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.

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it.

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

2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model

2.4 Final Model: Compare all the model 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:

* Target: Claim Status (Claimed)
* Code of tour firm (Agency\_Code)
* Type of tour insurance firms (Type)
* Distribution channel of tour insurance agencies (Channel)
* Name of the tour insurance products (Product)
* Duration of the tour (Duration)
* Destination of the tour (Destination)
* Amount of sales of tour insurance policies (Sales)
* The commission received for tour insurance firm (Commission)
* Age of insured (Age)

**2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it.**

Here we have to perform descriptive statistics and basic EDA.

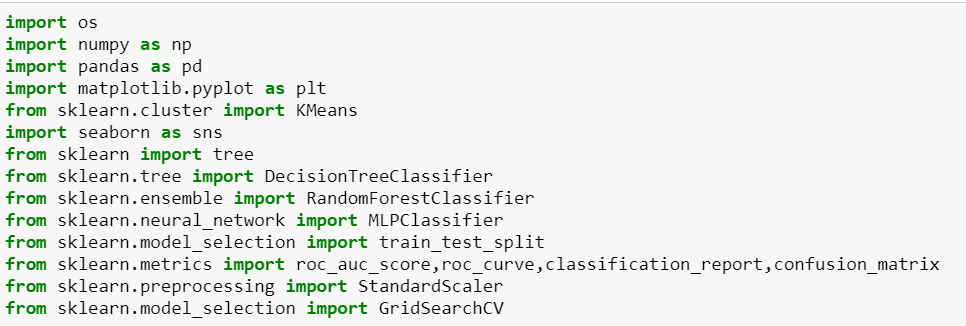
Firstly, we will import all the libraries we will be required further.

To predict the claimed status, Yes/No, we are going to use Three Algorithms and importing the libraries mentioned,

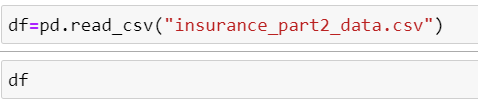
1.CART-Decision Tree (*import DecisionTreeClassifier from sklearn.tree*)

2.Random Forest (*RandomForestClassifier from sklearn.ensemble*)

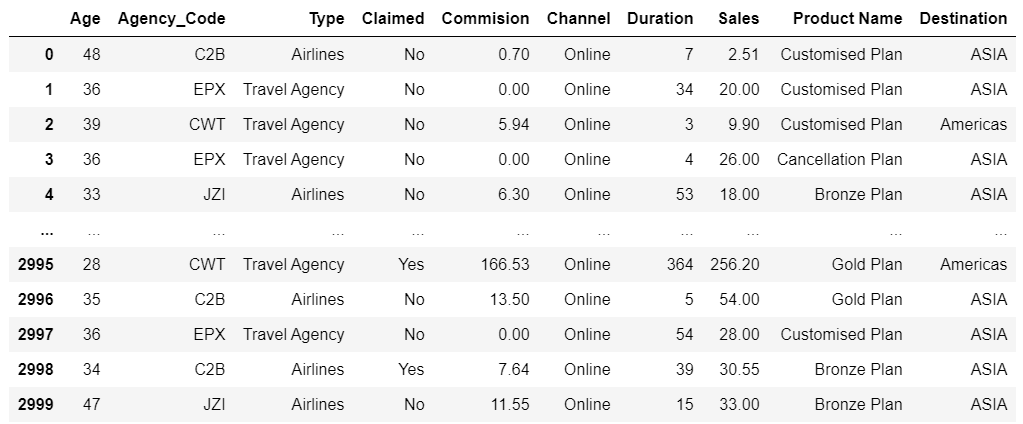
3. Artificial Neural Network (*MLPClassifier from sklearn.neural\_network)*



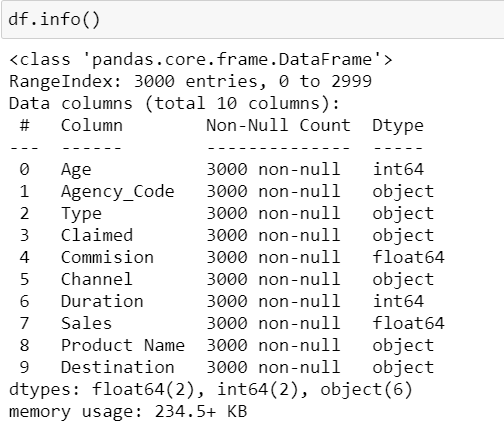
Read the Dataset and store it in dataframe format.



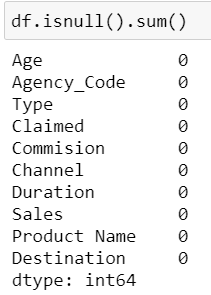
The output will be like this,



Again this code, will give us idea that there are 10 columns, having 3000 entries and No null values can be seen. However, Agency\_Code,Type , Claimed , Channel, Product name, Destination are having type as Object, we have to convert this categorical data into Numerical one for further processing,

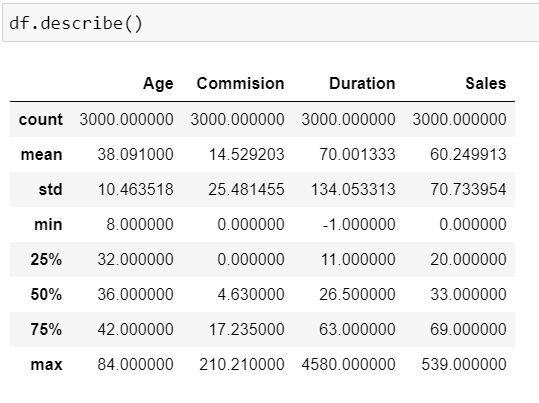


Check for Null Value:



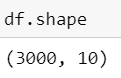
No Null value in data can be seen.

Lets check the data distribution for Continuous Variables:



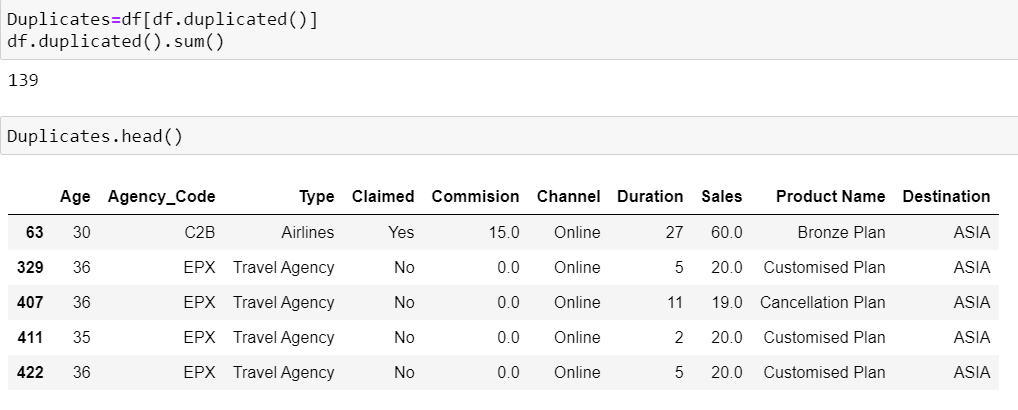
One thing to notice, Duration minimum value is -1. It cannot be negative,and mostly mean and Median are having different values, so data must be not Normally distributed.

Check Shape of dataset:



3000 rows and 10 Columns are present in dataset.

Check For Duplicates:

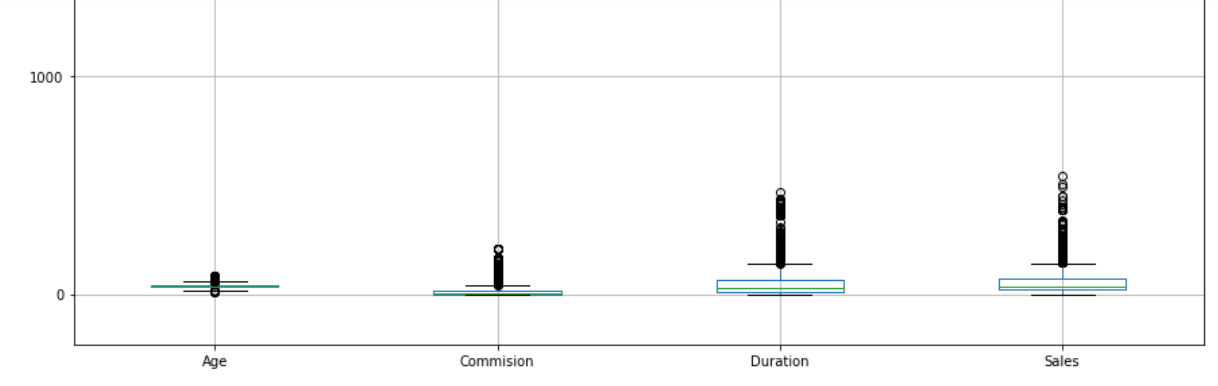


We can see there are 139 duplicates in 3000 records, but Insurance Company can provide Same Product to Different customer in Same Demographics and Via same Agency. And No separate Customer ID is given, so that we are not sure that these are same Customers or different customers.

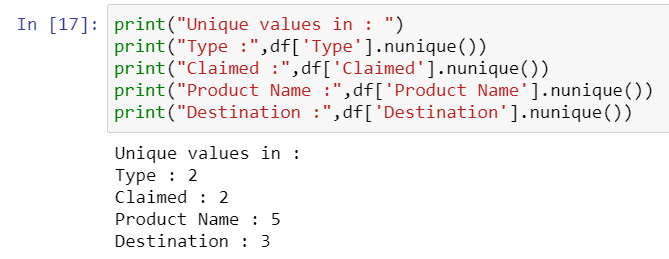
So no need to Process this duplicate value, we can keep it as it is.

To get the Idea of Outliers, we can plot a boxplot of Continuous Variables:

There are valid outliers, so no need to do treatment.



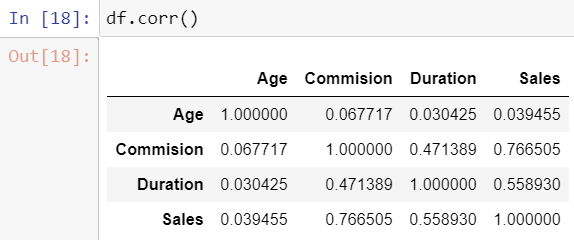
Check the unique values in Categorical Variables:



Maximum we are having 5 unique values and Minimum we have 2.

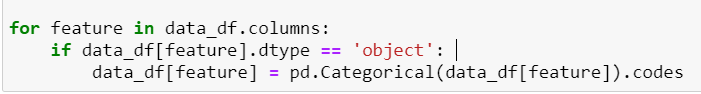
So we can assign codes to it.

We can plot Correlation plot to get the idea of correlation between variables:

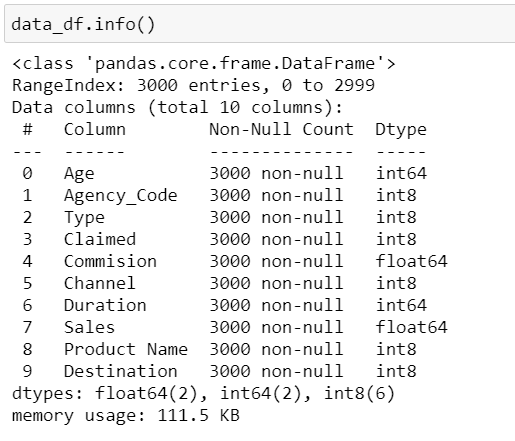


Convert Object data type into int/float by assigning codes:

Decision tree in Python can take only numerical / categorical colums. It cannot take string / object types. The following code loops through each column and checks if the column type is object then converts those columns into categorical with each distinct value becoming a category or code.

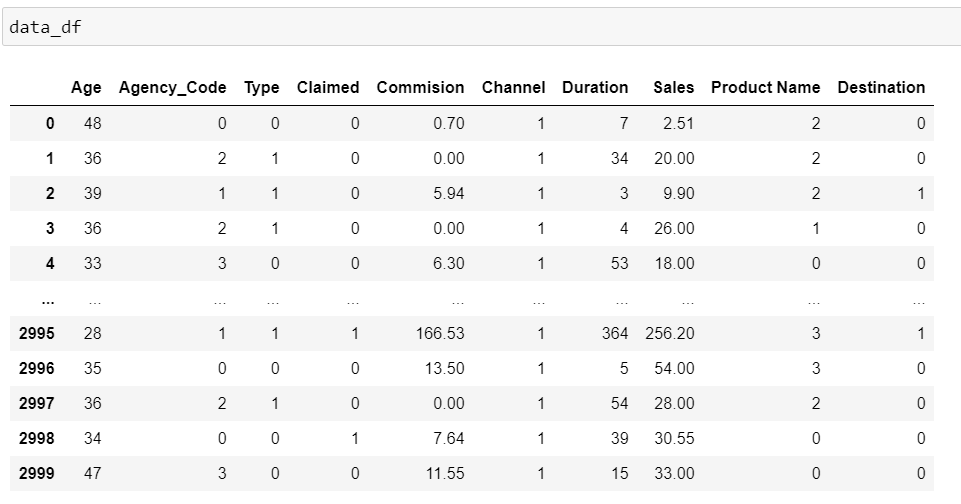


We can see all the values are converted into float datatype:

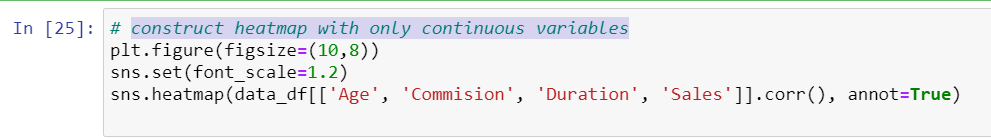


Now we can apply this data to Decision Tree.

We can see the codes are assigned to Categorical Variables,



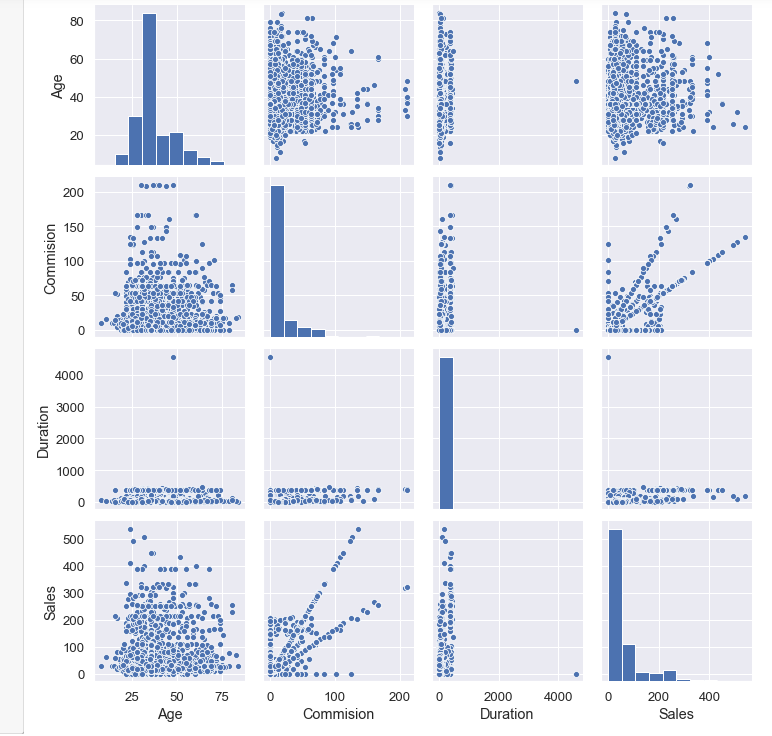
Lets construct heatmap with only continuous variables,



We can get the output as Heatmap,Here we can say the variables are strongly correlated if the value is more the 0.80 or lower negative value.

Lets do the Multivariate Analysis of the Featurtes by plotting pairplot for Continuous Variables:





So we can infer for this Exploratory data Analysis,

Total 10 variables in the dataset

Categorical Variables: Agency\_Code,Type, Claimed, Channel,Product Name,Destination Numerical Variables : Age, Commision, Duration, Sales are numeric variable

Total 3000 records, no missing values 9 independant variable and one target variable : "Clamied"

There are Ouliers in all 4 Numerical variables but One variable is quite high in Duration: 4580. we will think on it to remove this or impute it.

Minimum duration is -1, and Duration cannot be negative, This seems to be False entry. 139 rows are duplicated out of 3000 rows, but this can be entries for different customer and coincidences.so rows removal not required.

It seems that data need to be scaled for ANN as difference in 0%, 25% , 50%, 100%, Median and Max have large gap in Commision and Sales.

We have maximum 5 values which are unique or less than that so we can perform encoding and convert it into Data type Integer from Object

Correlation Heatmap shows that Sales, Commission and Duration are Interconnected.

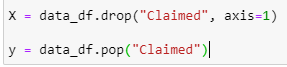
Duration and Commision shows stronger correlation.

Sales and Commission shows stronger correlation.

Sales and Duration shows stronger correlation.

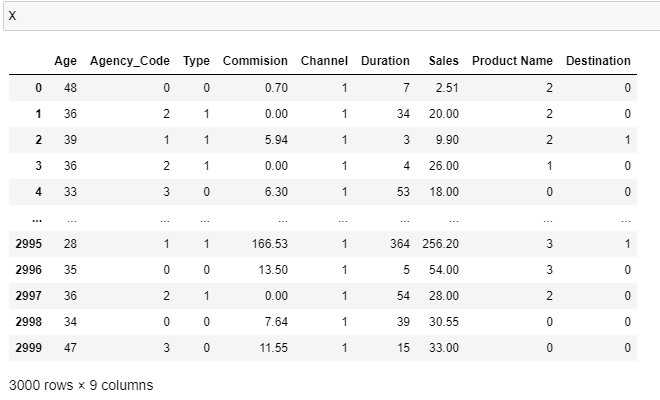
##### ***2.2 Data Split: Split the data into test and train, build classification model CART, Random Forest, Artificial Neural Network***

plit the data into two parts, Dependent Variables and Independent Variables.

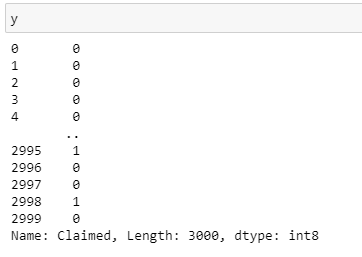


X is the Data without Claim column.

y is the part, Claimed is the only column.

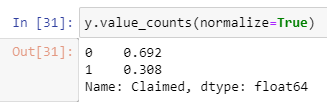


Y is the only one Column

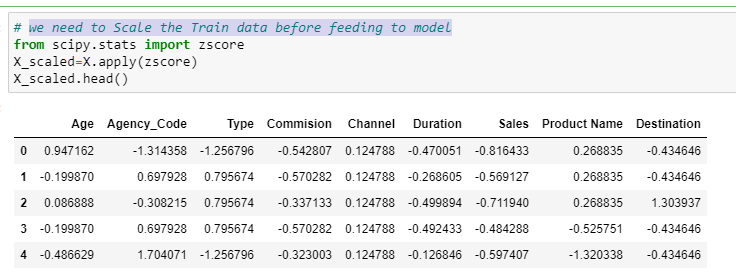


Target Variable y : It is Status Claimed is 30% and Status not claimed is 70%. So data is little bit balanced.

Not the ideal one but still can be worked upon.



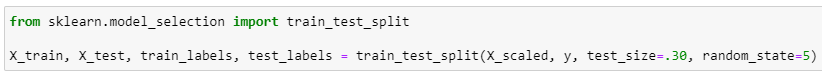
Lets Scale the data to bring all variables on same data,we need to Scale the Train data before feeding to model by using z score scaling.



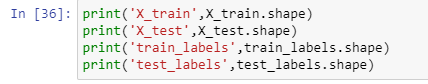
##### *Splitting data into training and test set .*

Train data: 70%

Test Data : 30%



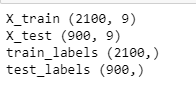
##### *Checking the dimensions of the training and test data*



Output can be seen:

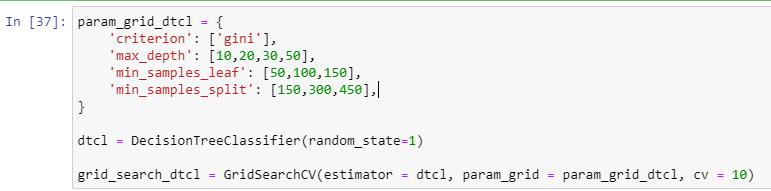
Train data has Dimension like 2100 rows out of 3000.

Test data has Dimension of 900 rows out of 3000.

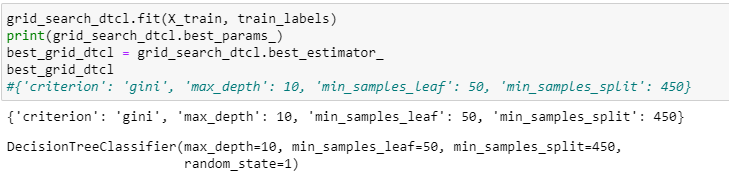


##### ***Building a Decision Tree Classifier :***

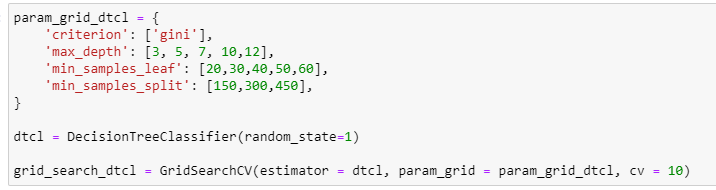
We can use DecisionTreeClassifier , Cross validation technique as grid search

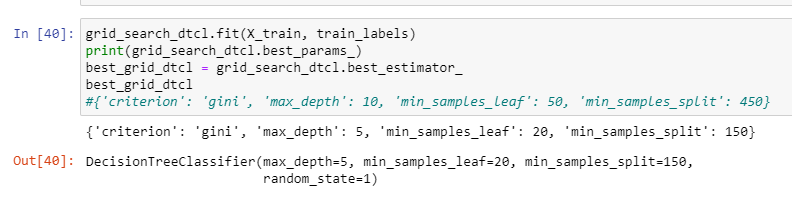


Fit the Model on Trained data:

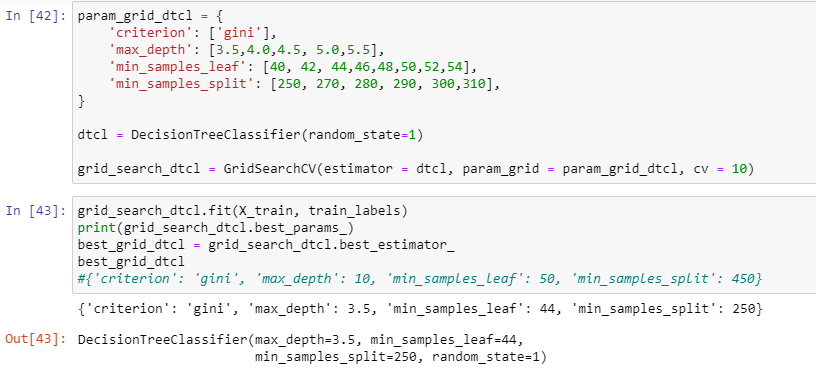


We can vary the inputs in Techniques to get the idea of best Decision Tree with best suited parameters





Again We can vary the inputs in Techniques to get the idea of best Decision Tree with best suited parameters



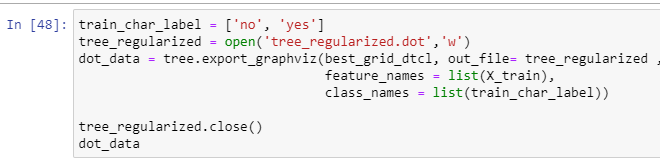
Again in similar way we will vary the inputs:



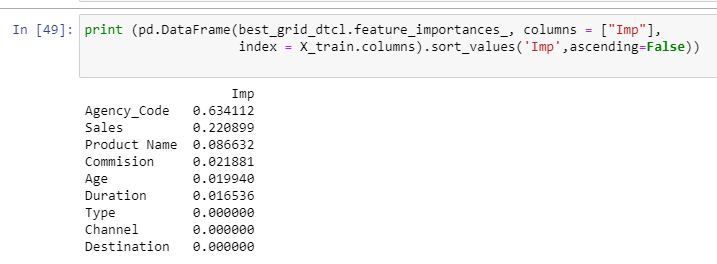
Here we can note that max depth, min sample leaf, split value are different and best grid decision tree Classifier.

Generate the Tree:

The tree can trained on Yes and No, Target variable.



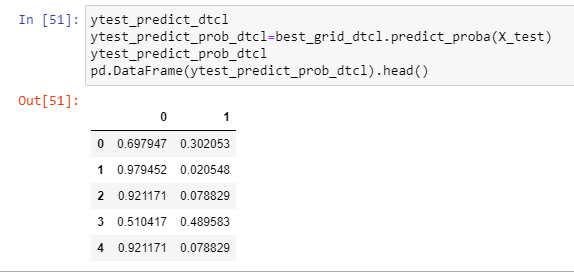
We can get the feature Importances as shown below in sorted way.



##### *Predicting on Training and Test dataset*

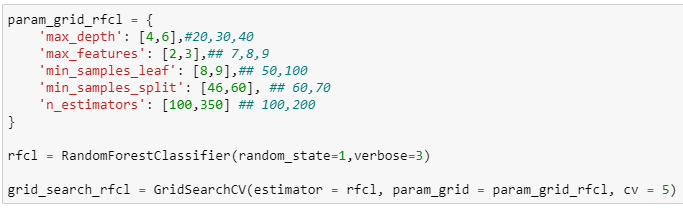


##### *Getting the Predicted Classes and Probabilities*



### **Building a Random Forest Classifier**

We can build a Random Forest Classifier, Create Model

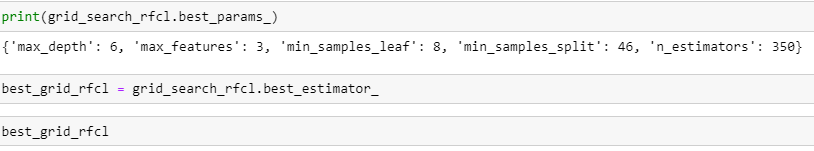


Verbose =3 is set because we can see the processing of RF.

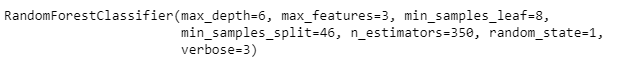
Fit the Model-Train Data by using RF Classifier



To get the best Parameter we can run below set of code:



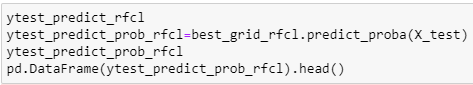
Output Can be:



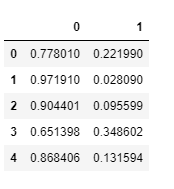
##### *Predicting the Training and Testing data*[*¶*](http://127.0.0.1:8888/notebooks/Final%20Assignment%20Q2%20Data%20Mining.ipynb#Predicting-the-Training-and-Testing-data)



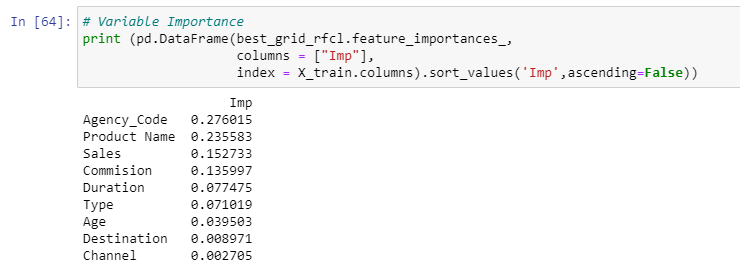
##### *Getting the Predicted Classes and Probs*



Output Probabilities can be seen below



##### *Variable Importance via RF*



In the above output we can clearly see Agency code has Highest importance followed by Product Name and Sales, Commission.

### **ARTIFICIAL NEURAL NETWORK**

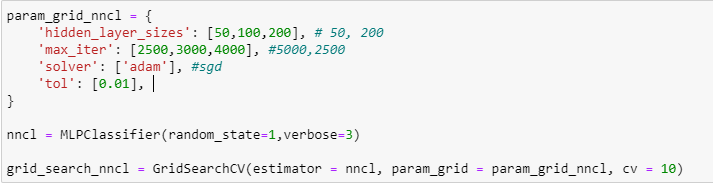
Create the Model:

Hidden layer Size is 50,100,200.

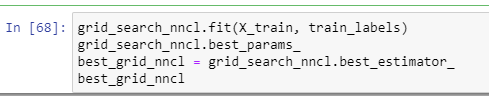
Iteration varies from 2500 to 4000.

Tolerance value is 0.01

Cross validation value is 10, This is to cross check and gives better Accuracy.



Fit the Model on Train Data:



### Predicting the Training and Testing data

We are predicting the Target variable



#### Getting the Predicted Classes and Probs

#### 

### **2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC\_AUC score for each model**

#### CART - AUC and ROC for the training data

#### 

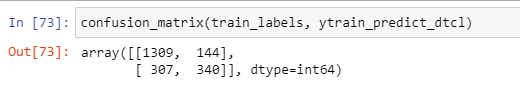
##### ***CART -AUC and ROC for the test data***

#### 

For Train data Area under Curve is 0.82 and for Test data it is 0.80.

##### ***CART Confusion Matrix and Classification Report for the training data***

Confusion Matrix:



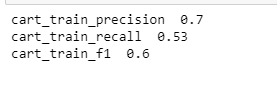
Train Data Accuracy :



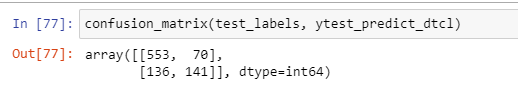
**Classification Report:**



So in a nutshell, This is the Required Values out of this report,



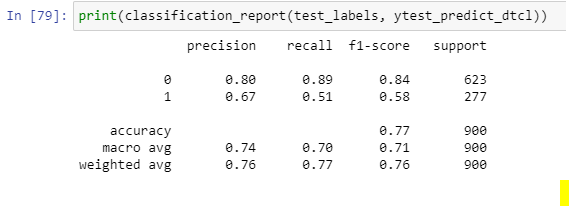
##### ***CART Confusion Matrix and Classification Report for the testing data***



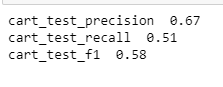
**Accuracy on Test Data:**



**Classification Report for Tree on test data:**



**Performance Measures on Test Data:**



CART conclusion:

**Train Data :**

* AUC: 82%
* Accuracy: 79%
* Precision: 70%
* f1-Score: 60%

**Test Data:**

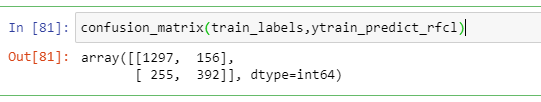
* AUC: 80%
* Accuracy: 77%
* Precision: 80%
* f1-Score: 84%

Training and Test set results are almost similar, and with the overall measures high, the model is a good model.

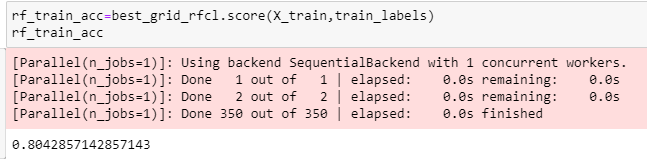
Change is the most important variable for predicting diabetes

##### ***Random Forest Model Performance Evaluation on Training data***

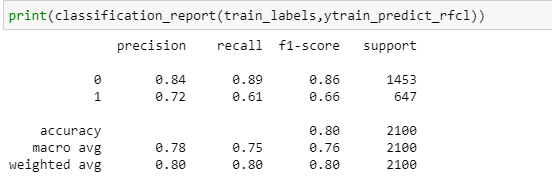
Confusion matrix on Random Forest Model



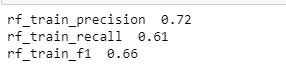
Score on Train data:



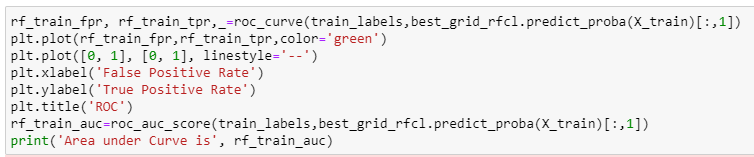
**Classification Report for Random Forest :**

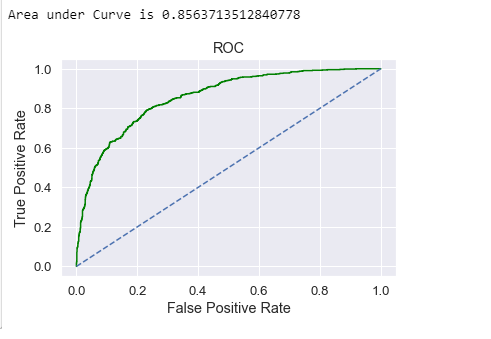


**Performance Measures on Random Forest on train data:**



Plot ROC curve for Train Data:

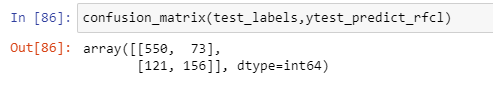




Area under Curve is almost 85%.

##### ***RF Model Performance Evaluation on Test data***

**Confusion Matrix:**

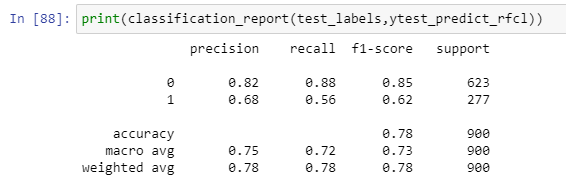


**Accuracy Score on Test Data:**

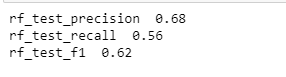




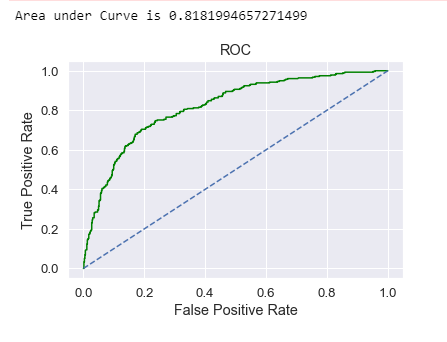
**Classification Report :**



**Total Scores on Test Data:**



**ROC Curve for Test Data :**



Random Forest Conclusion

Train Data:

AUC: 86%

Accuracy: 80%

Precision: 72%

f1-Score: 66%

Test Data:

AUC: 82%

Accuracy: 78%

Precision: 68%

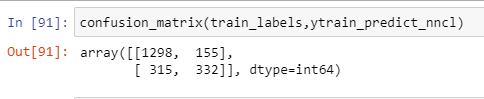
f1-Score: 62

Training and Test set results are almost similar, and with the overall measures high, the model is a good model.

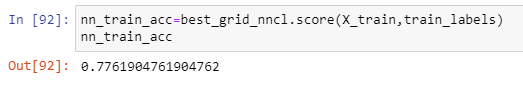
Change is again the most important variable for predicting diabetes

##### ***ANN Model Performance Evaluation on Training data***

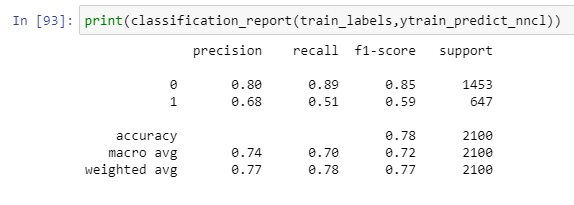
**Confusion Matrix:**



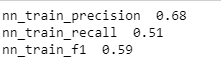
**Accuracy on Train data:**



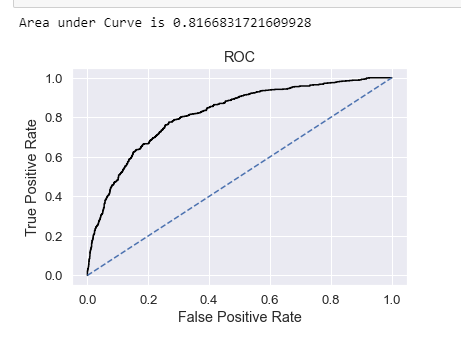
**Classification Report on Train Data:**



**Overall Parameters:**

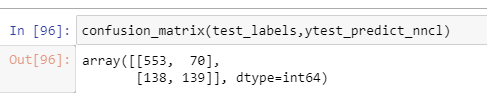


**ROC curve for Train Data for ANN model:**

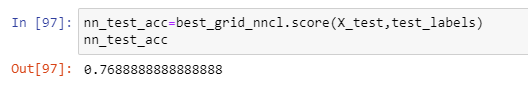


##### ***ANN Model Performance Evaluation on Test data***

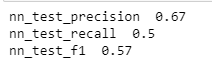
**Confusion Matrix on Test Data:**



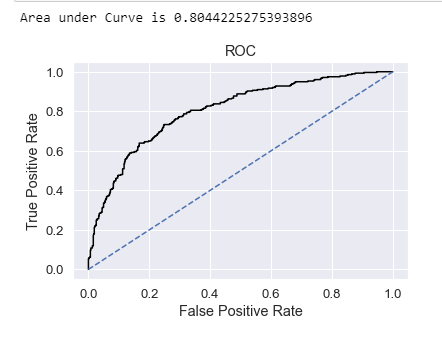
**Accuracy:**



**Overall Parameters:**



**Area under ROC curve is 80% for Test data:**



Train Data:

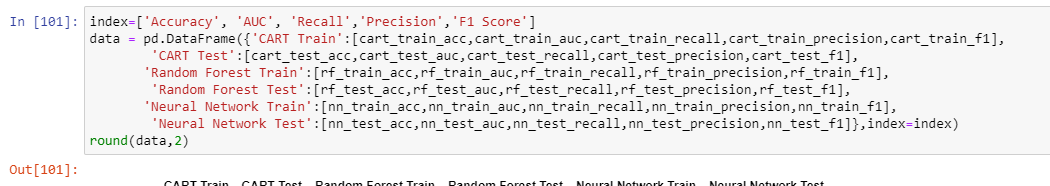
* AUC: 82%
* Accuracy: 78%
* Precision: 68%
* f1-Score: 59

Test Data:

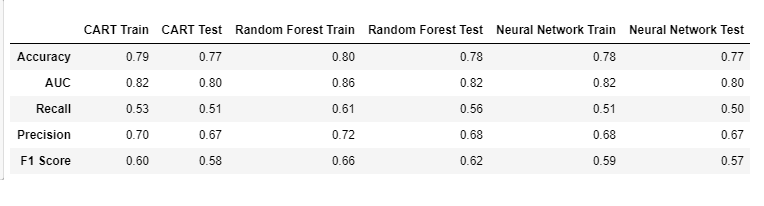
* AUC: 80%
* Accuracy: 77%
* Precision: 67%
* f1-Score: 57%

Training and Test set results are almost similar, and with the overall measures high, the model is a good model.

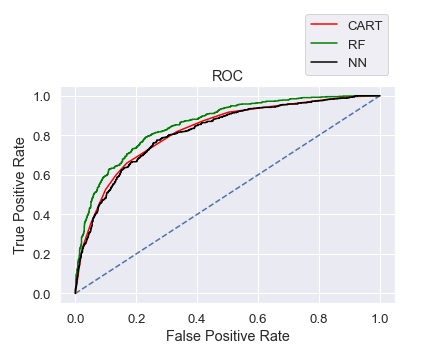
### **2.4 Final Model: Compare all the model and write an inference which model is best/optimized.**



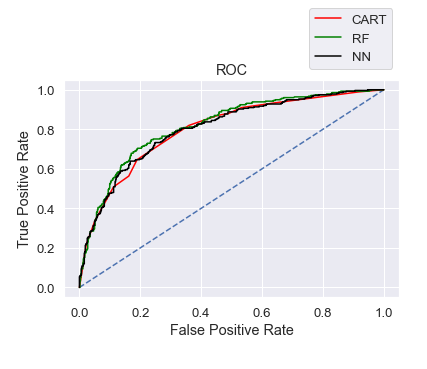
Output:



##### ***ROC Curve for the 3 models on the Training data***



##### ***ROC Curve for the 3 models on the Test data***



##### ***CONCLUSION :***

We Can Select Random Forest Model over other two, as it has better accuracy, precsion, recall, f1 score better than other two CART & NN.

### **2.5 Inference: Basis on these predictions, what are the business insights and recommendations**

We have seen the Imbalance Between Target Variable “Claimed” somewhat around 70-30%.

So if we take balanced target then the models can improve performance.

We need more details to predict accurate Target, so that model can learn better.

• Streamlining online experiences benefitted customers, leading to an increase in conversions, which subsequently raised profits.

• As per the data 90% of insurance is done by online channel.

• We have to dig deeper to all the offline business has a claimed associated, whether there queries are not resolved properly, or not a valid approval or It’s a Fraudulent because it is offline.

• JZI agency’s performance is low as compared to others, to improve its performance The organization must keep some Sales/Marketing strategies.

• Also based on the model we are getting 80%accuracy, so we need customer books airline tickets or plans, cross sell the insurance based on the claim data pattern.

• Other interesting fact is more sales happen via Agency than Airlines and the trend shows the claim are processed more at Airline. So we may need to deep dive into the process to understand the workflow and why?

• Reduce claims processing time

• Need to focus more on customer satisfaction

• Fraud detection need to be focused on.

• Claim recovery need to be scrutinized.