

# **NAME OF THE PROJECT**

[Micro Credit Project]

**SUBMITTED BY:**

Arpan Chaudhary

# **ACKNOWLEDGEMENT**

This project includes the reference of company, undertaking my internship is going on. Full credit of data source goes to company. Wherever I need, full support is provided by company. I have fully contributed from my side for completion this project. Otherwise, I did not take help from any other source for this project completion.

# INTRODUCTION

## **BUSINESS PROBLEM FRAMING:**

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.

## **Conceptual Background of the Domain Problem:**

To understanding this project, will work with the help of Python. Where it can be easy to understand the trend of data. Where we can describe the data of this project, visualize the data with the help of various tools of visualization. We can manipulate the data in various ways. We can apply various methods to understand the dataset.

## **Review of Literature:**

- First, we import important libraries for reading the dataset.
- Assign dataset to a variable.
- Checking for null values present in dataset or not.
- As seen by checking, no nulls are present in dataset.
- Now, check for datatype, in which form data is present.
- Some columns are present as object datatype, others are present as int datatype and float datatype.
- We have to encode these columns which are present as object datatype.
- Import encoder for encoding required columns.
- As seen a nominal column is present in the dataset, if we want we can delete it, because it will not affect more to dataset.
- Now, we will visualize the dataset.

- Import libraries for visualization.
- First plot, boxplot.
- As by boxplot, a lot of outliers are seen in dataset.
- We have to treat these outliers.
- Let's find quantiles and inter quantile range first, which will help to removing outliers.
- Now, by plotting distribution plot individually, start with removing outliers, but we will not even touch to target variable.
- After doing this whole process of removing outliers, a lot of outliers have removed from dataset.
- Now, will plot heatmap for checking the multicollinearity problem among columns.
- Some columns are shown having high relation, let's check them by plotting scatter plot.
- By scatter plot seems like, really having a good relation.
- This relation can impact our dataset, so we have to delete some columns.
- Deleting 5 columns, because their respective columns are already in dataset.
- Will not more harm to dataset after deleting these columns.
- Now our dataset is ready for further processing.
- We will split our dataset into two variables.
- First, with the help of counter we check our target variable, either balanced or imbalanced.

- A much difference in target variable is shown after observation.
- Let's make it balance with the help of SMOTE(a tool for proper balancing).
- After balancing the dataset, will do standardize our assigned data which is more important for model training.
  - After standardize the data, its time to do train test split.
  - Importing metrics for checking the accuracy and error of model.
  - Now its time to train our models.
  - Because our target variable is having categorical data, so will train our classification models on given dataset.
  - First model, Logistic Regression.
  - Second model, Decision Tree.
  - Third model, Random Forest.
  - Fourth model, SVM.

## **Motivation for the Problem Undertaken:**

Main motive to do this project is that these all observations done on this project and preparing Model for this project will help in micro finance sector. By this Model, will get a focused mindset whether to invest in this sector or not. How to do all things related to this project. What are main things we have to

more focus, which one we can ignore and all about things related to it.

# Analytical Problem Framing

## Mathematical/ Analytical Modeling of the Problem

- By plotting boxplot, outliers can be seen in given dataset.
- No nulls values are there in dataset, no more worry about nulls.
- The most critical situation was that to handle outliers because a lot of outliers was present in dataset.
- A huge data-loss happened after removing the outliers.
- But along these outliers was more risk to train our different models.
- Just did most simple and low risk attempts for removing outliers.
- Just taken help of quantiles and inter quantile range to remove the outliers.

## Data Sources and their formats:

- Data for this project has been given by company under which internship is going on.
- Data is given as zip file, csv format, comma separated values files can be opened as in a very simple command.



## **Data-Preprocessing Done:**

- In this process, some steps are taken.
- Standardize the data after splitting the dataset into two variables.
- Train-test-split is must for dataset.
- After these steps we can train our models.

## **Data Inputs- Logic- Output Relationships:**

- We can't pass our input data as same for further process, means for model training because our input data may be imbalanced, not in a proper way. The all-thing is that we can't pass input data as it is for model training.
- We have to treat with input data, cleaning the data, manipulating the data, standardizing the data, filling the null values if present, we have to check the statistics for dataset. We have to know the trend of dataset, using visualization techniques.
- We can only and only pass the data after cleaning, filling the null values, removing outliers, checking multicollinearity relation, standardize the data, and last after doing train-test-split.

# Model/s Development and Evaluation

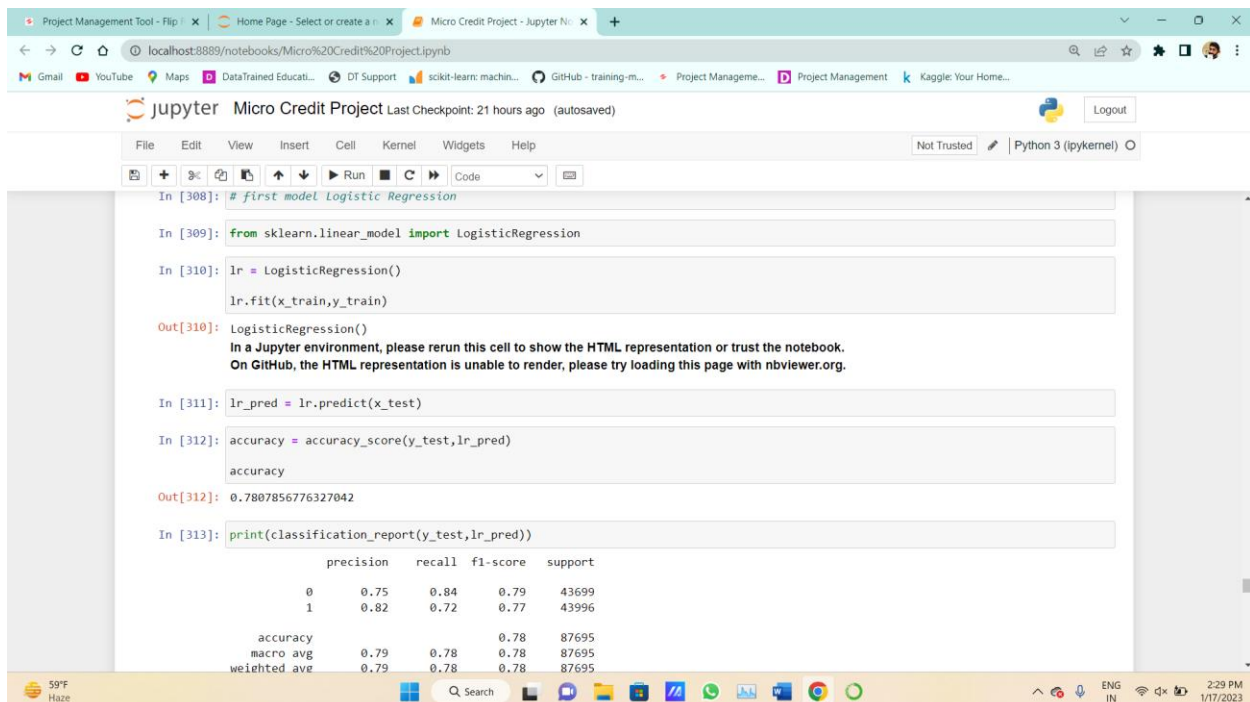
## Identification of possible problem-solving approaches (methods):

- Just read the dataset with the help of PANDAS and NUMPY.
- Assign the dataset to a variable.
- Check the statistics of dataset by help of “describe()” method.
- Check for null values.
- Check for outliers with the help of boxplot.
- Remove outliers with the help of quantiles and inter quantile range.
- Removed all required outliers.
- Again check the data trend with the help of distribution plot.
- Plot heatmap for checking relation among columns.
- Standardize the data with the help of StandardScaler().
- Train-test-split.
- Then, trained our models.
- Check error of models with the help of confusion matrix and classification report.
- Hyperparameter tuning for required model which we want to consider.

## Testing of Identified Approaches (Algorithms):

- First Algorithm --> Logistic Regression
- Second Algorithm --> Decision Tree
- Third Algorithm --> Random Forest
- Fourth Algorithm --> SVM (Support Vector Machine)
- Hyperparameter tuning for Random Forest
- Confusion matrix for checking error
- Classification report

## Run and evaluate selected models:



```
In [308]: # first model Logistic Regression

In [309]: from sklearn.linear_model import LogisticRegression

In [310]: lr = LogisticRegression()
          lr.fit(x_train,y_train)

Out[310]: LogisticRegression()
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [311]: lr_pred = lr.predict(x_test)

In [312]: accuracy = accuracy_score(y_test,lr_pred)
          accuracy

Out[312]: 0.7807856776327042

In [313]: print(classification_report(y_test,lr_pred))
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.75      | 0.84   | 0.79     | 43699   |
| 1            | 0.82      | 0.72   | 0.77     | 43996   |
| accuracy     |           |        | 0.78     | 87695   |
| macro avg    | 0.79      | 0.78   | 0.78     | 87695   |
| weighted avg | 0.79      | 0.78   | 0.78     | 87695   |

Project Management Tool - Flip x Home Page - Select or create a x Micro Credit Project - Jupyter No x +

localhost:8889/notebooks/Micro%20Credit%20Project.ipynb

Gmail YouTube Maps DataTrained Educati... DT Support scikit-learn: machin... GitHub - training-m... Project Managem... Project Management Kaggle: Your Home...

Jupyter Micro Credit Project Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

weighted avg 0.79 0.78 0.78 87695

```
In [314]: confusion_matrix(y_test,lr_pred)
Out[314]: array([[36879, 6820],
               [12404, 31592]], dtype=int64)

In [ ]:

In [315]: fpr,tpr,threshold = roc_curve(y_test,lr_pred)

In [316]: print('False Positive Rate =',fpr)
           print('True Positive Rate =',tpr)
           print('Threshold =',threshold)

False Positive Rate = [0.          0.15606764 1.          ]
True Positive Rate = [0.          0.71806528 1.          ]
Threshold = [2 1 0]

In [317]: plt.plot(fpr,tpr,color='green',label='ROC')
           plt.plot([0,1],[0,1],color='darkred',linestyle='--')
           plt.xlabel('False Positive Rate')
           plt.ylabel('True Positive Rate')
           plt.title('Reciever Operating Characteristic (ROC) Curve')
           plt.legend()
           plt.show()
```

59°F Haze

Project Management Tool - Flip x Home Page - Select or create a x Micro Credit Project - Jupyter No x +

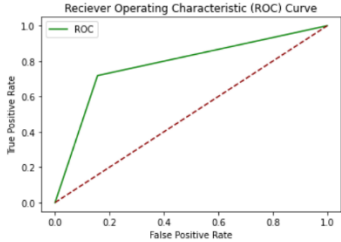
localhost:8889/notebooks/Micro%20Credit%20Project.ipynb

Gmail YouTube Maps DataTrained Educati... DT Support scikit-learn: machin... GitHub - training-m... Project Managem... Project Management Kaggle: Your Home...

Jupyter Micro Credit Project Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [317]: plt.plot(fpr,tpr,color='green',label='ROC')
           plt.plot([0,1],[0,1],color='darkred',linestyle='--')
           plt.xlabel('False Positive Rate')
           plt.ylabel('True Positive Rate')
           plt.title('Reciever Operating Characteristic (ROC) Curve')
           plt.legend()
           plt.show()
```



Reciever Operating Characteristic (ROC) Curve

```
In [318]: auc_score = roc_auc_score(y_test,lr_pred)
           auc_score

Out[318]: 0.7809988170465855
```

59°F Haze

Project Management Tool - Flip | Home Page - Select or create a | Micro Credit Project - Jupyter No | +

localhost:8889/notebooks/Micro%20Credit%20Project.ipynb

Gmail YouTube Maps DataTrained Educati... DT Support scikit-learn: machin... GitHub - training-m... Project Managem... Project Management Kaggle: Your Home...

Jupyter Micro Credit Project Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

In [319]: # second model Decision Tree

In [320]: from sklearn.tree import DecisionTreeClassifier

In [321]: dt = DecisionTreeClassifier()  
dt.fit(x\_train,y\_train)

Out[321]: DecisionTreeClassifier()  
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.  
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [322]: dt\_pred = dt.predict(x\_test)

In [323]: accuracy = accuracy\_score(y\_test,dt\_pred)  
accuracy

Out[323]: 0.858552197958835

In [324]: print(classification\_report(y\_test,dt\_pred))

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.86      | 0.86   | 0.86     | 43699   |
| 1            | 0.86      | 0.86   | 0.86     | 43996   |
| accuracy     |           |        | 0.86     | 87695   |
| macro avg    | 0.86      | 0.86   | 0.86     | 87695   |
| weighted avg | 0.86      | 0.86   | 0.86     | 87695   |

59°F Haze Search ENG IN 2:29 PM 1/17/2023

Project Management Tool - Flip | Home Page - Select or create a | Micro Credit Project - Jupyter No | +

localhost:8889/notebooks/Micro%20Credit%20Project.ipynb

Gmail YouTube Maps DataTrained Educati... DT Support scikit-learn: machin... GitHub - training-m... Project Managem... Project Management Kaggle: Your Home...

Jupyter Micro Credit Project Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

In [325]: confusion\_matrix(y\_test,dt\_pred)

Out[325]: array([[37603, 6096],  
[ 6308, 37688]], dtype=int64)

In [ ]:

In [326]: fpr,tpr,threshold = roc\_curve(y\_test,dt\_pred)

In [327]: print('False Positive Rate =',fpr)  
print('True Positive Rate =',tpr)  
print('Threshold =',threshold)

False Positive Rate = [0. 0.13949976 1. ]  
True Positive Rate = [0. 0.85662333 1. ]  
Threshold = [2 1 0]

In [328]: plt.plot(fpr,tpr,color='green',label='ROC')  
plt.plot([0,1],[0,1],color='darkred',linestyle='--')  
plt.xlabel('False Positive Rate')  
plt.ylabel('True Positive Rate')  
plt.title('Reciever Operating Characteristic (ROC) Curve')  
plt.legend()  
plt.show()

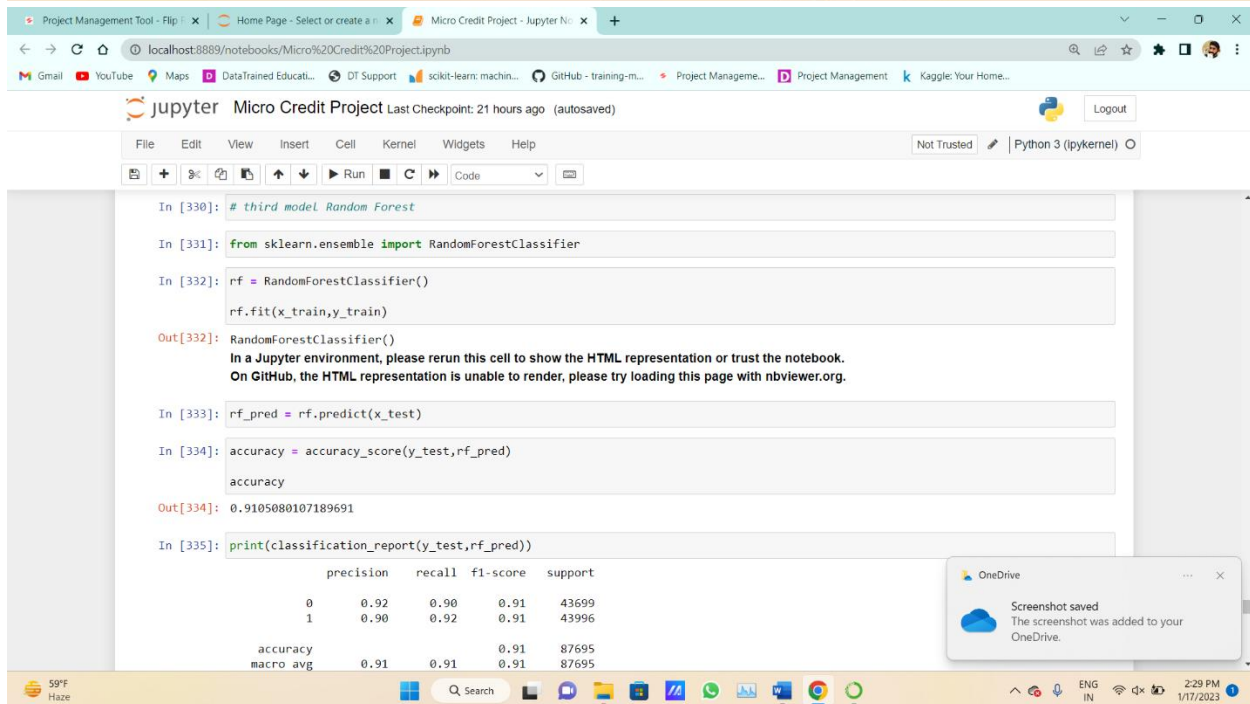
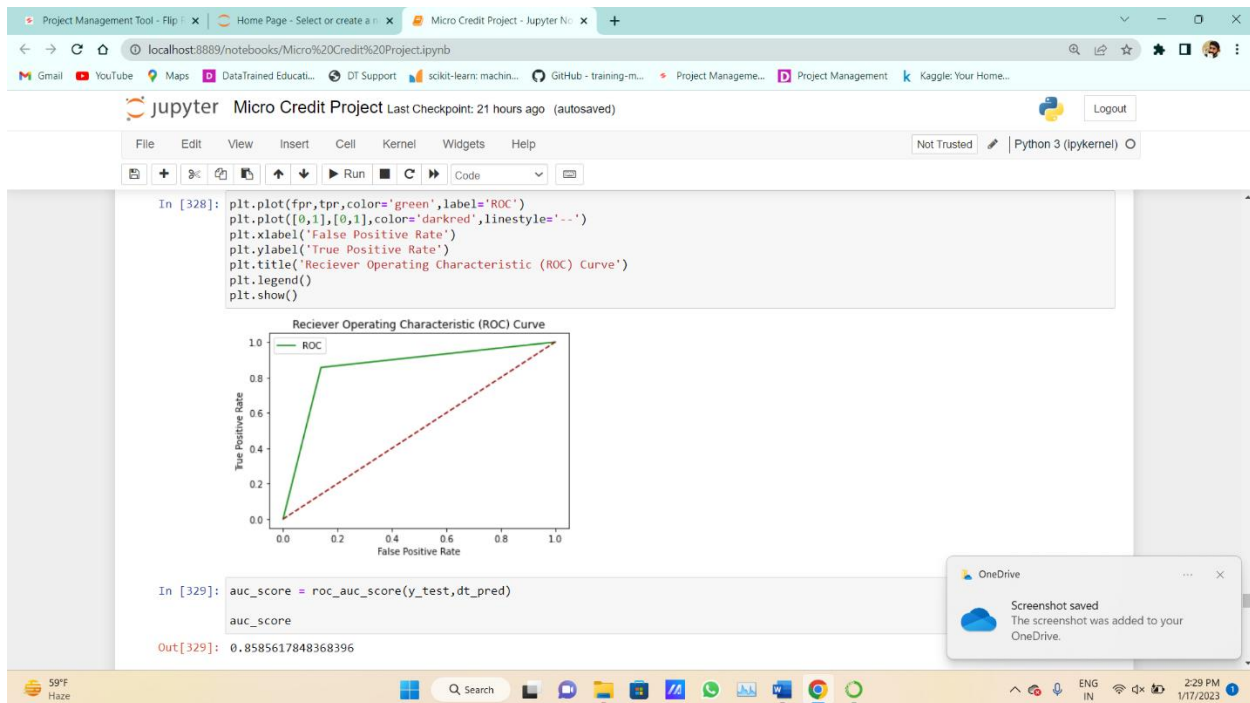
Reciever Operating Characteristic (ROC) Curve

1.0

ROC

OneDrive  
Screenshot saved  
The screenshot was added to your OneDrive.

59°F Haze Search ENG IN 2:29 PM 1/17/2023



Project Management Tool - Flip | Home Page - Select or create a | Micro Credit Project - Jupyter No | +

localhost:8889/notebooks/Micro%20Credit%20Project.ipynb

Jupyter Micro Credit Project Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [335]: print(classification_report(y_test,rf_pred))
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.92      | 0.90   | 0.91     | 43699   |
| 1            | 0.90      | 0.92   | 0.91     | 43996   |
| accuracy     |           |        | 0.91     | 87695   |
| macro avg    | 0.91      | 0.91   | 0.91     | 87695   |
| weighted avg | 0.91      | 0.91   | 0.91     | 87695   |

```
In [336]: confusion_matrix(y_test,rf_pred)
```

```
Out[336]: array([[39289, 4410],
               [ 3438, 40558]], dtype=int64)
```

```
In [ ]:
```

```
In [337]: fpr,tpr,threshold = roc_curve(y_test,rf_pred)
```

```
In [338]: print('False Positive Rate =',fpr)
           print('True Positive Rate =',tpr)
           print('Threshold =',threshold)
```

```
False Positive Rate = [0.          0.10091764 1.         ]
True Positive Rate = [0.          0.92185653 1.         ]
Threshold = [2 1 0]
```

OneDrive Screenshot saved The screenshot was added to your OneDrive.

59°F Haze

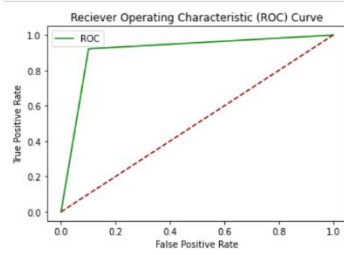
Project Management Tool - Flip | Home Page - Select or create a | Micro Credit Project - Jupyter No | +

localhost:8889/notebooks/Micro%20Credit%20Project.ipynb

Jupyter Micro Credit Project Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [339]: plt.plot(fpr,tpr,color='green',label='ROC')
           plt.plot([0,1],[0,1],color='darkred',linestyle='--')
           plt.xlabel('False Positive Rate')
           plt.ylabel('True Positive Rate')
           plt.title('Reciever Operating Characteristic (ROC) Curve')
           plt.legend()
           plt.show()
```



```
In [340]: auc_score = roc_auc_score(y_test,rf_pred)
```

```
Out[340]: 0.9104694456380424
```

OneDrive Screenshot saved The screenshot was added to your OneDrive.

59°F Haze



Project Management Tool - Flip | Home Page - Select or create a | Micro Credit Project - Jupyter No | +

localhost:8889/notebooks/Micro%20Credit%20Project.ipynb

Jupyter Micro Credit Project Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [341]: # fourth model SVM

In [342]: from sklearn.svm import SVC

In [343]: svc = SVC()
           svc.fit(x_train,y_train)

Out[343]: SVC()
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [344]: svc_pred = svc.predict(x_test)

In [345]: accuracy = accuracy_score(y_test,svc_pred)
           accuracy

Out[345]: 0.8213809225155368

In [346]: print(classification_report(y_test,svc_pred))
```

|           | precision | recall | f1-score | support |
|-----------|-----------|--------|----------|---------|
| 0         | 0.79      | 0.87   | 0.83     | 43699   |
| 1         | 0.86      | 0.77   | 0.81     | 43996   |
| accuracy  |           |        | 0.82     | 87695   |
| macro avg | 0.83      | 0.82   | 0.82     | 87695   |

OneDrive  
Screenshot saved  
The screenshot was added to your OneDrive.

59°F  
Haze

Project Management Tool - Flip | Home Page - Select or create a | Micro Credit Project - Jupyter No | +

localhost:8889/notebooks/Micro%20Credit%20Project.ipynb

Jupyter Micro Credit Project Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [346]: print(classification_report(y_test,svc_pred))
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.79      | 0.87   | 0.83     | 43699   |
| 1            | 0.86      | 0.77   | 0.81     | 43996   |
| accuracy     |           |        | 0.82     | 87695   |
| macro avg    | 0.83      | 0.82   | 0.82     | 87695   |
| weighted avg | 0.83      | 0.82   | 0.82     | 87695   |

```
In [347]: confusion_matrix(y_test,svc_pred)

Out[347]: array([[38224, 5475],
                 [10189, 33807]], dtype=int64)

In [ ]:

In [348]: fpr,tpr,threshold = roc_curve(y_test,svc_pred)

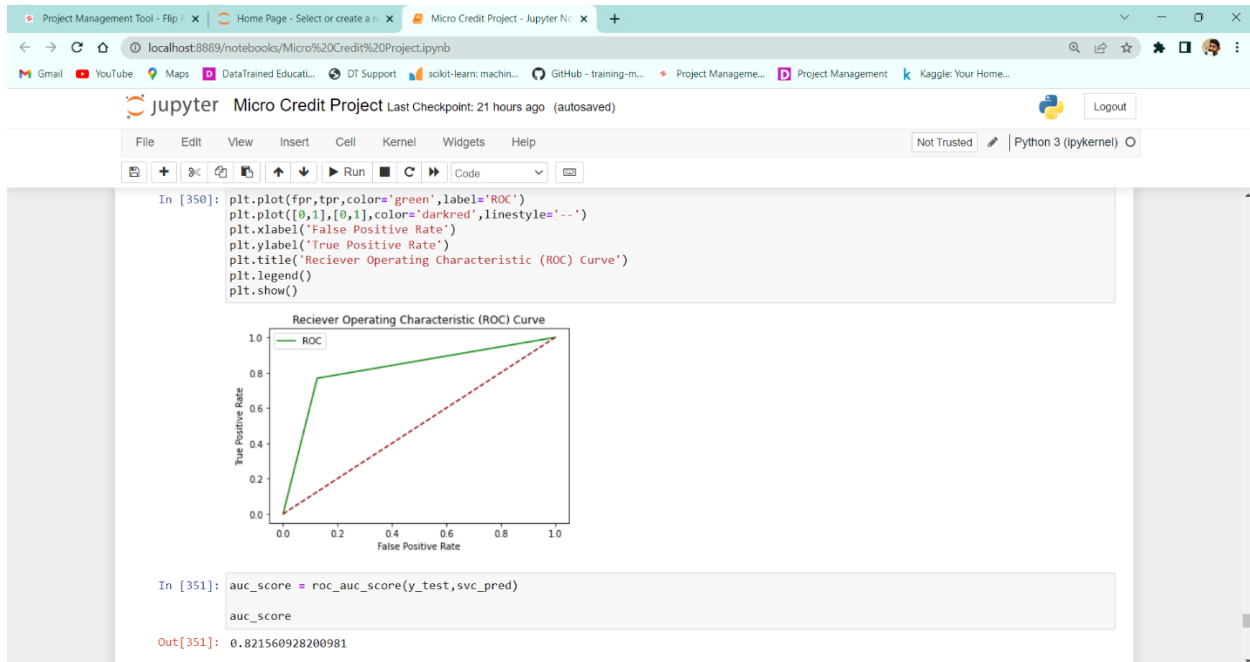
In [349]: print('False Positive Rate =',fpr)
           print('True Positive Rate =',tpr)
           print('Threshold =',threshold)

False Positive Rate = [0.
 True Positive Rate = [0.
 Threshold = [2 1 0]
                0.12528891 1.
                0.76841076 1.
                ]
                ]
```

OneDrive  
Screenshot saved  
The screenshot was added to your OneDrive.

59°F  
Haze





59°F Haze

Project Management Tool - Flip | Home Page - Select or create a | Micro Credit Project - Jupyter No | +

localhost:8889/notebooks/Micro%20Credit%20Project.ipynb

Gmail YouTube Maps DataTrained Educati... DT Support scikit-learn: machin... GitHub - training-m... Project Managem... Project Management Kaggle: Your Home...

Jupyter Micro Credit Project Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [352]: # Hyperparameter tuning for Random Forest because Random Forest is performing well, if i consider.
In [ ]:
In [353]: from sklearn.model_selection import GridSearchCV
In [354]: param_grid = {'n_estimators':[10,15],
                        'criterion':['gini','entropy'],
                        'max_depth':[10,15],
                        'min_samples_split':[10,12],
                        'min_samples_leaf':[10,12]}
In [355]: grd_srch = GridSearchCV(RandomForestClassifier(),param_grid,cv=10)
grd_srch.fit(x_train,y_train)
Out[355]: GridSearchCV(cv=10, estimator=RandomForestClassifier(),
                    param_grid={'criterion': ['gini', 'entropy'],
                                'max_depth': [10, 15], 'min_samples_leaf': [10, 12],
                                'min_samples_split': [10, 12],
                                'n_estimators': [10, 15]})
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.
In [356]: grd_srch.best_estimator_
Out[356]: RandomForestClassifier(criterion='entropy', max_depth=15, min_samples_leaf=10,
                                min_samples_split=12, n_estimators=10)
```

OneDrive

Screenshot saved  
The screenshot was added to your OneDrive.

59°F Haze

Project Management Tool - Flip | Home Page - Select or create a | Micro Credit Project - Jupyter No. x

localhost:8889/notebooks/Micro%20Credit%20Project.ipynb

Jupyter Micro Credit Project Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [356]: grd_srch.best_estimator_
Out[356]: RandomForestClassifier(criterion='entropy', max_depth=15, min_samples_leaf=10,
                                min_samples_split=12, n_estimators=10)
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [357]: Rf = RandomForestClassifier(max_depth=15,min_samples_leaf=10,min_samples_split=10,n_estimators=10)
          Rf.fit(x_train,y_train)
Out[357]: RandomForestClassifier(max_depth=15, min_samples_leaf=10, min_samples_split=10,
                                n_estimators=10)
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [358]: Rf_pred = Rf.predict(x_test)

In [359]: accuracy = accuracy_score(y_test,Rf_pred)
          accuracy
Out[359]: 0.8780318148127031

In [360]: print(classification_report(y_test,Rf_pred))
```

|   | precision | recall | f1-score | support |
|---|-----------|--------|----------|---------|
| 0 | 0.87      | 0.89   | 0.88     | 43699   |
| 1 | 0.89      | 0.87   | 0.88     | 43996   |

59°F Haze

OneDrive Screenshot saved The screenshot was added to your OneDrive.

Project Management Tool - Flip | Home Page - Select or create a | Micro Credit Project - Jupyter No. x

localhost:8889/notebooks/Micro%20Credit%20Project.ipynb

Jupyter Micro Credit Project Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [358]: Rf_pred = Rf.predict(x_test)

In [359]: accuracy = accuracy_score(y_test,Rf_pred)
          accuracy
Out[359]: 0.8780318148127031

In [360]: print(classification_report(y_test,Rf_pred))
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.87      | 0.89   | 0.88     | 43699   |
| 1            | 0.89      | 0.87   | 0.88     | 43996   |
| accuracy     |           |        | 0.88     | 87695   |
| macro avg    | 0.88      | 0.88   | 0.88     | 87695   |
| weighted avg | 0.88      | 0.88   | 0.88     | 87695   |

```
In [361]: confusion_matrix(y_test,Rf_pred)
Out[361]: array([[38900, 4799],
                 [ 5897, 38099]], dtype=int64)

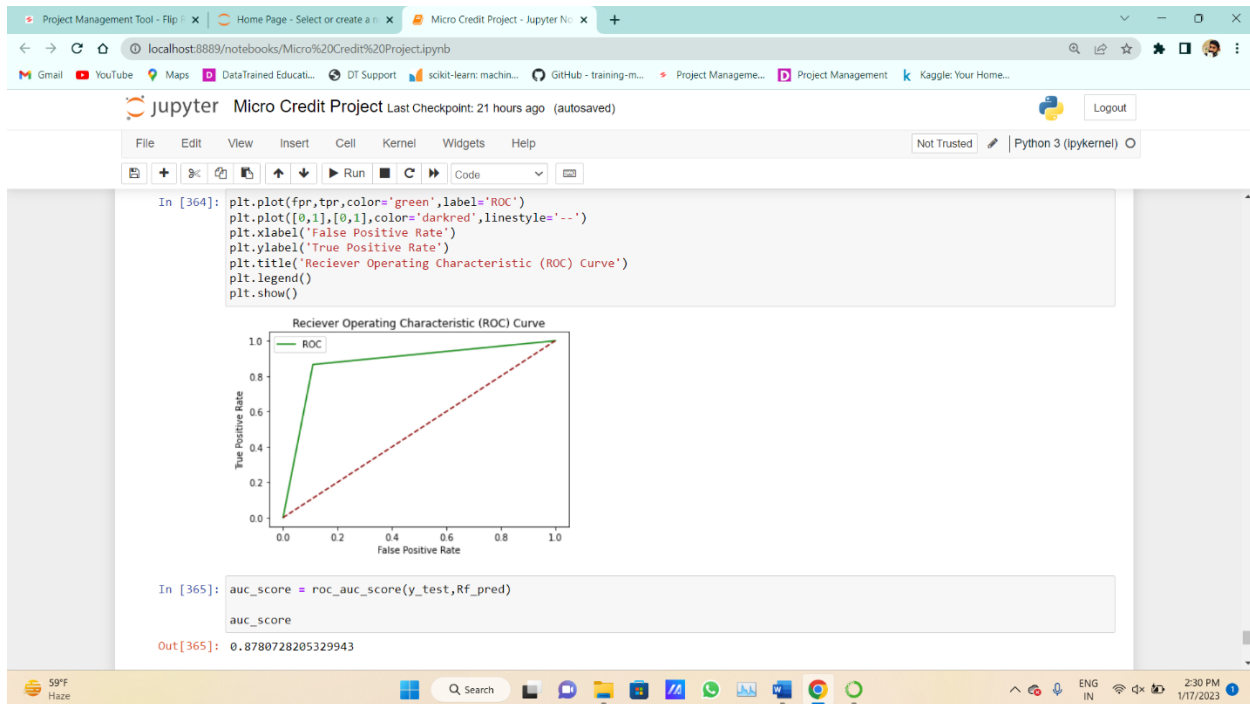
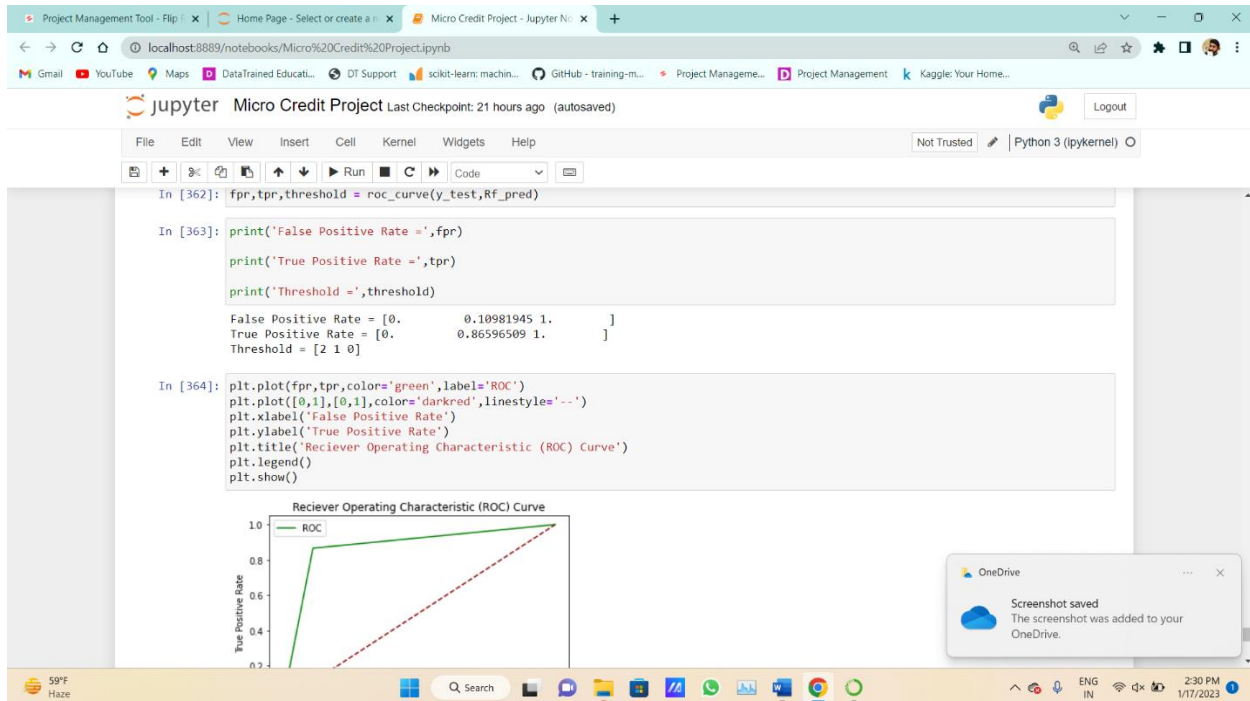
In [ ]:

In [362]: fpr, tpr, threshold = roc_curve(y_test, Rf_pred)

In [363]: print('False Positive Rate =', fpr)
```

59°F Haze

OneDrive Screenshot saved The screenshot was added to your OneDrive.



## CONCLUSION

From all above observations, saving Random Forest Regressor after hyperparameter tuning.

Because after hyperparameter tuning, Random Forest remains approx. same, a very little changes came in sight. So, it is better to consider. Most risky part of this project was to remove outliers and a little tricky part was while plotting heatmap and deleting columns which was affecting to other columns.