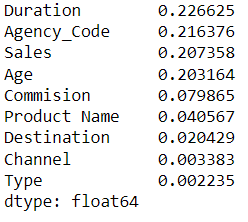


### **Visualizing the Decision Tree**

Now, that we have built the tree let us go ahead and visualize the tree to understand the various nuances of the Classification Tree that we just built. We can only share a little snip here as its not possible to capture complete tree here.This tree is unpruned .

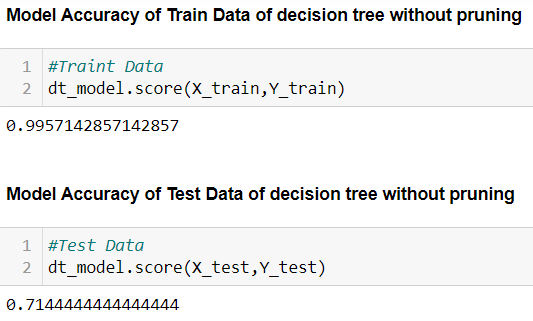
**Importance of features in the tree building :**

* The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance.



From the above output, we can see that 'Duration' is the most important variable followed by 'Sales' and so on.

**Let us take a look at the overall accuracy of the train and test data using the model that we just built**.



The accuracy on the Training Data is 99% and the accuracy on the Test Data is lesser (71%) substantially. The model has surely been overfitted. Thus, we need to prune or regularize the tree.

**Overfitting**:

Overfitting is a concept in data science, which occurs when a statistical model fits exactly against its training data. When this happens, the algorithm unfortunately cannot perform accurately against unseen data, defeating its purpose. **If our model does much better on the training set than on the test set**, then we're likely overfitting.

To reduce overfitting, we will now be regularizing or pruning the tree.

## Pruning/Regularizing the Tree

## *Regularize the Decision Tree and Check the train and test score after regularization. For Pruning/Regularizing the Tree we need to be sure as to what parameters and how to prune the tree.*

**Type of parameters to choose**

**Criterion –**The function to measure the quality of a split. Supported criteria are “gini” for the Gini impurity and “log\_loss” and “entropy” both for the Shannon information gain.

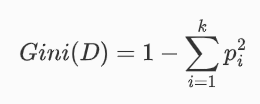
**max\_depth -**The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min\_samples\_split samples.

**min\_samples\_split-** The minimum number of samples required to split an internal node:

**min\_samples\_leaf -**The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least min\_samples\_leaf training samples in each of the left and right branches. This may have the effect of smoothing the model, especially in regression.

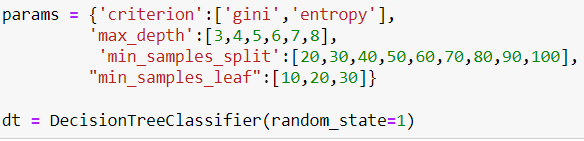
**Gini Impurity**: is a measurement used to build Decision Trees to determine how the features of a dataset should split nodes to form the tree.

Consider a dataset D that contains samples from k classes. The probability of samples belonging to class i at a given node can be denoted as pi. Then the Gini Impurity of D is defined as:



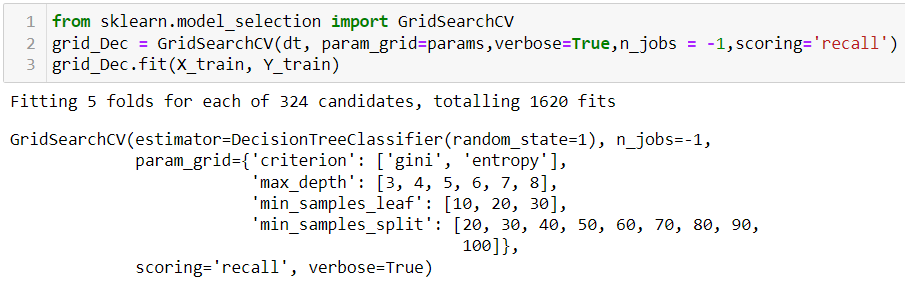
When training a decision tree, the attribute that provides the smallest GiniA(D) is chosen to split the node.

**Searching best parameters for the model:**

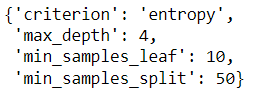
 

**Fitting the model into best parameters range and finding the most suited parameters for the model:**

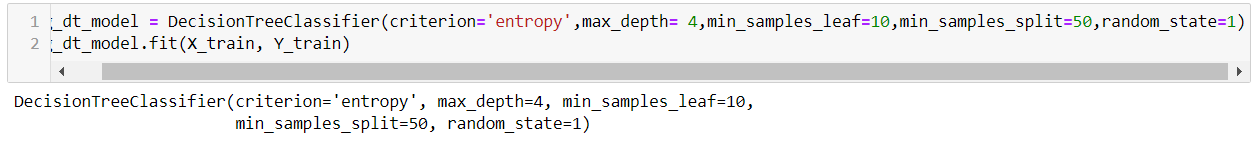
**Grid search** is a process that searches exhaustively through a manually specified subset of the hyperparameter space of the targeted algorithm.



**Below mentioned list in the best suited parameters for our decision tree model:**



**Fitting the model into best parameters :**

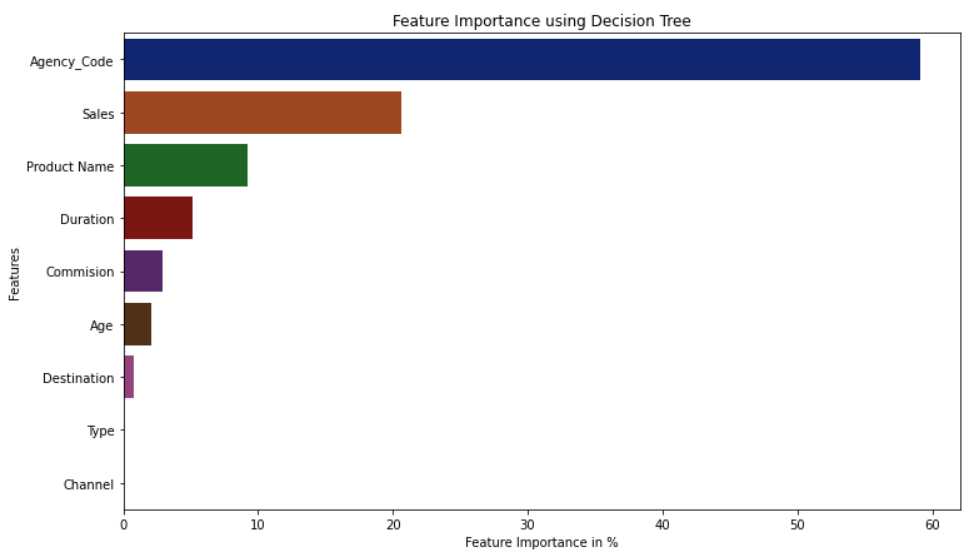


### **Visualizing pruned Decision Tree**

Now, that we have built the tree let us go ahead and visualize the tree to understand the various nuances of the Classification Tree that we just built.

**Importance of features in the tree building after pruning:**

* The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance.

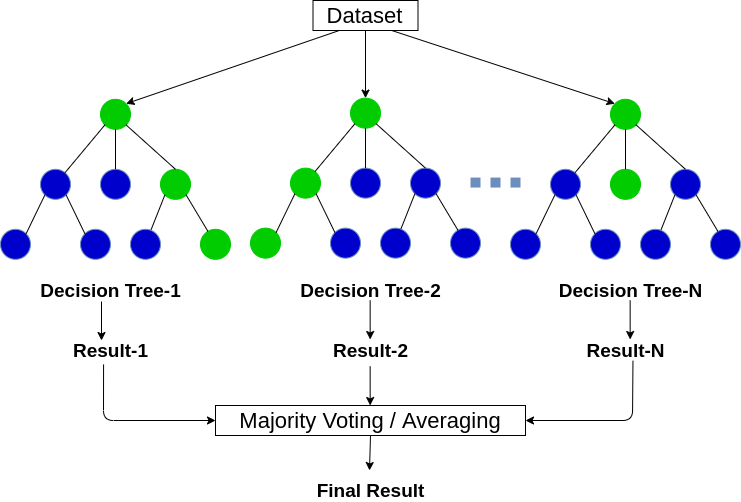


*Figure 26 Skewness of all the variables of data*

* The importance of features changed after we perform pruning.
* Now Agency code and sales becomes the top most important variable to predict claims.

**2. Random Forest model**

* Random forest is a *Supervised Machine Learning Algorithm* that is *used widely in Classification and Regression problems*. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.
* One of the most important features of the Random Forest Algorithm is that it can handle the data set containing *continuous variables* as in the case of regression and *categorical variables* as in the case of classification. It performs better results for classification problems.
* Bagging, also known as *Bootstrap Aggregation* is the ensemble technique used by random forest. In the case of regression, instead of determining the most populous vote the random forest will average the results of each decision tree. Because random forests utilize the results of multiple learners (decisions trees), random forests are a type of ensemble machine learning algorithm. Ensemble learning methods reduce variance and improve performance over their constituent learning models.



*Figure 27 Skewness of all the variables of data*

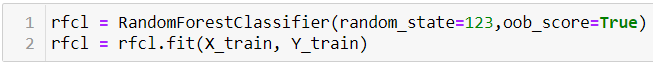
**Following hyperparameters increases the predictive power:**

* n\_estimators = number of trees in the forest
* max\_features = max number of features considered for splitting a node
* max\_depth = max number of levels in each decision tree
* min\_samples\_split = min number of data points placed in a node before the node is split
* min\_samples\_leaf = min number of data points allowed in a leaf node
* bootstrap = method for sampling data points (with or without replacement)
* Criterion =Here we can define the measure of impurity that works best. We will keep 'gini' and 'entropy' as the two options

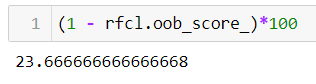
**Following hyperparameters increases the speed:**

* ***n\_jobs****–*it tells the engine how many processors it is allowed to use. If the value is 1, it can use only one processor but if the value is -1 there is no limit.
* ***random\_state****–*controls randomness of the sample. The model will always produce the same results if it has a definite value of random state and if it has been given the same hyperparameters and the same training data.
* ***oob\_score****– OOB* means out of the bag. It is a random forest cross-validation method. In this one-third of the sample is not used to train the data instead used to evaluate its performance. These samples are called out of bag samples.

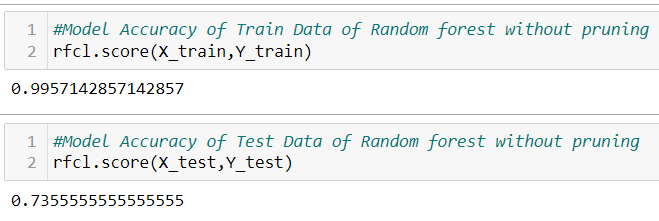
**We will start by building a basic Random Forest Model and then we will go ahead and tune the various parameters of the Random Forest model**



**OOB Error rate:**  is the average error for each calculated using predictions from the trees that do not contain in their respective bootstrap sample.



**Model Accuracy of Train and test Data of Random Forest without pruning**



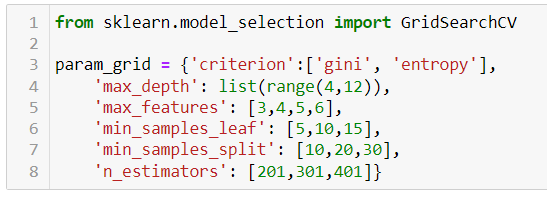
We see that the accuracy on the Training data is 0.99 and the accuracy on the test data is lower and can do with a better model. This shows a case of overfitting. We have allowed the Decision Trees inside the Random Forest algorithm to grow to their fullest. There is scope to regularize this particular Random Forest model to achieve comparative accuracy on both the Training and Test data.

We regularized the Decision Tree by looking at a largely overgrown tree. Here, in the case of Random Forest it is not possible to look at every tree and then go ahead and prune the trees. To make this job simpler we are going to use a command called GridSearchCV.

We will now build a random forest model now on the same train and test data. We will use parameters as follows to do a grid search and obtain the optimal parameters for the random forest.

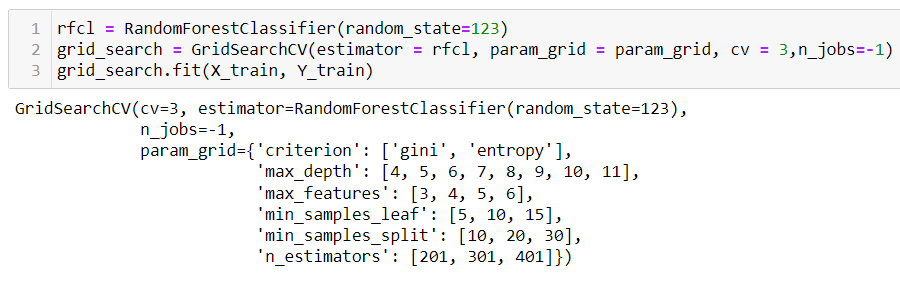
## Pruning/Regularizing the Tree

## Regularize the Decision Tree and Check the train and test score after regularization.For Pruning/Regularizing the Tree we need to be sure as to what parameters and how to prune the tree



Now that we have defined a dictionary in Python with all these parameters, it is time to use these parameters to build the model and check for the best set of parameters. The above dictionary makes sure that all the different combinations of values are used to build several Random Forest models.

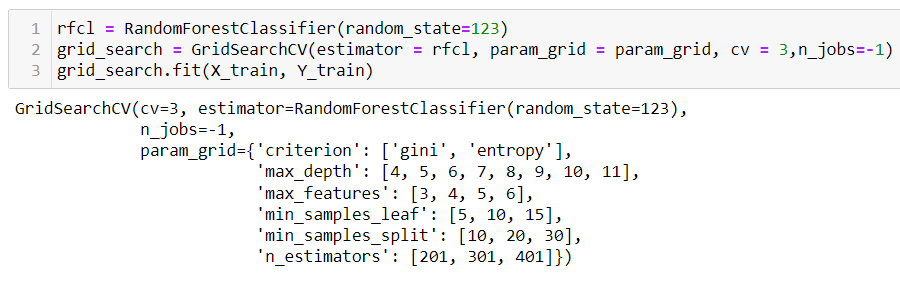
Now, we will define an empty Random Forest Classifier. This helps us to pass the Research parameters using the above defined dictionary.



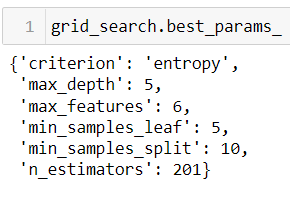
* n\_jobs=-1 allows paralle processing instead of serial processing In the above GridSearch .
* CV function we have passed the empty Random Forest Classifier defined by us, the parameters to be passed in this empty Random Forest function and we have also mentioned a 3-fold cross-validation as well.
* The value of the n\_jobs parameter as -1 helps us in running the models in parallel and use the complete strength of the processors. This helps fast track the computation.

**Fitting the model into best parameters range and finding the most suited parameters for the model:**

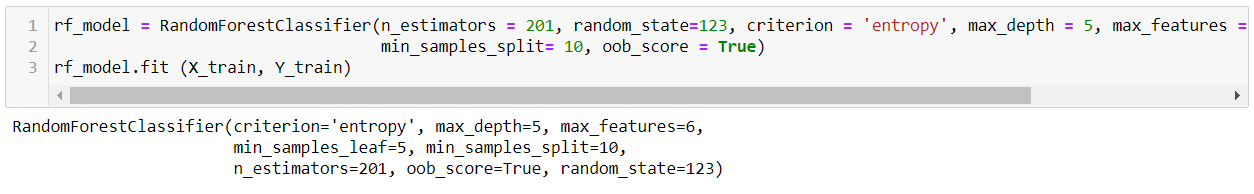
**Grid search** is a process that searches exhaustively through a manually specified subset of the hyperparameter space of the targeted algorithm.



**Below mentioned list in the best suited parameters for our Random forest model:**



**Fitting the model into best parameters :**



**Importance of features in the tree building after pruning:**



*Figure 28 Skewness of all the variables of dat*

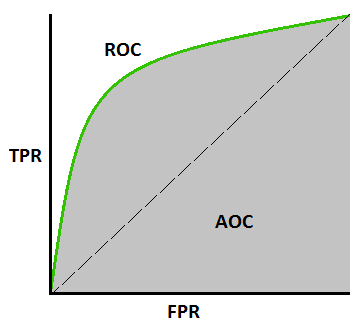
* Now Agency code and sales becomes the top most important variable to predict claims.

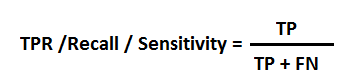
2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy (1 pts), Confusion Matrix (2 pts), Plot ROC curve and get ROC\_AUC score for each model (2 pts), Make classification reports for each model. Write inferences on each model (2 pts). Calculate Train and Test Accuracies for each model. Comment on the validness of models (overfitting or underfitting) Build confusion matrix for each model. Comment on the positive class in hand. Must clearly show obs/pred in row/col Plot roc\_curve for each model. Calculate roc\_auc\_score for each model. Comment on the above calculated scores and plots. Build classification reports for each model. Comment on f1 score, precision and recall, which one is important here.

Sol 2.3

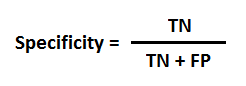
**AUC - ROC curve**: is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By analogy, the Higher the AUC, the better the model is at distinguishing between patients with the disease and no disease.

**Defining terms used in AUC and ROC Curve.**

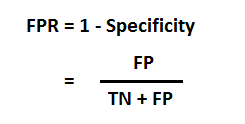
* **TPR (True Positive Rate) / Recall /Sensitivity**



* **Specificity**

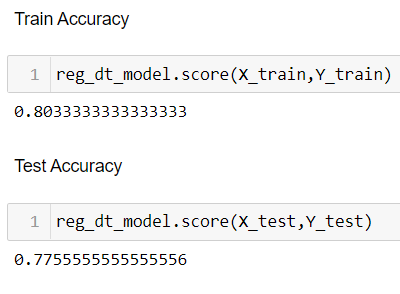


* **FPR**



**For Decision Tree**

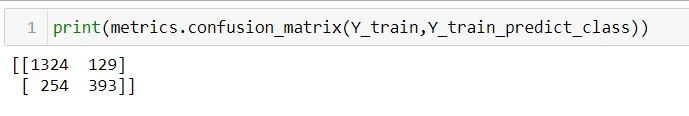
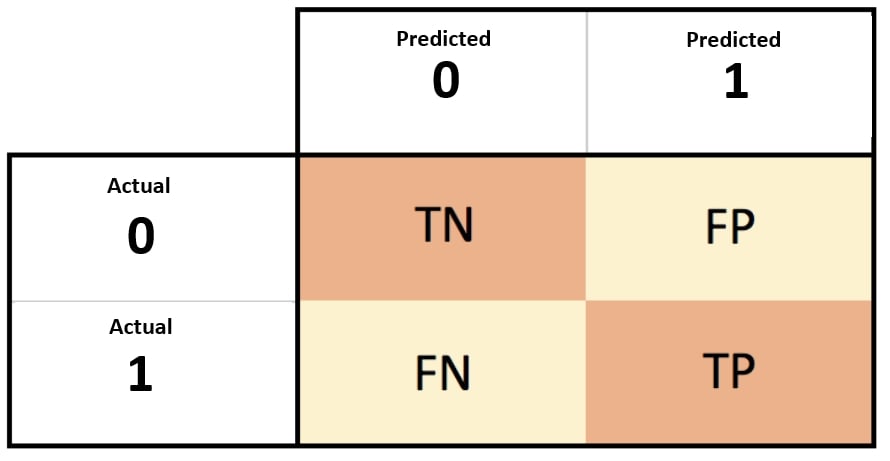
**Now we will check the accuracy for pruned tree:**

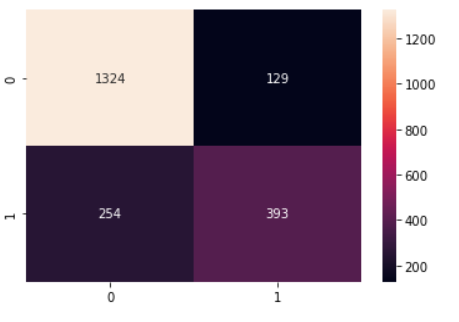


* **Train Accuracy-80%**
* **Test Accuracy -77.5%**

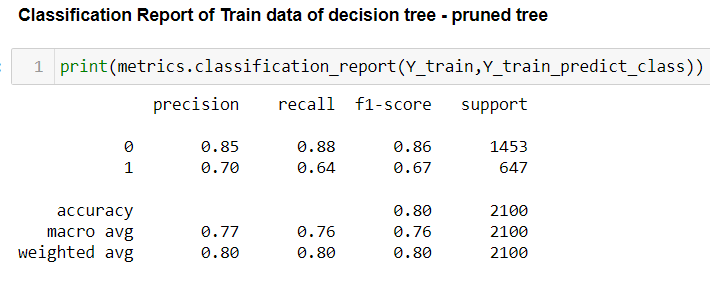
The accuracy on the Training Data is 80% and the accuracy on the Test Data is 77% substantially. It does not seem to overfitting

**Confusion matrix for Training Data**



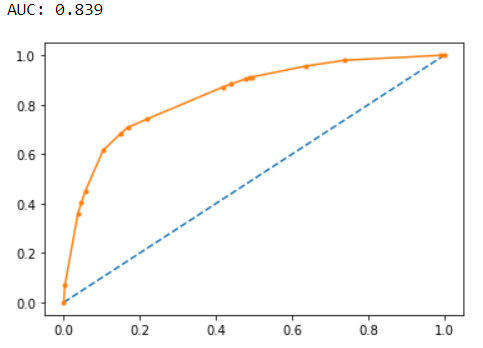


#### **Classification Report of Train data of decision tree - pruned tree**

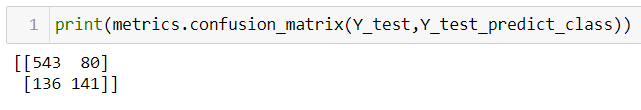


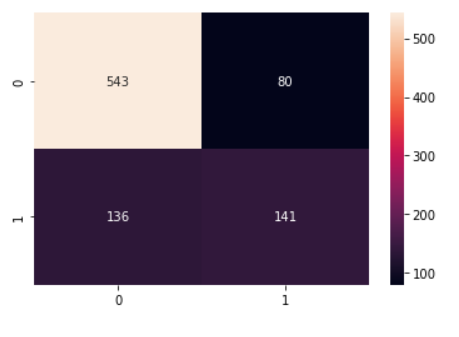
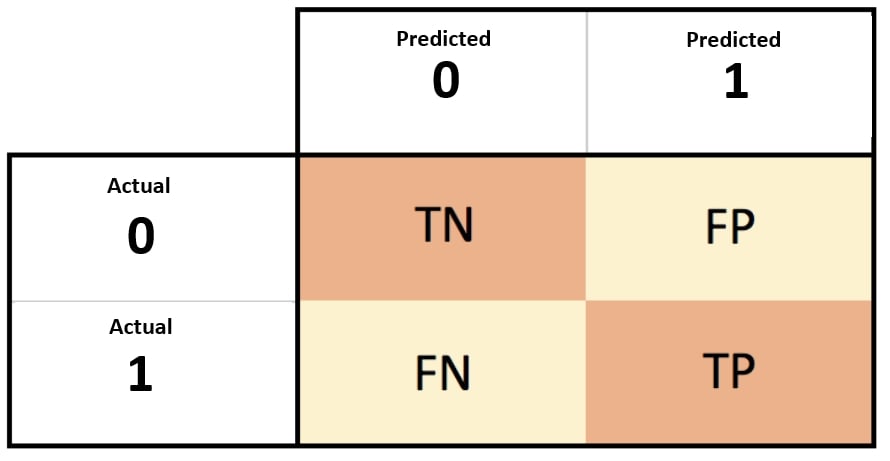
We have been able to predict 80% of the target variables correctly.

#### **Decision tree ROC & AUC For training data**



**Confusion matrix for Test Data**





**Classification Report of Test data of decision tree - pruned tree**



**Decision tree ROC & AUC For test data**



# Conclusion

Accuracy on the Training Data: 80%  
Accuracy on the Test Data: 78%

AUC on the Training Data: 83.9%  
AUC on the Test: 83.2%

Area under the curve on the training data is 84%, which indicates good performance that most of classes have been correctly classified. Whereas on the test data model performance is fine with AUC 83%, which is good compare to the performance of the training data.

Accuracy, AUC, Precision and Recall for test data is almost in line with training data. This proves very slight overfitting has happened, and overall, the model is a good model for classification

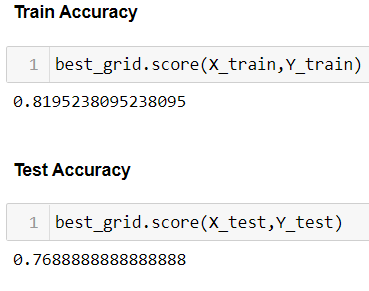
AUC is 0.83, it means there is a 83% chance that the model will be able to distinguish between positive class and negative class.

An excellent model has AUC near to the 1 which means it has a good measure of separability. A poor model has an AUC near 0 which means it has the worst measure of separability.

Agency code (Code of tour firm), Sales and Product name are the most important variables in predicting higher claim frequency.

**For Random Forest**

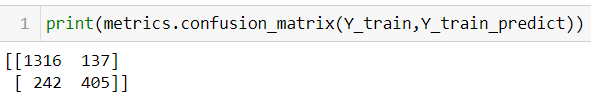
**Now we will check the accuracy for pruned tree:**



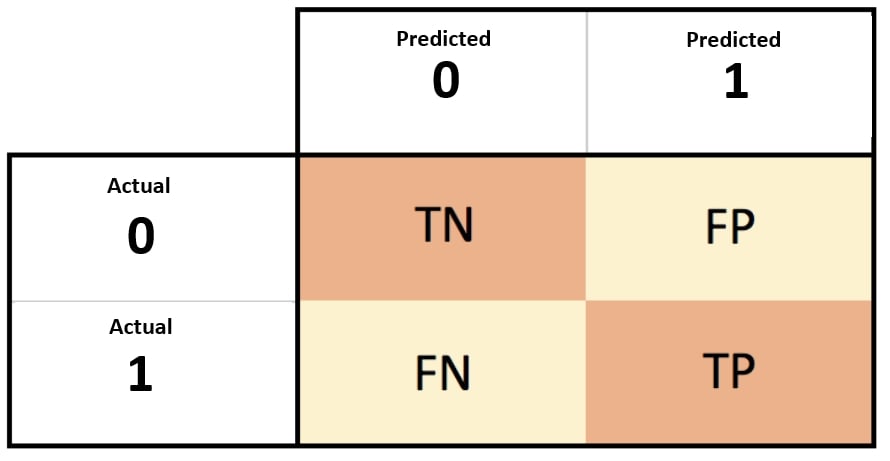
* **Train Accuracy-81%**
* **Test Accuracy -77%**

The accuracy on the Training Data is 81% and the accuracy on the Test Data is 77% substantially. It does not seem to overfitting

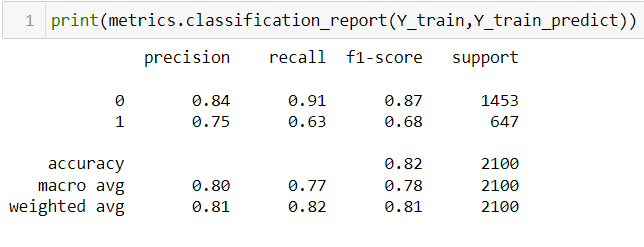
**Confusion matrix for Training Data**



**Heat Map - Confusion matrix of train data**

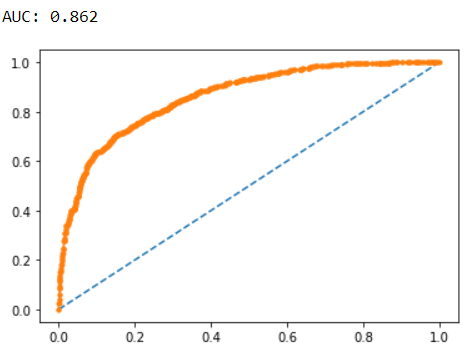
 

#### **Classification Report of Train data of Random forest - pruned tree**

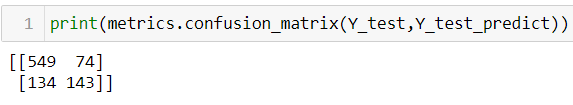


We have been able to predict 82% of the target variables correctly.

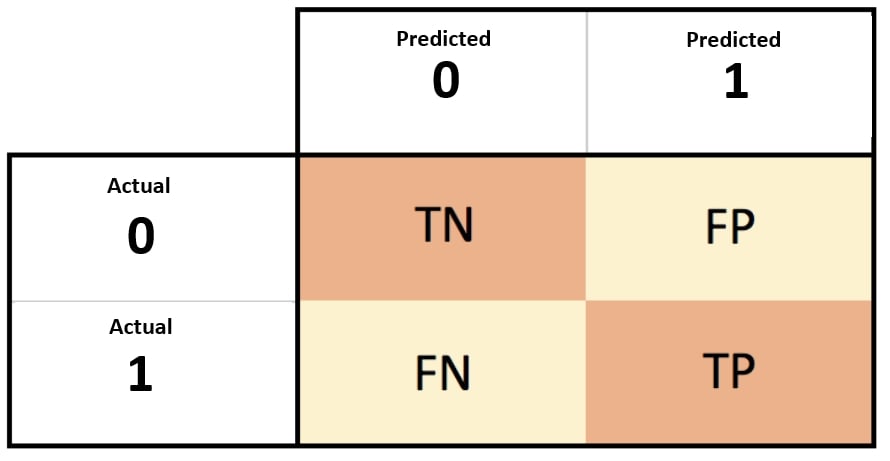
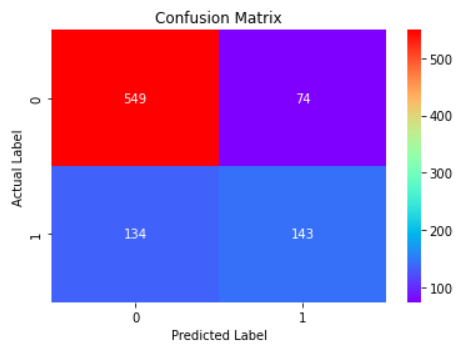
#### **Decision tree ROC & AUC For training data**



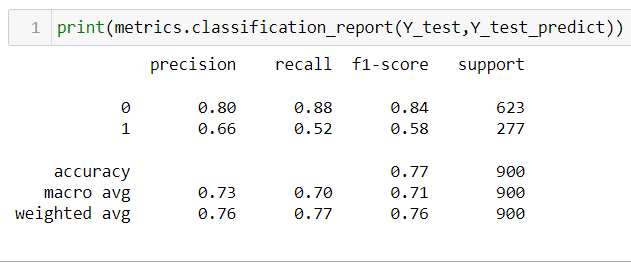
**Confusion matrix for Test Data**



**Heat Map - Confusion matrix of test data**

**Classification Report of Test data of Random forest - pruned tree**



**Decision tree ROC & AUC For test data**



# Conclusion

Accuracy on the Training Data: 81%  
Accuracy on the Test Data: 77%

AUC on the Training Data: 86.2%  
AUC on the Test: 81%

Area under the curve on the training data is 86%, which indicates good performance that most of classes have been correctly classified. Whereas on the test data model performance is fine with AUC 81%, which is good compare to the performance of the training data.

Accuracy, AUC, Precision and Recall for test data is almost in line with training data. This proves very slight overfitting has happened, and overall, the model is a good model for classification

AUC is 0.83, it means there is a 83% chance that the model will be able to distinguish between positive class and negative class.

An excellent model has AUC near to the 1 which means it has a good measure of separability. A poor model has an AUC near 0 which means it has the worst measure of separability.

Agency code (Code of tour firm), Sales and Product name are the most important variables in predicting higher claim frequency.

2.4 Final Model - Compare all models on the basis of the performance metrics in a structured tabular manner (2.5 pts). Describe on which model is best/optimized (1.5 pts ). A table containing all the values of accuracies, precision, recall, auc\_roc\_score, f1 score. Comparison between the different models(final) on the basis of above table values. After comparison which model suits the best for the problem in hand on the basis of different measures. Comment on the final model.

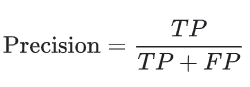
Sol 2.4

**Table containing all the values of accuracies, precision, recall, auc\_roc\_score, f1 score for both the models:**

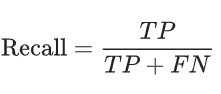
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model type** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **ROC AUC score** |
| Decision tree |  |  |  |  |  |
| Train Data | 80% | 70% | 64% | 67% | 84% |
| Test Data | 78% | 65% | 57% | 61% | 83% |
| Random forest |  |  |  |  |  |
| Train Data | 82% | 75% | 63% | 68% | 86% |
| Test Data | 77% | 66% | 52% | 58% | 81% |

**True Positive** **(TP)** — model correctly predicts the positive class (prediction and actual both are positive). In the above example, **10 people** who have tumors are predicted positively by the model.  
**True Negative (TN)** — model correctly predicts the negative class (prediction and actual both are negative). In the above example, **60 people** who don’t have tumors are predicted negatively by the model.  
**False Positive (FP)** — model gives the wrong prediction of the negative class (predicted-positive, actual-negative). In the above example, **22 people** are predicted as positive of having a tumor, although they don’t have a tumor. FP is also called a **TYPE I** error.  
**False Negative (FN)** — model wrongly predicts the positive class (predicted-negative, actual-positive). In the above example, **8 people** who have tumors are predicted as negative. FN is also called a **TYPE II** error.

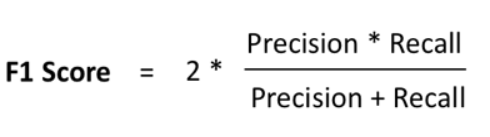
Precesion : Out of all the positive predicted, what percentage is truly positive. Precision is how good the model is at predicting a specific category.



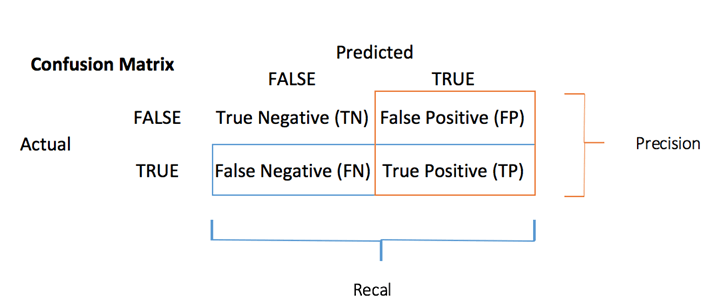
Recall: Out of the total positive, what percentage are predicted positive. It is the same as TPR (true positive rate).Recall tells you how many times the model was able to detect a specific category.

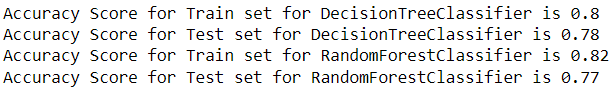


F1 score: a single metric that combines recall and precision using the harmonic mean.  it is not possible to maximize both these metrics at the same time, as one comes at the cost of another. For problems where both precision and recall are important, one can select a model which maximizes this F-1 score.



* Confusion matrix, precision, recall, and F1 score provides better insights into the prediction as compared to accuracy performance metrics.
* We do not want to **miss any claimed insurence**. Therefore, we **want False-Negative to be as low as possible.** In these situations, we can compromise with the low precision, but recall should be high.

**Comparing Accuracies from all the models for Train and Test Sets**



2.5 Based on your analysis and working on the business problem, detail out appropriate insights and recommendations to help the management solve the business objective. There should be at least 3-4 Recommendations and insights in total. Recommendations should be easily understandable and business specific, students should not give any technical suggestions. Full marks should only be allotted if the recommendations are correct and business specific.

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 and

compare the models' performances in train and test sets.

In the case of regression, instead of determining the most populous vote the random forest will average the results of each decision tree. Because random forests utilize the results of multiple learners (decisions trees), random forests are a type of ensemble machine learning algorithm. Ensemble learning methods reduce variance and improve performance over their constituent learning models.