FLIP ROBO

MICRO CREDIT DEFAULTER PROJECT

Submitted By:-

RISHABH JOHRI

DATA SCIENCE-IN

ACKNOWLEDGEMENT:-

It is a great pleasure to express my gratitude to Team Flip Robo, for giving me the opportunity to work on a interesting project, which helped me in improving my knowledge, coding skills and my analysation skills.

Team Flip Robo also gave me opportunity to build PowerPoint Presentation and Project Report, which will help me to share steps taken while building the entire model. It has helped me in deciding about the future prospects of various Data Science fields. Now, I will explain the understanding of the project through this report.

INTRODUCTION OF PROJECT:-

A Microfinance Institution (MFI) is an organization that offers financial services to low income populations. MFS becomes very useful when targeting especially the unbanked poor families living in remote areas with not much sources of income. The Microfinance services (MFS) provided by MFI are Group Loans, Agricultural Loans, Individual Business Loans and so on.

Many microfinance institutions (MFI), experts and donors are supporting the idea of using mobile financial services (MFS) which they feel are more convenient and efficient, and cost saving, than the traditional high-touch model used since long for the purpose of delivering microfinance services. Though, the MFI industry is primarily focusing on low income families and are very useful in such areas, the implementation of MFS has been uneven with both significant challenges and successes.

Today, microfinance is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients.

We are working with one such client that is in Telecom Industry. They are a fixed wireless telecommunications network provider. They have launched various products and have developed its business and organization based on the budget operator model, offering better products at Lower Prices to all value conscious customers through a strategy of disruptive innovation that focuses on the subscriber.

Conceptual Background of the Domain Problem:-

Telecom Industries understand the importance of communication and how it affects a person's life, thus, focusing on providing their services and products to low income families and poor customers that can help them in the need of hour.

They are collaborating with an MFI with a aim to provide micro-credit on mobile balances to be paid back in 5 days. The Consumer is believed to be defaulter if he deviates from the path of paying back the loan amount within the time duration of 5 days.

The sample data is provided to us from our client database. It is hereby given to you for this exercise. In order to improve the selection of customers for the credit, the client wants some predictions that could help them in further investment and improvement in selection of customers.

We have to build a model which can be used to predict in terms of a probability for each loan transaction, whether the customer will be paying back the loan amount within 5 days of insurance of loan. In this case, Label '1' indicates that the loan has been paid which is Non- defaulter, while, Label '0' indicates that the loan has not been paid which is. defaulter.

MOTIVATION FOR THE PROBLEM UNDERTAKEN:-

For Modelling this dataset Micro Credit Defaulters with all given available independent variables. This model will then be used for management of how the customer is considered as defaulter or non-defaulter based on the independent variables. With the help of this prediction model, it will be decided accordingly and manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be prediction based insights to the management to understand whether the customer will be paying back the loan amount within 5 days of disbursement of loan or not.

Analytical Problem Framing:-

In this particular problem we have label as a target column and it has two classes Label '1' indicates that the loan has been paid6 i.e. Non- defaulter, while, Label '0' indicates that the loan has not been paid i.e. defaulter. So clearly it is a binary classification problem and I have to use all classification algorithms for building the model. There was no null values in the dataset. Also, I observed some columns where I found more than 90% zero values so I decided to drop those columns. If I would have kept that columns then it would have affected the model performance. To get better insight on the features I have used plotting like distribution plot, pie plot and reg plot.

With these plotting I was able to understand the relation between the features in better manner. Also, I found huge amount of outliers and high skewness levels present in the dataset so I have removed outliers and skewness using Yeo-Johnson method. I have used all the classification algorithms while building model then I tuned the best model and saved the best model. At last, I have predicted the label using saved model.

Data PreProcessing:-

As a first step I have imported required libraries and I have imported the dataset using csv file provided by the company.

Then we did all the statistical analysis like checking shape, nunique- for checking unique values in the dataset, value counts, info and description.

Then while looking into the value counts, I found some columns with more than 90% zero values which is creating skewness in the model and there are chances of getting model bias so I have dropped those columns with more than 90% zero values.

While checking for null values I found no null values.

I dropped Unnamed:0, msisdn and pcircle column as they were not required.

Next as a part of feature extraction, I converted the pdate column to day, month and year.

Also, I have dropped some columns when I tried to remove multicollinearity from the dataset using Variance Inflation Factor.

Since we have all numerical columns so, I used dist plot to see the distribution of each column data.

I have used reg plot for each pair that shows the relation between label and independent features. Also, we tried to observe whether the person pays back the loan within the date based on features.

In maximum features relation with Target, I observed Nondefaulter count is high as compared to defaulters.

Observations:-

We can see data imbalance in our target column which we will be rectified.

We can see Skewness towards the left, in columns aon, daily_dect30, daily_decr90, rental30, rental90, last_rech_date_ma, last_rech_amt_ma, cnt_ma_rech30, ft_ma_rech30, sunamnt_ma_rech30, medianamnt_ma_rech30, medianmarechprebal30, cnt_ma_rech90, fr_ma_rech90, sumamnt_ma_rech90, medianamnt_ma_rech90, medianmarechprebal90, cnt_loans30, amnt_loans30, maxamnt_loans30, cnt_loans90, amnt_loans90, payback30, payback90,.

maxamnt_loans90 column is skewed towards the right which denotes median is more than mean

Here we can see that there are only 2 values present in our Target column i.e. 1 and 0 and we can clearly see data imbalance in this column. we will balance the data using oversampling method.

Here we can see Correlation of our target column with other columns present in the dataset.

Here looking at plot of label and age on cellular network in days, we can say that defaults can happen even if the user is using services from Long time or not.

Here, correlation of target column label with columns daily_decr30, daily_decr90, rental30, rental90, last_rech_amt_ma, cnt_ma_rech30, sumamnt_ma_rech30, medianamnt_ma_rech30, cnt_ma_rech90, sumamnt_ma_rech90, medianamnt_ma_rech90, medianmarechprebel90, cnt_loans30, amnt_loans30, amnt_loans 90, we can say that There are less no. defaulters and most of the users paid there dues on time.

Here we can see that the user who took loans in last 30 days paid back most of loans on time.

Here we can observe that user who took loans in last 90 days was able to half of the loans on time while half users become defaulter as they couldn't pay back the loan on time.

Here we can observe that users paid most of the loans on time from total loans taken in 30 days, but they became defaulter in few loans.

Here we can see that users paid most of the loans on time from the total loans taken in last 90 days, but users became defaulters in paying back few loans.

Here we can observe that user was able to pay half of the amount of loan taken by him in last 30 days on time.

Here we can observe that users paid all small amount loans on time, but were able to pay half of large amount loan on time.

Here, we can observe multicollinearity in some columns which needs to be handled.

Here, we can observe that our target column label shares very little relation with all other columns present in the dataset.

Here, we can see that our target column label shares positive relation with all the columns except 3 columns.

Data Cleaning:-

In the dataset we did not found any null values, there were many 0 values, but I found huge amount of outliers and very high skewness.

To remove outliers we used zscore method. And for removing skewness I have used Yeo-Johnson method.

Then I have used Standard Scaler method to Scale the data.

As there is a issue of Multicollinearity in this, while Multivariate Analysis, we removed Multicollinearity using Variance Inflation Factor (VIF).

While performing Univariate Analysis, i found that our Target column was imbalanced, so i balanced the target column using SMOTE method in Oversampling. Then, after Preparing our data we have moved to model selection phase.

Model Selection:-

Since Label was our target column and it was a Categorical column, so this particular problem was a Classification problem. And we have used all Classification algorithms to build our model. We Tried multiple models and to avoid the confusion of overfitting we went through cross validation. Below are the list of Classification algorithms I have used in my project. By looking into the least difference between accuracy score and cross validation score, I found Random Forest Classifier as the best model. Other models which i have used were:

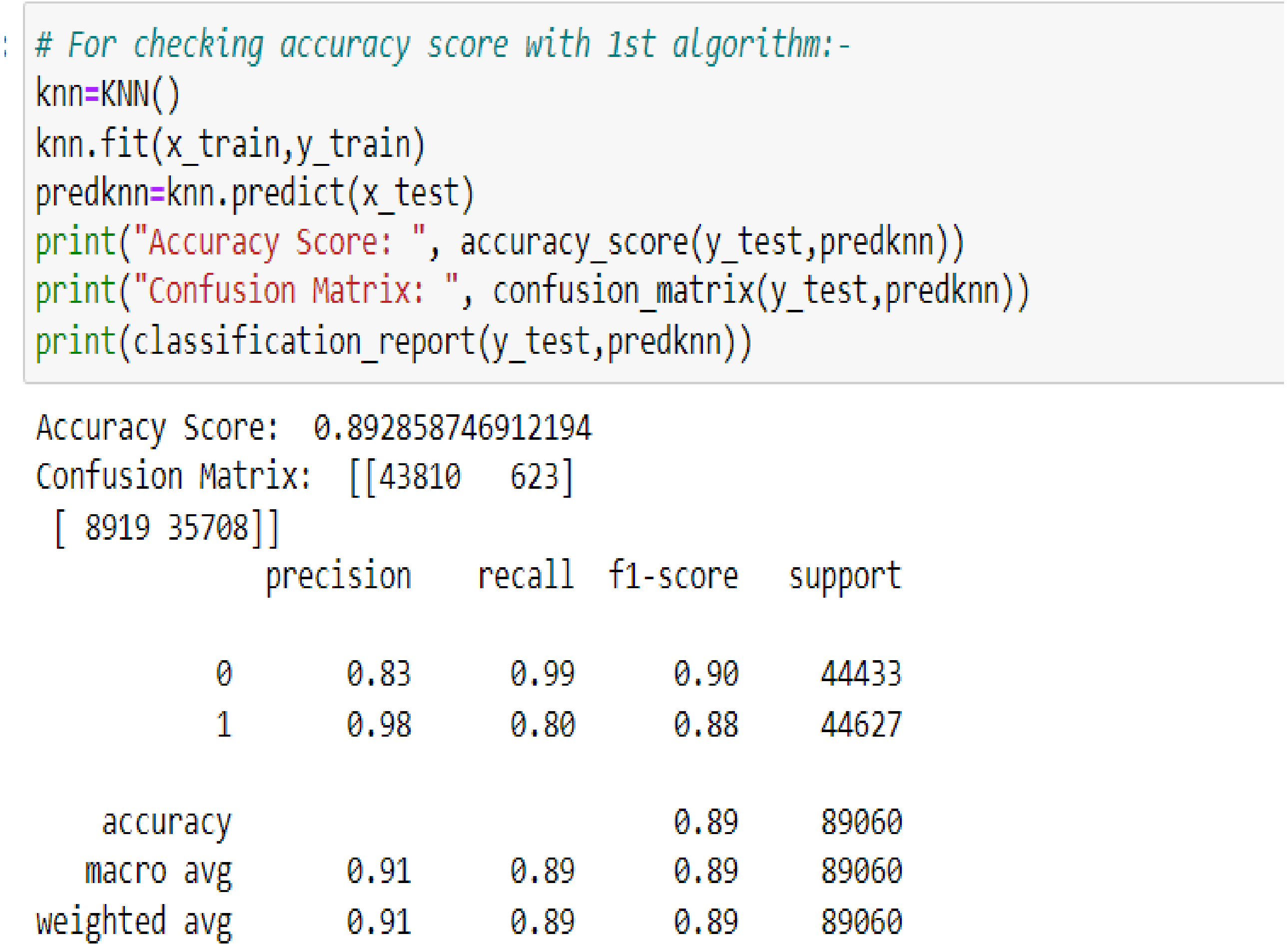
K Neighbors Classifier

Extra Trees Classifier

Gradient Boosting Classifier

Decision Tree Classifier.

Different Model Accuracy Scores:-

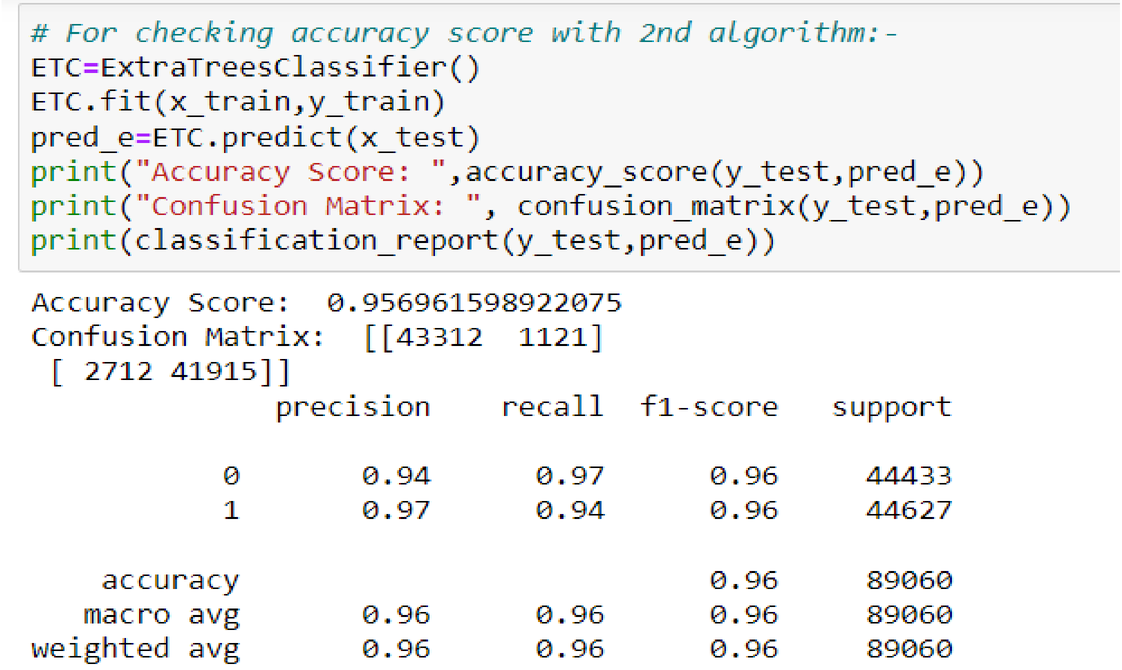


```
# For checking accuracy score with 1st algorithm:-
knn=KNN()
knn.fit(x_train,y_train)
predknn=knn.predict(x_test)
print("Accuracy Score: ", accuracy_score(y_test,predknn))
print("Confusion Matrix: ", confusion_matrix(y_test,predknn))
print(classification_report(y_test,predknn))
```

```
Accuracy Score:  0.892858746912194
Confusion Matrix:  [[43810   623]
 [ 8919 35708]]
              precision    recall  f1-score   support

           0       0.83      0.99      0.90     44433
           1       0.98      0.80      0.88     44627

    accuracy                           0.89     89060
   macro avg       0.91      0.89      0.89     89060
weighted avg       0.91      0.89      0.89     89060
```

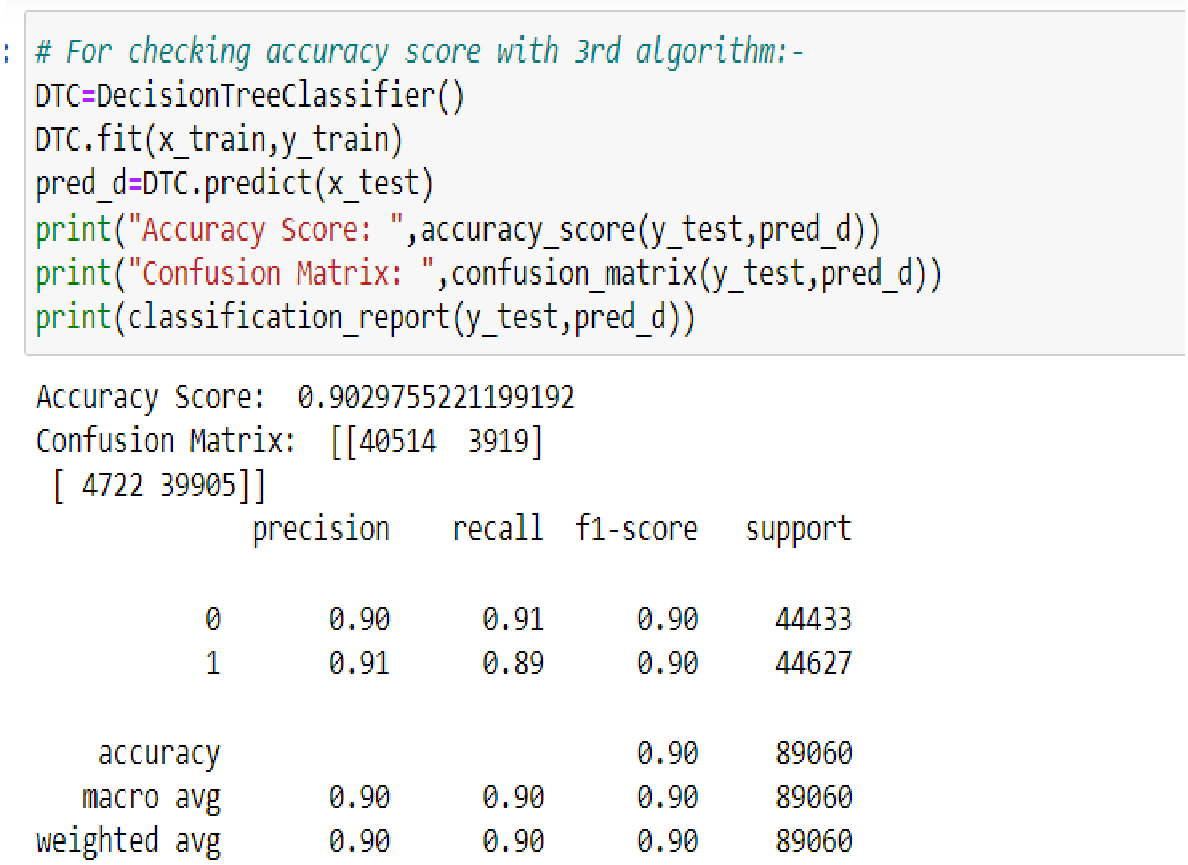


```
# For checking accuracy score with 2nd algorithm:-
ETC=ExtraTreesClassifier()
ETC.fit(x_train,y_train)
pred_e=ETC.predict(x_test)
print("Accuracy Score: ",accuracy_score(y_test,pred_e))
print("Confusion Matrix: ", confusion_matrix(y_test,pred_e))
print(classification_report(y_test,pred_e))
```

```
Accuracy Score:  0.956961598922075
Confusion Matrix:  [[43312  1121]
 [ 2712 41915]]
              precision    recall  f1-score   support

           0       0.94      0.97      0.96     44433
           1       0.97      0.94      0.96     44627

    accuracy                           0.96     89060
   macro avg       0.96      0.96      0.96     89060
weighted avg       0.96      0.96      0.96     89060
```

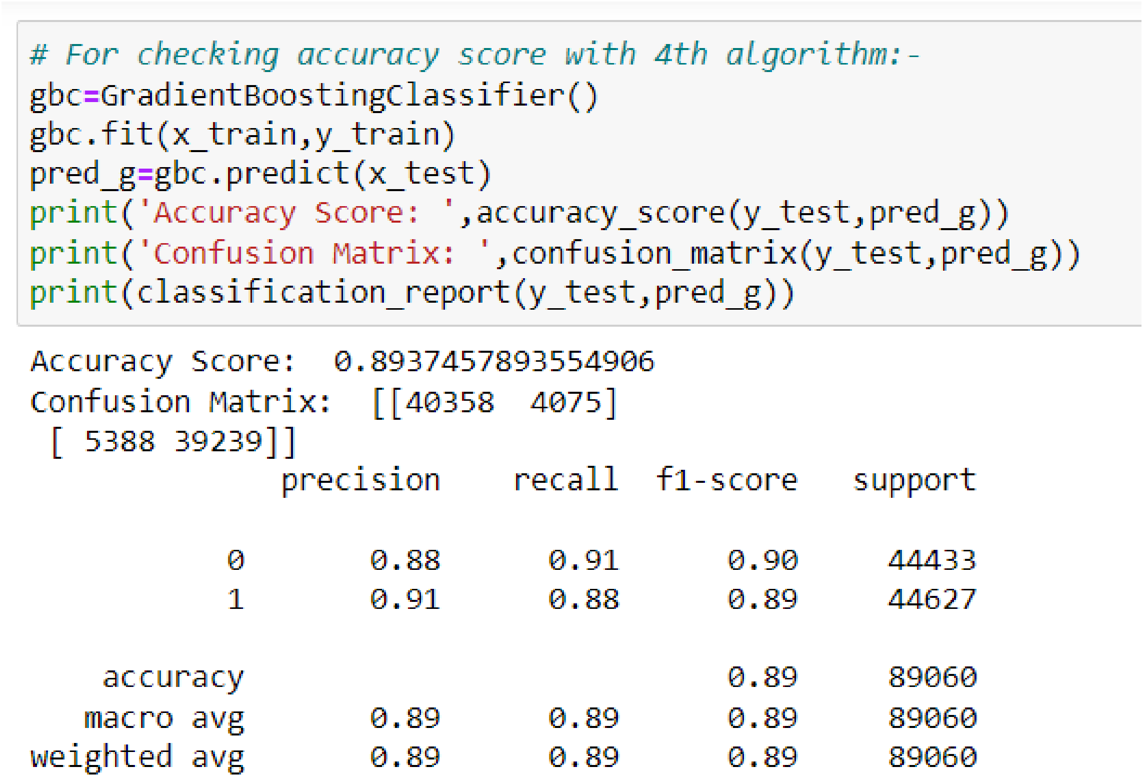


```
# For checking accuracy score with 3rd algorithm:-
DTC=DecisionTreeClassifier()
DTC.fit(x_train,y_train)
pred_d=DTC.predict(x_test)
print("Accuracy Score: ",accuracy_score(y_test,pred_d))
print("Confusion Matrix: ",confusion_matrix(y_test,pred_d))
print(classification_report(y_test,pred_d))
```

```
Accuracy Score:  0.9029755221199192
Confusion Matrix:  [[40514  3919]
 [ 4722 39905]]
              precision    recall  f1-score   support

           0       0.90      0.91      0.90     44433
           1       0.91      0.89      0.90     44627

    accuracy                           0.90     89060
   macro avg       0.90      0.90      0.90     89060
weighted avg       0.90      0.90      0.90     89060
```

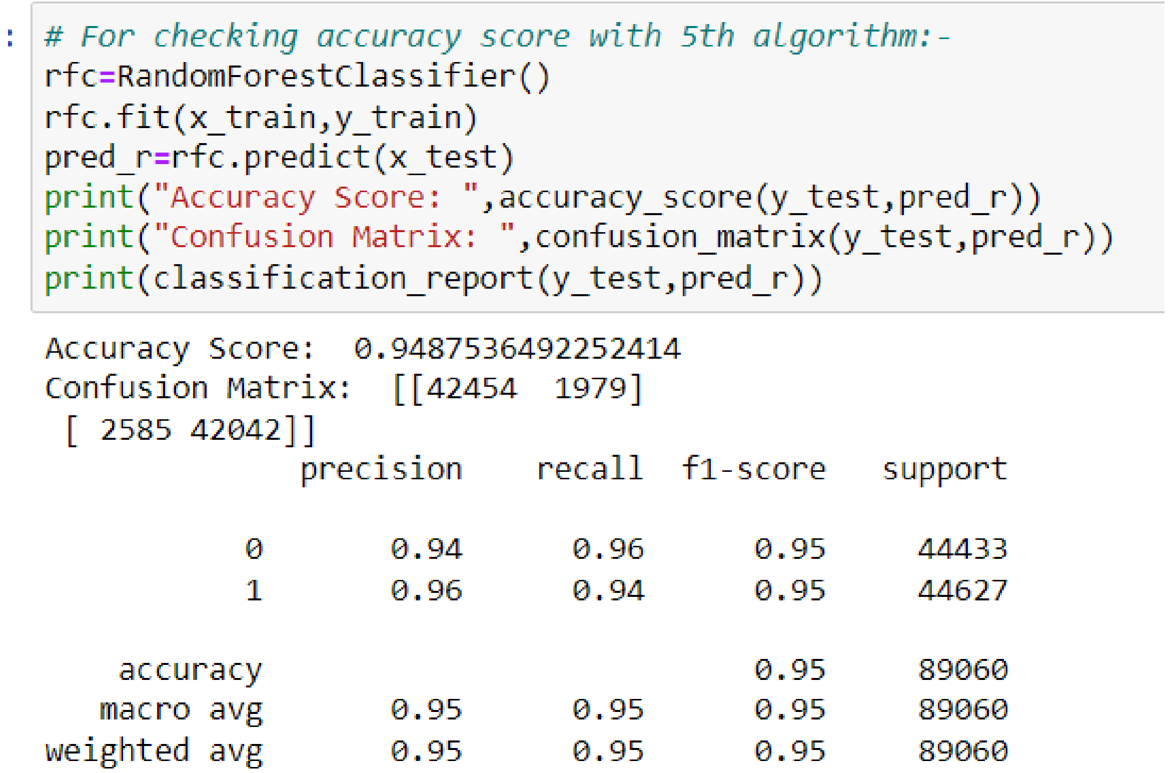


```
# For checking accuracy score with 4th algorithm:-
gbc=GradientBoostingClassifier()
gbc.fit(x_train,y_train)
pred_g=gbc.predict(x_test)
print('Accuracy Score: ',accuracy_score(y_test,pred_g))
print('Confusion Matrix: ',confusion_matrix(y_test,pred_g))
print(classification_report(y_test,pred_g))
```

```
Accuracy Score:  0.8937457893554906
Confusion Matrix:  [[40358  4075]
 [ 5388 39239]]
              precision    recall  f1-score   support

           0       0.88      0.91      0.90     44433
           1       0.91      0.88      0.89     44627

    accuracy                           0.89     89060
   macro avg       0.89      0.89      0.89     89060
weighted avg       0.89      0.89      0.89     89060
```



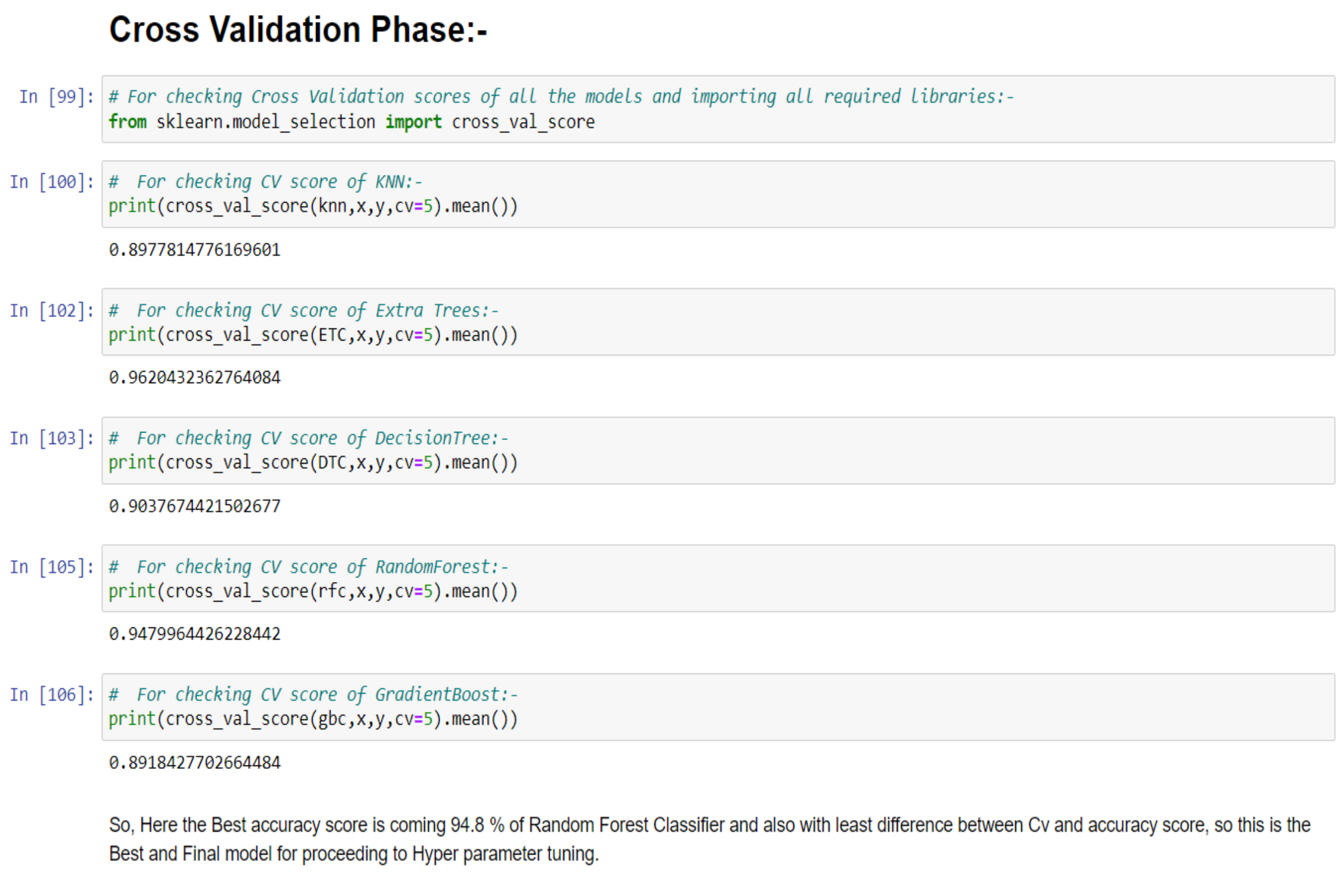
```
: # For checking accuracy score with 5th algorithm:-
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
pred_r=rfc.predict(x_test)
print("Accuracy Score: ",accuracy_score(y_test,pred_r))
print("Confusion Matrix: ",confusion_matrix(y_test,pred_r))
print(classification_report(y_test,pred_r))
```

```
Accuracy Score:  0.9487536492252414
Confusion Matrix:  [[42454  1979]
 [ 2585 42042]]
              precision    recall  f1-score   support

           0       0.94      0.96      0.95     44433
           1       0.96      0.94      0.95     44627

    accuracy                           0.95     89060
   macro avg       0.95      0.95      0.95     89060
weighted avg       0.95      0.95      0.95     89060
```

CROSS VALIDATION PHASES OF MODELS:-



Cross Validation Phase:-

In [99]:
```
# For checking Cross Validation scores of all the models and importing all required libraries:-
from sklearn.model_selection import cross_val_score
```

In [100]:
```
#  For checking CV score of KNN:-
print(cross_val_score(knn,x,y,cv=5).mean())
```

```
0.8977814776169601
```

In [102]:
```
#  For checking CV score of Extra Trees:-
print(cross_val_score(ETC,x,y,cv=5).mean())
```

```
0.9620432362764084
```

In [103]:
```
#  For checking CV score of DecisionTree:-
print(cross_val_score(DTC,x,y,cv=5).mean())
```

```
0.9037674421502677
```

In [105]:
```
#  For checking CV score of RandomForest:-
print(cross_val_score(rfc,x,y,cv=5).mean())
```

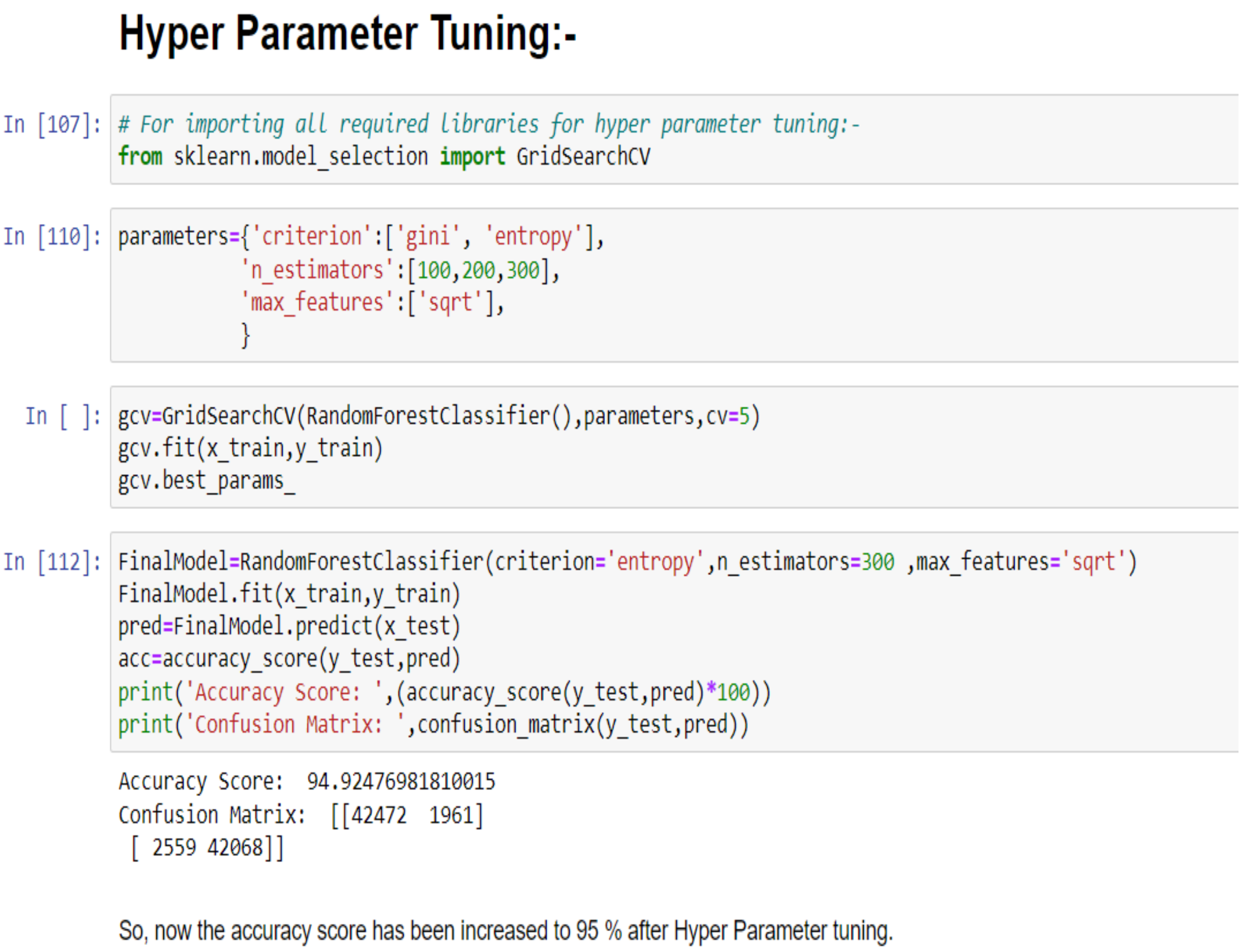
```
0.9479964426228442
```

In [106]:
```
#  For checking CV score of GradientBoost:-
print(cross_val_score(gbc,x,y,cv=5).mean())
```

```
0.8918427702664484
```

So, Here the Best accuracy score is coming 94.8 % of Random Forest Classifier and also with least difference between Cv and accuracy score, so this is the Best and Final model for proceeding to Hyper parameter tuning.

HYPER PARAMETER TUNING:-



Hyper Parameter Tuning:-

In [107]:
```
# For importing all required libraries for hyper parameter tuning:-
from sklearn.model_selection import GridSearchCV
```

In [110]:
```
parameters={'criterion':['gini', 'entropy'],
            'n_estimators':[100,200,300],
            'max_features':['sqrt'],
            }
```

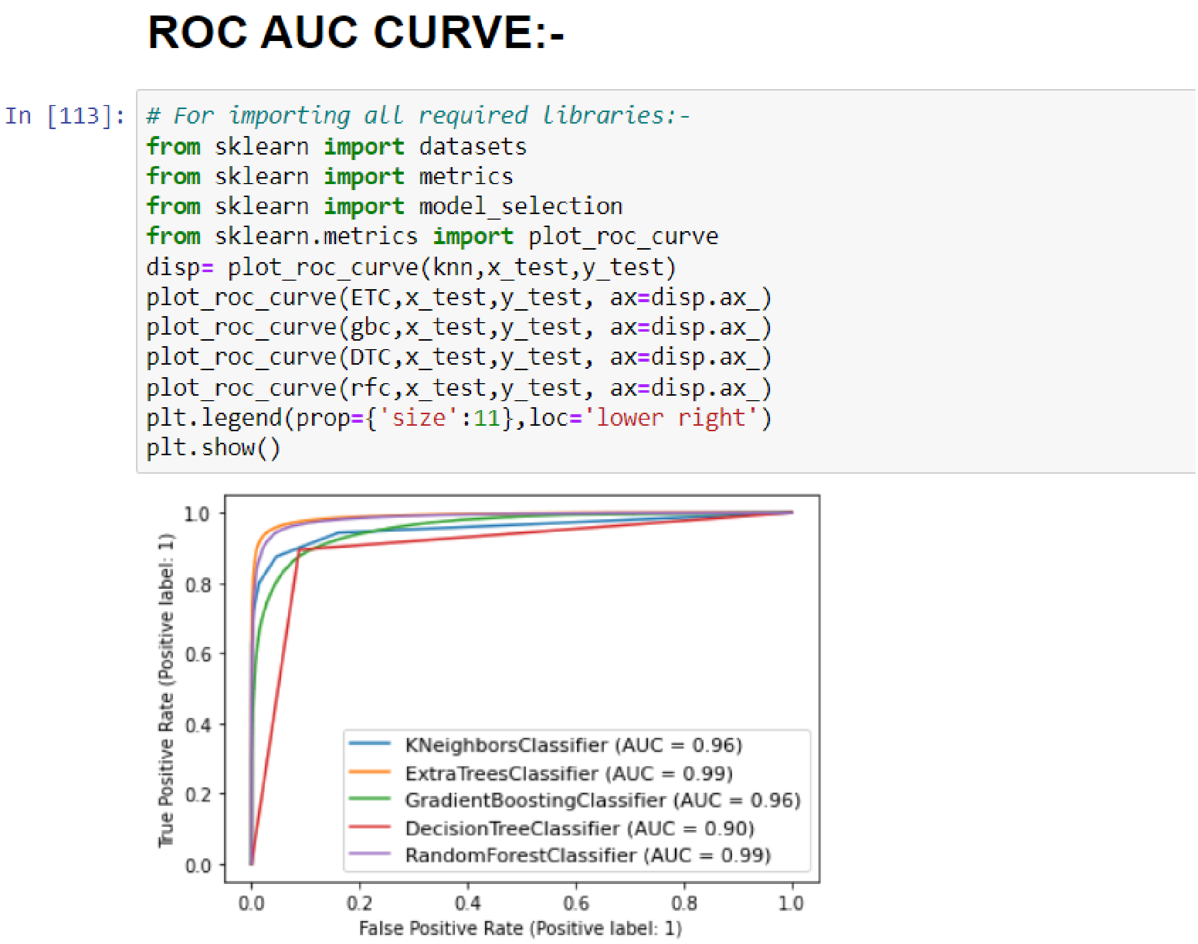
In [ ]:
```
gcv=GridSearchCV(RandomForestClassifier(),parameters,cv=5)
gcv.fit(x_train,y_train)
gcv.best_params_
```

In [112]:
```
FinalModel=RandomForestClassifier(criterion='entropy',n_estimators=300 ,max_features='sqrt')
FinalModel.fit(x_train,y_train)
pred=FinalModel.predict(x_test)
acc=accuracy_score(y_test,pred)
print('Accuracy Score: ',(accuracy_score(y_test,pred)*100))
print('Confusion Matrix: ',confusion_matrix(y_test,pred))
```

```
Accuracy Score:  94.92476981810015
Confusion Matrix:  [[42472  1961]
 [ 2559 42068]]
```

So, now the accuracy score has been increased to 95 % after Hyper Parameter tuning.

ROC/ AUC CURVE:-

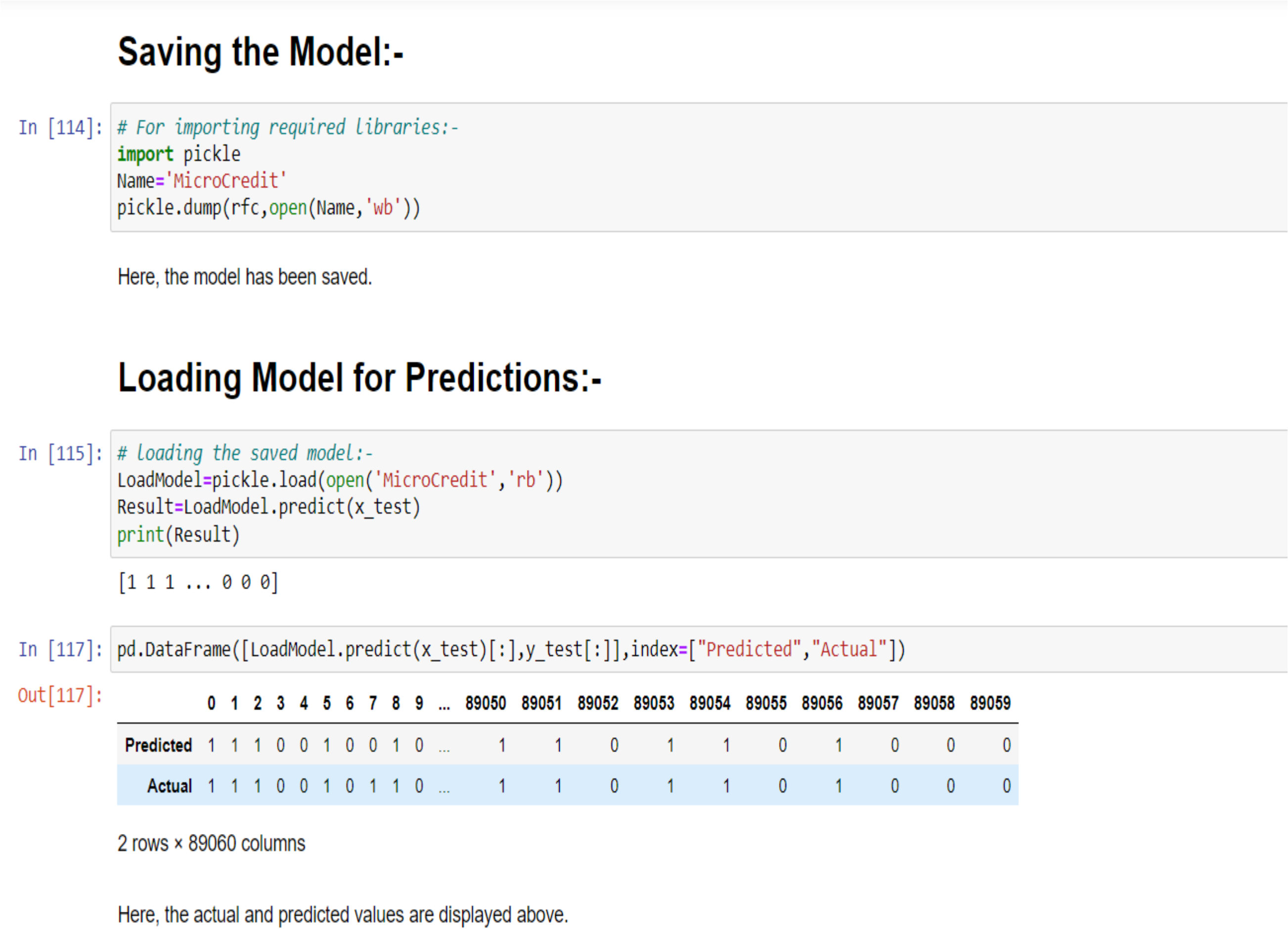


ROC AUC CURVE:-

```
In [113]: # For importing all required libraries:-
          from sklearn import datasets
          from sklearn import metrics
          from sklearn import model_selection
          from sklearn.metrics import plot_roc_curve
          disp= plot_roc_curve(knn,x_test,y_test)
          plot_roc_curve(ETC,x_test,y_test, ax=disp.ax_)
          plot_roc_curve(gbc,x_test,y_test, ax=disp.ax_)
          plot_roc_curve(DTC,x_test,y_test, ax=disp.ax_)
          plot_roc_curve(rfc,x_test,y_test, ax=disp.ax_)
          plt.legend(prop={'size':11},loc='lower right')
          plt.show()
```

True Positive Rate (Positive label: 1)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1)
KNeighborsClassifier (AUC = 0.96)
ExtraTreesClassifier (AUC = 0.99)
GradientBoostingClassifier (AUC = 0.96)
DecisionTreeClassifier (AUC = 0.90)
RandomForestClassifier (AUC = 0.99)

SAVING THE MODEL AND LOADING PREDICTIONS:-



Saving the Model:-

```
In [114]: # For importing required libraries:-
          import pickle
          Name='MicroCredit'
          pickle.dump(rfc,open(Name,'wb'))
```

Here, the model has been saved.

Loading Model for Predictions:-

```
In [115]: # loading the saved model:-
          LoadModel=pickle.load(open('MicroCredit','rb'))
          Result=LoadModel.predict(x_test)
          print(Result)

          [1 1 1 ... 0 0 0]
```

```
In [117]: pd.DataFrame([LoadModel.predict(x_test)[:],y_test[:]],index=["Predicted","Actual"])
Out[117]:
```

	0	1	2	3	4	5	6	7	8	9	...	89050	89051	89052	89053	89054	89055	89056	89057	89058	89059
Predicted	1	1	1	0	0	1	0	0	1	0	...	1	1	0	1	1	0	1	0	0	0
Actual	1	1	1	0	0	1	0	1	1	0	...	1	1	0	1	1	0	1	0	0	0

2 rows × 89060 columns

Here, the actual and predicted values are displayed above.

CONCLUSION:-

In this project report, I have used Machine Learning algorithms to predict the Micro-Credit Defaulters. We have used proper procedure to analyse the dataset and finding the correlation between the features.

Here we have selected the features which are correlated to each other and are independent in nature. Visualization helped us in understanding the data by graphical representation and it has made things easy for us to understand what data actually is.

Data cleaning is one of the most important steps to remove 0 values and columns which are having more than 90% 0 values.

Using these feature, we deployed 5 algorithms to find the best model and Hyper parameter tuning was done on the Best Model and we have achieved improvement of accuracy score.

Then we saved the best model and predicted the label. Our model performance was good when we saw the predicted and actual values were almost same it felt really good and feels good performance by our model.

To conclude, the Project Micro Credit Defaulter , We hope this study will move a small step ahead in providing some methodological and empirical contributions to crediting institutes, and presenting an alternative approach to the valuation of defaulters.