Micro Credit Defaulter Project

Submitted By,
Poovarasi Vijayan

Acknowledgement:

I wish to express my sincere thanks to the following companies, without whom I would have not got opportunity to work on this project; Data Trained Institute and Flip Robo Technology-Bangalore.

Introduction:

Business Problem Facing:

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.

They 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 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 loaned amount within the time duration of 5 days. For the loan amount of 5 (in Indonesian Rupiah), payback amount should be 6 (in Indonesian Rupiah), while, for the loan amount of 10 (in Indonesian Rupiah), the payback amount should be 12 (in Indonesian Rupiah).

Concept Background of the Domain Problem:

To understand the business problem, there are certain features that influence the micro finance such as the borrowers are generally from low-income backgrounds, loans availed under microfinance are usually of small amount, i.e., micro loans, the loan tenure is short, microfinance loans do not require any collateral, these loans are usually repaid at higher frequencies, the purpose of most microfinance loans is income generation.

Review of Literature:

With the rapid growth of technology and increased competition, telecom firms are looking for ways to improve the quality of their service and, as a result, the health of their revenue. Miniature credit arrangement furnishes administrators and specialist organizations with the capacity to stretch out their support of their clients through a little, transient credit office. At the point when we go through the dataset given, we should look at deliberately every one of the characteristics given, arrange the clients

between defaulters and non-defaulters, and lessen the possibility of deceitfulness in miniature credit advances by clients.

Motivation for Problem Undertaken:

This problem is taken 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 loaned amount within 5 days of insurance of loan. In this case, Label '1' indicates that the loan has been paid i.e., non-defaulter, while, Label '0' indicates that the loan has not been paid i.e., defaulter.

Analytical Problem Framing:

Mathematical/Analytical Modelling of the Problem:

This dataset contains 209593 rows and 36 columns. By looking into the target column, it is concluded that it a classification problem as label column has categorical values so classification algorithms are used build the model. Also, it is observed that in some columns there are more than 90% of values are 0. So, those columns are dropped, as it will create high skewness while model building. While checking the null values in the datasets it is found that there are no null values. To get better insight on the features plots like distribution plot, bar plot and heatmap are used so that easy understanding can be done. Also, outliers and skewness are present in dataset which is removed by z-score and power transform methods respectively. Then Data Pre-processing, Standard Scaler is used to standardize the data and VIF is checked for multicollinearity in dataset.

Once this is all done then data is ready for modelling. As the target variable contains categorical data, Classification is used. 4 classifiers – Decision Tree Classifier, Support Vector Classifier, KNeighbors Classifier, Logistic Regression, 4 ensemble -Random Forest Classifier, AdaBoost Classifier, Gradient Boosting Classifier, XGB Classifier, 5 metrics – r2_score, mean_squared_error, mean_absolute_error, Classification report, Confusion matrix are used in this project to build Classification model. From this the best model is identified and using Cross Validation technique is checked for overfitting and underfitting and Hypertuning is done to increase accuracy. Then, Finally the best model is saved.

Data Source and their Format:

The data is provided by Fliprobo, which is stored in csv format.

- Data contains 209593 entries each having 36 variables.
- Here the dataset is divided into independent and target variable.

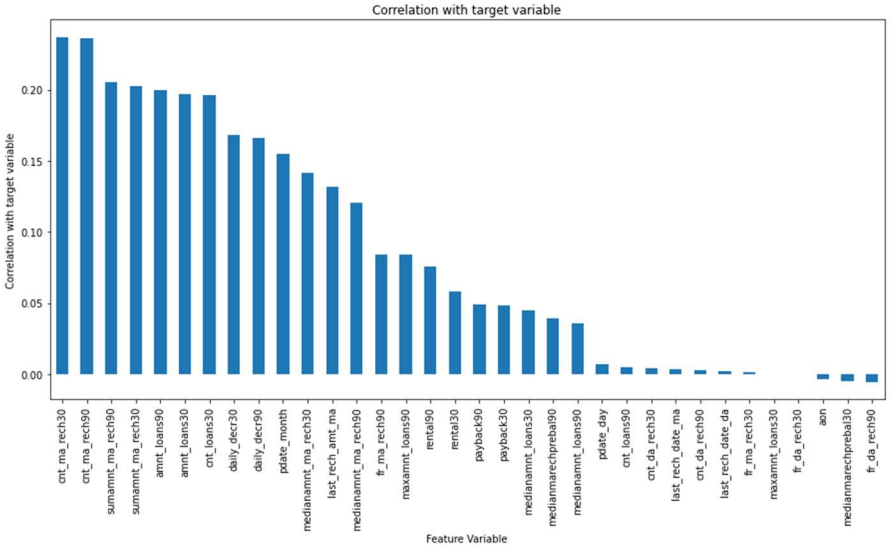
Data Pre-Processing:

In Machine Learning, data pre-processing refers to the process of cleaning and organising raw data in order to make it appropriate for creating and training Machine Learning models. In other words, anytime data is received from various sources, it is collected in raw format, which makes analysis impossible. Data pre-processing is a crucial stage in Machine Learning since the quality of data and the relevant information that can be gleaned from it has a direct impact on our model's capacity

to learn; consequently, we must pre-process our data before feeding it into our model. As a result, it is the first and most important stage in developing a machine learning model.

Some of the techniques used in this project are listed below:

1. Started with importing the required libraries and then the dataset which is in csv format.
2. Dataset contains 209593 rows and 36 variables. Then the information of the columns is observed carefully.
3. Few columns like 'Unnamed: 0' is dropped as it does not contribute much in model building.
4. Then Null values is checked, as there are no null values, proceeded with further analysis process.
5. Numeric data and Categorical data are then separated so that visualization is done with distplot for numeric data and countplot for categorical data.
6. Then all the categorical data is converted to numeric by using LabelEncoder so that analysis can be made in better way.
7. Data Description is made followed by correlation where positive and negative correlated data is checked.



- 'cnt_ma_rech30', 'cnt_ma_reach90' are highly correlated with the target column, whereas 'fr_da_rech90' is less correlated with target column.

8. Outliers and Skewness is checked and removed in order to avoid bias while model building.
    - Skewness is removed by using power_transform method.
    - Outliers are removed by using z-score method.

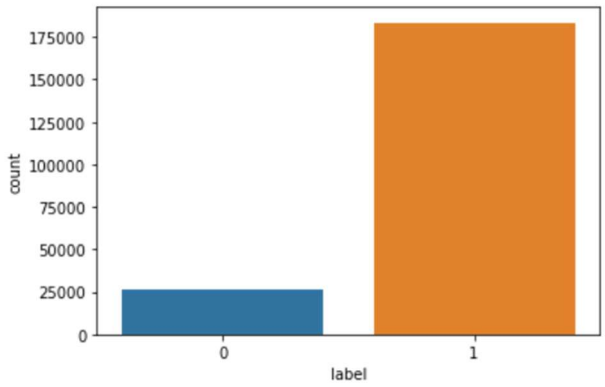
9. Standard Scaler is used for data standardization and VIF is used to check Multicolinearity is checked to find if any data variable is correlated with each other and it is removed.

10. Once all these processes are done then data is ready for model building where various Machine Learning models is used to check the accuracy of data.

Data Input Logic Output Relations:

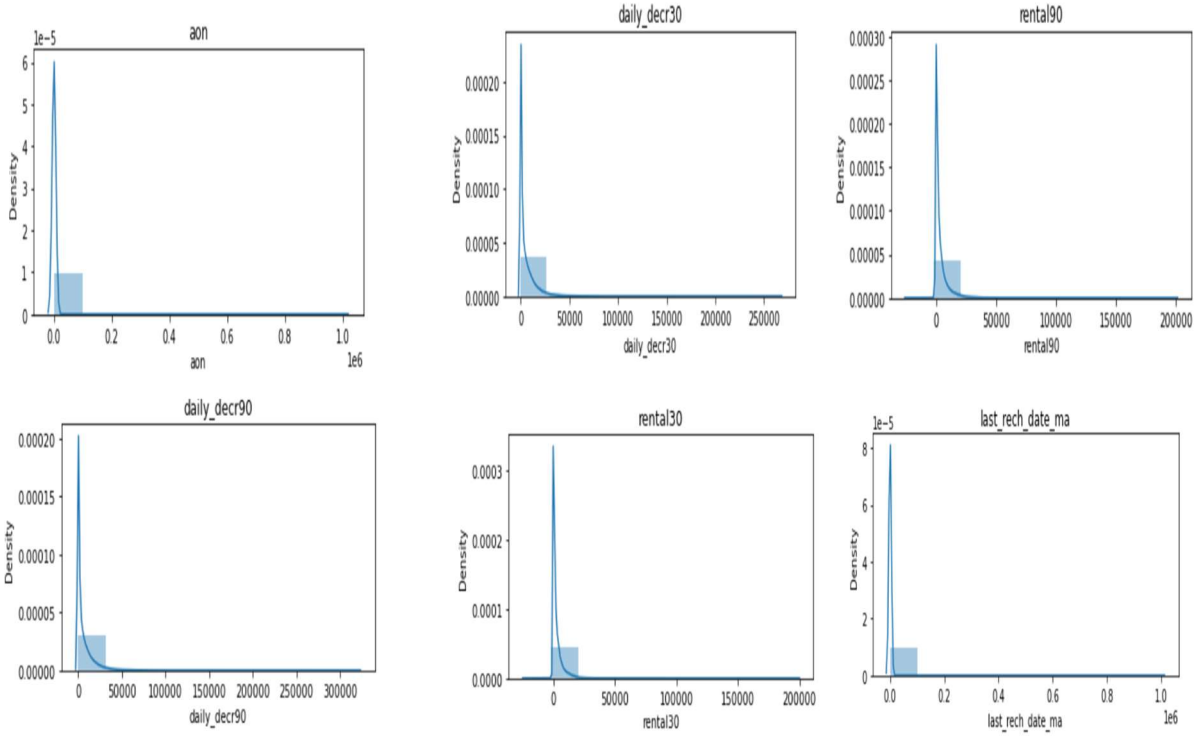
Data visualization is used to find the relation between input and the output variable.

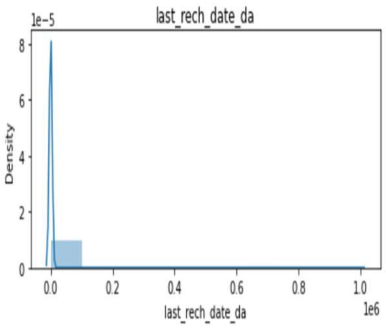
Data Visualization:

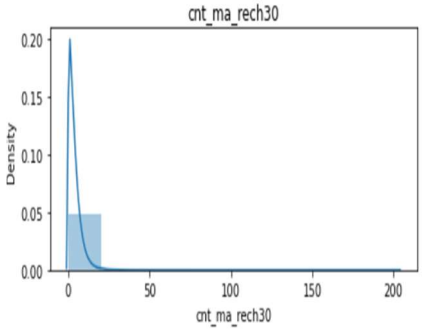


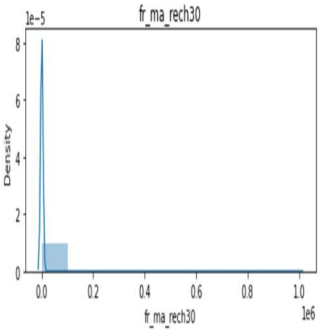
Key Observations:

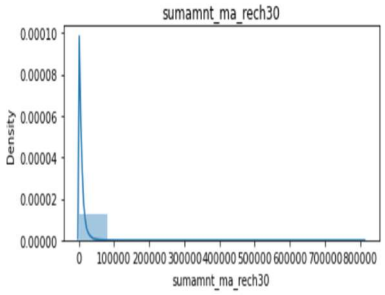
Here we have large number of 1's which means the dataset is not balanced.

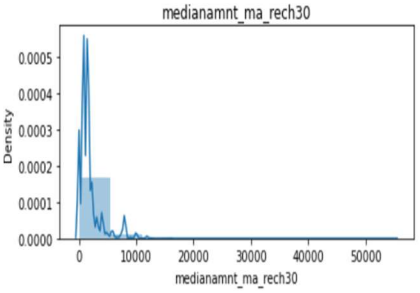


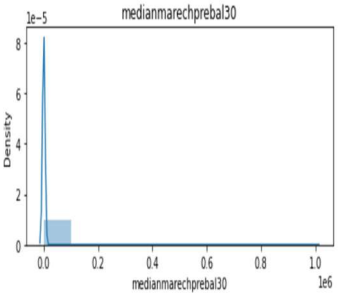

last_rech_date_da
1e−5
Density
last_rech_date_da
1e6

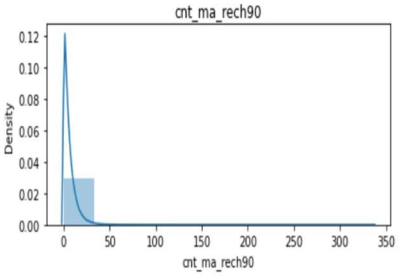

cnt_ma_rech30
Density
cnt_ma_rech30

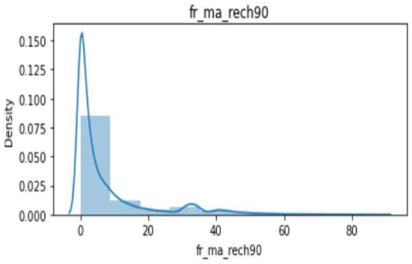

fr_ma_rech30
1e−5
Density
fr_ma_rech30
1e6

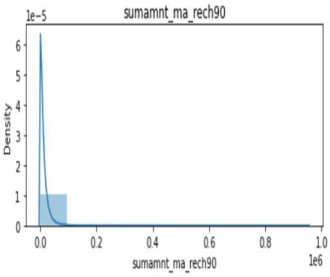

sumamnt_ma_rech30
Density
sumamnt_ma_rech30

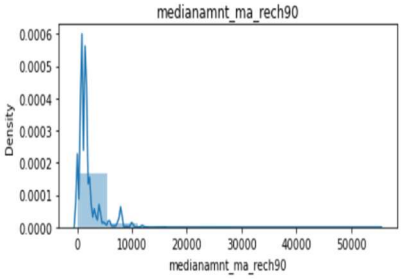

medianamnt_ma_rech30
Density
medianamnt_ma_rech30

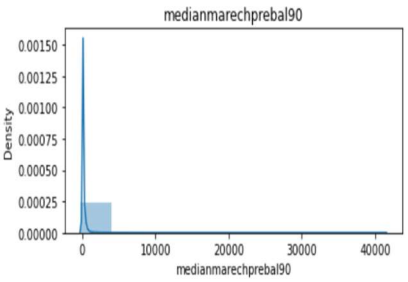

medianmarechprebal30
1e−5
Density
medianmarechprebal30
1e6

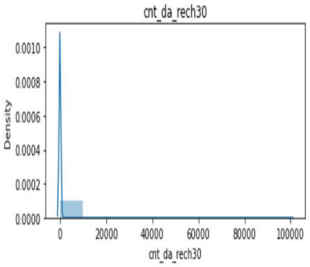

cnt_ma_rech90
Density
cnt_ma_rech90

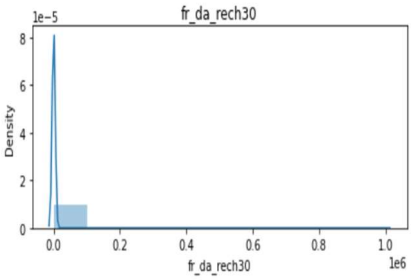

fr_ma_rech90
Density
fr_ma_rech90

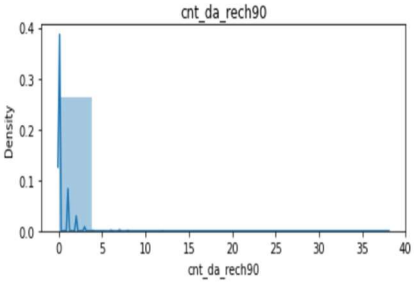

sumamnt_ma_rech90
1e−5
Density
sumamnt_ma_rech90
1e6

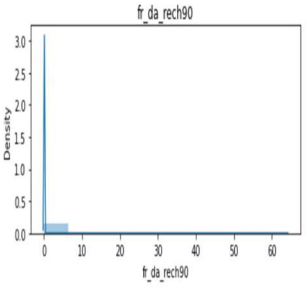

medianamnt_ma_rech90
Density
medianamnt_ma_rech90

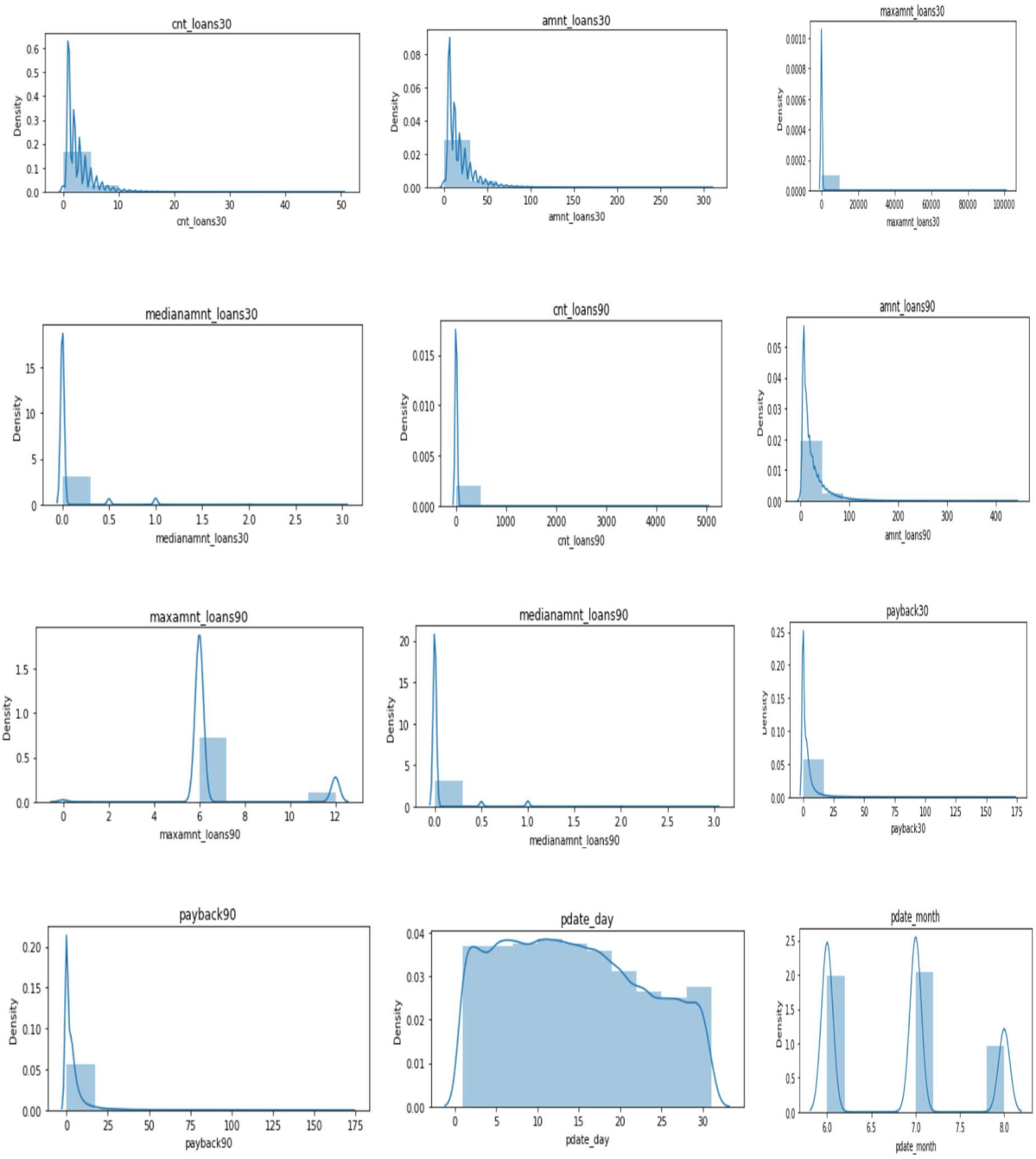

medianmarechprebal90
Density
medianmarechprebal90


cnt_da_rech30
Density
cnt_da_rech30


fr_da_rech30
1e−5
Density
fr_da_rech30
1e6


cnt_da_rech90
Density
cnt_da_rech90


fr_da_rech90
Density
fr_da_rech90



Key Observations:

1. Most of our columns are extremely right skewed.
2. maxamnt_loans90 have only three values 6, 12, 0 and 6 is in the most of the record.
3. For fr_ma_rech90 the frequency of recharge in last 90 days the frequency is very lesser i.e. most of the people are recharging in very lesser time.
4. aon age on cellular network in days most of the people are actually lesser in age on cellular network.
5. rental90, rental90, average main balance over last 90, 30 days are mostly between 0 - 1500.

Tools Used:

1. Python 3.8
2. Numpy
3. Pandas
4. Matplotlib
5. Seaborn
6. Data Science
7. Machine Learning

Model/s Development and Evaluation:

Identification of possible problem solving approach:

In this project, both Statistical and Analytical methods are used, in which Data Pre-processing, Exploratory Data Analysis is used after ensuring that data is cleaned. Here Target column is Label, as it is categorical column so classification algorithm is used. This project consists of 4 classifiers – Decision Tree Classifier, Support Vector Classifier, KNeighbors Classifier, Logistic Regression, 4 ensemble -Random Forest Classifier, AdaBoost Classifier, Gradient Boosting Classifier, XGB Classifier , 5 metrics – r2_score, mean_squared_error, mean_absolute_error, classification report, confusion matrix techniques. Out of which Rnadom Forest Classifier gave high accuracy which is also chosen as best model in which there is least difference between r2 score and cv. Then with this Hypertuning is done in which accuracy is slightly reduced in order to correct the over fitting of the model. Once all this is done, best model is saved using joblib. Then by using this saved model the output is determined which predicted the micro credit defaulter.

Test of Identified Approaches:

The approaches followed in this project are

1. Find the best random state of a model.
2. Use all the other models to find accuracy score, mean squared error, mean absolute error and r2 score, confusion_matrix, classification_report.
3. Then find Cross Validation of all models to find the best accuracy, which is the least difference between r2 score and cv score.
4. With the best model accuracy, we need to do hyper tuning using GridSearchCV.
5. Then finally saving the best model and by keeping this we need to predict the micro credit defaulter of dataset.

Run and evaluate of selected model:

```
acc = 0

for i in range(0,1000):
    x_train,x_test,y_train,y_test = train_test_split(X,y,random_state=i,test_size=.22)
    lr = LogisticRegression()
    lr.fit(x_train,y_train)
    pred_y = lr.predict(x_test)
    temp = accuracy_score(y_test,pred_y)

    if temp>acc:
        acc = temp
        best_rstate = i
print("Accuracy_Score :", acc*100, "Best Random State : ",best_rstate)
```

```
Accuracy_Score : 75.86142809352117 Best Random State :  95
```

```
mean_squared_err = []
mean_absolute_err = []
r2 = []

for m in model:
    m.fit(x_train,y_train)
    m.score(x_train,y_train)
    predm = m.predict(x_test)

    print("Accuracy Score of ", m ," is ", accuracy_score(y_test,predm))
    print("Mean Squared Error is ", mean_squared_error(y_test,predm))
    mean_squared_err.append(mean_squared_error(y_test,predm))
    print("Mean absolute error is ",mean_absolute_error(y_test,predm))
    mean_absolute_err.append(mean_absolute_error(y_test,predm))
    print("r2_score is ",r2_score(y_test,predm))
    r2.append(r2_score(y_test,predm))

    print("Confusion Matrix is ",confusion_matrix(y_test,predm))
    print("Classification Report is ", classification_report(y_test,predm))
    print("\n\n")
```

```
Accuracy Score of  LogisticRegression()  is  0.7579576007632358
Mean Squared Error is  0.24204239923676418
Mean absolute error is  0.24204239923676418
r2_score is  0.03182850518825142
Confusion Matrix is  [[30115 10296]
 [ 9239 31059]]
Classification Report is               precision    recall  f1-score   support

           0       0.77      0.75      0.76     40411
           1       0.75      0.77      0.76     40298

    accuracy                           0.76     80709
   macro avg       0.76      0.76      0.76     80709
weighted avg       0.76      0.76      0.76     80709
```

Logistic Regression gives 75% accuracy

```
Accuracy Score of  KNeighborsClassifier()  is  0.9001226629000483
Mean Squared Error is  0.09987733709995168
Mean absolute error is  0.09987733709995168
r2_score is  0.6004898684577678
Confusion Matrix is  [[39979   432]
 [ 7629 32669]]
Classification Report is               precision    recall  f1-score   support

           0       0.84      0.99      0.91     40411
           1       0.99      0.81      0.89     40298

    accuracy                           0.90     80709
   macro avg       0.91      0.90      0.90     80709
weighted avg       0.91      0.90      0.90     80709
```

KNeighborsClassifier gives 90% accuracy

```
Accuracy Score of  SVC()  is  0.8562489932969062
Mean Squared Error is  0.14375100670309382
Mean absolute error is  0.14375100670309382
r2_score is  0.4249948460298998
Confusion Matrix is  [[35542  4869]
 [ 6733 33565]]
Classification Report is               precision    recall  f1-score   support

           0       0.84      0.88      0.86     40411
           1       0.87      0.83      0.85     40298

    accuracy                           0.86     80709
   macro avg       0.86      0.86      0.86     80709
weighted avg       0.86      0.86      0.86     80709
```

SVC give 85% accuracy score

```
Accuracy Score of  DecisionTreeClassifier()  is  0.9135412407538193
Mean Squared Error is  0.08645875924618072
Mean absolute error is  0.08645875924618072
r2_score is  0.6541642850884882
Confusion Matrix is  [[37167  3244]
 [ 3734 36564]]
Classification Report is               precision    recall  f1-score   support

           0       0.91      0.92      0.91     40411
           1       0.92      0.91      0.91     40298

    accuracy                           0.91     80709
   macro avg       0.91      0.91      0.91     80709
weighted avg       0.91      0.91      0.91     80709
```

DecisionTreeClassifier gives 91% accuracy score

```
Accuracy Score of  GaussianNB()  is  0.7385173896343654
Mean Squared Error is  0.2614826103656346
Mean absolute error is  0.2614826103656346
r2_score is  -0.04593249175874803
Confusion Matrix is  [[31398  9013]
 [12091 28207]]
Classification Report is               precision    recall  f1-score   support

           0       0.72      0.78      0.75     40411
           1       0.76      0.70      0.73     40298

    accuracy                           0.74     80709
   macro avg       0.74      0.74      0.74     80709
weighted avg       0.74      0.74      0.74     80709
```

GaussianNB gives 73% accuracy

```
Accuracy Score of  RandomForestClassifier()  is  0.9535491704766507
Mean Squared Error is  0.046450829523349314
Mean absolute error is  0.046450829523349314
r2_score is  0.8141963176836834
Confusion Matrix is  [[38850  1561]
 [ 2188 38110]]
Classification Report is               precision    recall  f1-score   support

           0       0.95      0.96      0.95     40411
           1       0.96      0.95      0.95     40298

    accuracy                           0.95     80709
   macro avg       0.95      0.95      0.95     80709
weighted avg       0.95      0.95      0.95     80709
```

RandomForestClassifier gives 95% accuracy

```
Accuracy Score of  AdaBoostClassifier()  is  0.848740536990918
Mean Squared Error is  0.15125946300908202
Mean absolute error is  0.15125946300908202
r2_score is  0.3949609619318235
Confusion Matrix is  [[35242  5169]
 [ 7039 33259]]
Classification Report is               precision    recall  f1-score   support

           0       0.83      0.87      0.85     40411
           1       0.87      0.83      0.84     40298

    accuracy                           0.85     80709
   macro avg       0.85      0.85      0.85     80709
weighted avg       0.85      0.85      0.85     80709
```

AdaBoostClassifier gives 84% accuracy

```
Accuracy Score of  XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
              importance_type=None, interaction_constraints='',
              learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
              missing=nan, monotone_constraints='()', n_estimators=100,
              n_jobs=0, num_parallel_tree=1, predictor='auto', random_state=0,
              reg_alpha=0, reg_lambda=1, ...)  is  0.9493117248386177
Mean Squared Error is  0.05068827516138225
Mean absolute error is  0.05068827516138225
r2_score is  0.7972465019055611
Confusion Matrix is  [[38055  2356]
 [ 1735 38563]]
Classification Report is               precision    recall  f1-score   support

           0       0.96      0.94      0.95     40411
           1       0.94      0.96      0.95     40298

    accuracy                           0.95     80709
   macro avg       0.95      0.95      0.95     80709
weighted avg       0.95      0.95      0.95     80709
```

XGBClassifier gives 94% accuracy

Cross Validation:

```
Score of LogisticRegression()  is  [0.7528485  0.75248051 0.75449
763 0.75289965 0.75656595]
Mean Score is  0.7538584459074107
Standard Deviation is  0.0015216508136214198
```

CV Score of Logistic Regression is 0.75

```
Score of KNeighborsClassifier()  is  [0.89471461 0.89963474 0.900
07087 0.90035573 0.90000136]
Mean Score is  0.8989554608218103
Standard Deviation is  0.002132841848831863
```

CV Score of KNeighborsClassifier 0.89

```
Score of SVC()  is  [0.84705337 0.85186447 0.85339094 0.85386597
0.85709613]
Mean Score is  0.8526541758766196
Standard Deviation is  0.003278256824718337
```

CV Score of SVC is 0.85

```
Score of DecisionTreeClassifier()  is  [0.84532247 0.92482146 0.9
2559832 0.92441155 0.92874569]
Mean Score is  0.909779897392035
Standard Deviation is  0.03226458765024184
```

CV Score of DecisionTreeClassifier 0.90

```
Score of RandomForestClassifier()  is  [0.89153901 0.96613149 0.9
6569536 0.96598111 0.96512246]
Mean Score is  0.9508938862333576
Standard Deviation is  0.029679443007705014
```

CV Score of RandomForestClassifier is 0.95

```
Score of AdaBoostClassifier()  is  [0.81340293 0.85134656 0.84983
372 0.85278925 0.8555969 ]
Mean Score is  0.8445938733744403
Standard Deviation is  0.015710890134473403
```

CV Score of AdaBoostClassifier is 0.84

```
Score of GradientBoostingClassifier()  is  [0.84721692 0.90304203
0.90550891 0.90590288 0.90620272]
Mean Score is  0.8935746936313589
Standard Deviation is  0.0232058196876137
```

CV Score of GradientBoostingClassifier is 0.89

```
Score of XGBClassifier(base_score=0.5, booster='gbtree', callbacks=None,
              colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
              early_stopping_rounds=None, enable_categorical=False,
              eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
              importance_type=None, interaction_constraints='',
              learning_rate=0.300000012, max_bin=256, max_cat_to_onehot=4,
              max_delta_step=0, max_depth=6, max_leaves=0, min_child_weight=1,
              missing=nan, monotone_constraints='()', n_estimators=100,
              n_jobs=0, num_parallel_tree=1, predictor='auto', random_state=0,
              reg_alpha=0, reg_lambda=1, ...)  is  [0.79669356 0.97045194 0.97077904 0.97107849 0.97095583]
Mean Score is  0.935991773522103
Standard Deviation is  0.06964942522362982
```

CV Score of XGBClassifier is 0.93

HyperTuning:

```
from sklearn.model_selection import GridSearchCV

param_grid = [{'n_estimators':[20,50,100],
               'criterion':['gini','entropy'],
               'max_features':['auto','sqrt','log2']}]

rf = RandomForestClassifier(random_state=42,n_jobs=-1)
```

```
rf_random = GridSearchCV(rf,param_grid,cv=5)
```

```
rf_random.fit(x_train,y_train)
```

```
GridSearchCV(cv=5, estimator=RandomForestClassifier(n_jobs=-1, random_state=42),
             param_grid=[{'criterion': ['gini', 'entropy'],
                          'max_features': ['auto', 'sqrt', 'log2'],
                          'n_estimators': [20, 50, 100]}])
```

```
rf_random.best_params_
```

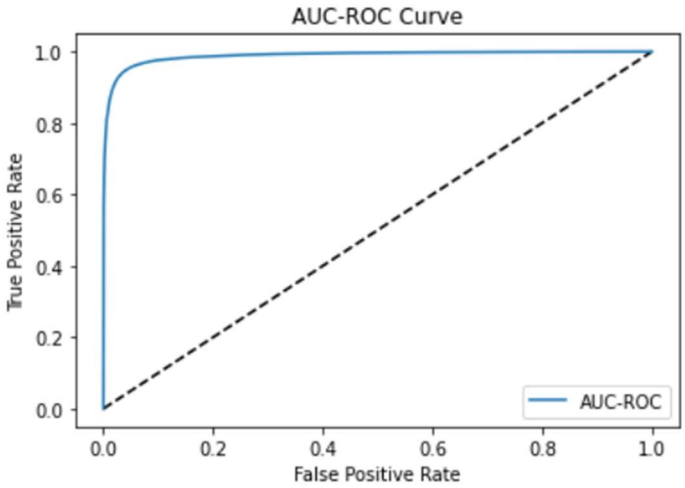
```
{'criterion': 'entropy', 'max_features': 'auto', 'n_estimators': 100}
```

```
rf_random.best_score_
```

```
0.9493445714370315
```

After Hypertuning we have 94.93% accuracy score.

AUC-ROC Curve:




```
auc_score = roc_auc_score(y_test,rfc.predict(x_test))
print(auc_score)
```

```
0.9535382025768411
```

With AUC-ROC we get 95.3% accuracy

Saving the best model:

The best model is saved using joblib.

```
import joblib
joblib.dump(predy,"Micro_Credit_Defaulter.obj")
```

```
['Micro_Credit_Defaulter.obj']
```

Key Metrics for success in solving problem under consideration:

Some of key metrics for the evaluation of this project are

1. R2_score
2. Mean_squared_error
3. Mean_absolute_error
4. Classification_report
5. Confusion_matrix

6. Cross_validation
7. Hypertuning

R2 Score:

R-squared is a statistical measure that represents the goodness of fit of a regression model. The ideal value for r-square is 1. Closer the r2_square value to 1 better the model fits.

Mean Squared Error:

The average of the squares of the errors — that is, the average squared difference between the estimated values and what is estimated is measured by the MSE of an estimator (of a process for estimating an unobserved variable). MSE is a risk function that represents the squared error loss's anticipated value. The Root Mean Squared Error is abbreviated as RMSE.

Mean Absolute Error:

MAE is a statistic that assesses the average magnitude of mistakes in a set of forecasts without taking into account their direction. It's the average of the absolute differences between forecast and actual observation over the test sample, where all individual deviations are given equal weight.

Classification Report:

A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report.

Confusion Matrix:

A confusion matrix is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm.

Cross Validation:

Cross-validation aids in determining the model's overfitting and underfitting. The model is constructed to run on several subsets of the dataset in cross validation, resulting in numerous measurements of the model. If we fold the data five times, it will be separated into five parts, each representing 20% of the whole dataset. During the Cross-validation, the first part (20%) of the 5 parts will be left out as a holdout set for validation, while the rest of the data will be utilised for training. We'll acquire the initial estimate of the dataset's model quality this way.

Further rounds are produced in the same way for the second 20% of the dataset, which is kept as a holdout set while the remaining four portions are utilised for training data during the process. We'll acquire the second estimate of the dataset's model quality this way. During the cross-validation procedure, these stages are repeated to obtain the remaining estimate of model quality.

Hypertuning:

Hyperparameters in Machine learning are those parameters that are explicitly defined by the user to control the learning process. These hyperparameters are used to improve the learning of the model, and their values are set before starting the learning process of the model.

Interpretation of the Results:

Data is trained on different models and got different results for the different algorithms. Random Forest classifier model gave us 95% accuracy and cross validation score of 95%. The code prints out the number of false positives it detected and compares it with the actual values. This is used to calculate the accuracy score and precision of the algorithms. These results along with the classification report for each algorithm are given in the output, where class 0 means the transaction was determined to be valid and 1 means it was determined as a fraud transaction. This result matched the class values to check for false positives.

CONCLUSION:

Key Findings:

- Around 28% users are highly defaulters with a mostly negative or null balance.
- Users with high equilibrium and a much lower number are defaulter.
- Nonstandard loans (i.e.,98 percent of the category) are paid to users who take up more loans as they pay back the loan within 5 days.
  10% to 12% of users are defaulters in the Average and Low Balance categories.
- Non-defaulting users who have taken no loans.
  Around 97% users are taking large loans which fall into non-default categories.
- Defaulters include 40 percent of the users that do not have a single recharge in 3 months.
- Around 14 percent of users fall into the category of defaulting loans, on average.
- The default is only 40% of users who do not reload in 90 days.
- Users who recharge very high pay their loans on time. That is, 98% of them are non-defaulting ones.
- defaulting is 34 percent of users who reload less.
- Old and largely non default users are trusted
- 17% of users receiving small loans are non-performing.
- The new users constitute 32% of the users defaulting.
- Of users who recharge and pay their loans on time, 99 percent are more in number, which is good news for the company than for any other category.

Learning outcomes of study in Data Science:

The above research will help our client to study about the defaulter in micro credit defaulter which can be improved based on the outcomes of the model building.

Limitations of this work and scope of Future work:

The limitation of the study is that as the dataset contains large number of data, there are large number of outliers and skewness present which need to be removed before building the model. And as the dataset is large the time to run the models also increases which is cost effective.

Reference:

A few external references have been used to complete this project successfully. Below are the sources

1. https://scikit-learn.org/
2. https://github.com/
3. https://analyticsvidhya.com/