



PROJECT REPORT ON

“Micro-Credit Defaulter Model”

SUBMITTED BY

Ankit Dadarwala

ACKNOWLEDGMENT

I would like to express my special gratitude to “Flip Robo” team, who has given me this opportunity to deal with a beautiful dataset and it has helped me to improve my analyzation skills. And I want to express my huge gratitude to Ms. Sapna Verma , she is the person who has helped me in tickets to get out of all the difficulties I faced while doing the project. And also Mr. Shubham Yadav (SME Flip Robo) who has inspired me in so many aspects and also encouraged me a lot with his valuable words and with his unconditional support I have ended up with a beautiful Project.

A huge thanks to my academic team “Data trained” who are the reason behind what I am today. Last but not least my parents who have been my backbone in every step of my life. And also thank you for many other persons who has helped me directly or indirectly to complete the project.

INTRODUCTION

1.1 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.

1.2 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 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). 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 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.

1.3 Review of Literature

An attempt has been made in this report to review the available literature in the area of microfinance. Approaches to microfinance, issues related to measuring social impact versus profitability of MFIs, issue of sustainability, variables impacting sustainability, affect of regulations of profitability and impact assessment of MFIs have been summarized in the below report. We hope that the below report of literature will provide a platform for further research and help the industry to combine theory and practice to take microfinance forward and contribute to alleviating the poor from poverty.

1.4 Motivation for the Problem Undertaken

I have to model the micro credit defaulters with the available independent variables. This model will then be used by the management to understand how the customer is considered as defaulter or non-defaulter based on the independent variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the management to understand whether the customer will be paying back the loaned amount within 5 days of insurance of loan. The **relationship between predicting defaulter and the economy** is an important motivating factor for predicting micro credit defaulter model.

Analytical Problem Framing

2.1 Mathematical/ Analytical Modeling of the Problem

In this particular problem I had label as my target column and it was having two classes 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. So clearly it is a binary classification problem and I have to use all classification algorithms while building the model. There was no null values in the dataset. Also, I observed some unnecessary entries in some of the columns like in some columns I found more than 90% zero values so I decided to drop those columns. If I keep those columns as it is, it will create high skewness in the model. To get better insight on the features I have used plotting like distribution plot, bar plot and count plot. With these plotting I was able to understand the relation between the features in better manner. Also, I found outliers and skewness in the dataset so I removed outliers using percentile method and I removed skewness using yeo-johnson method. I have used all the classification algorithms while building model then tuned the best model and saved the best model. At last I have predicted the label using saved model.

2.2 Data Sources and their formats

The data was collected for my internship company – Flip Robo technologies in excel format. The sample data is provided to us from our client database. It is hereby given to us 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.

Also, my dataset was having 209593 rows and 36 columns including target. In this particular datasets I have object, float and integer types of data. The information about features is as follows.

Features Information:

1. label : Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan{1:success, 0:failure}
2. msisdn : mobile number of user
3. aon : age on cellular network in days
4. daily_decr30 : Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)
5. daily_decr90 : Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)
6. rental30 : Average main account balance over last 30 days
7. rental90 : Average main account balance over last 90 days
8. last_rech_date_ma : Number of days till last recharge of main account
9. last_rech_date_da: Number of days till last recharge of data account
10. last_rech_amt_ma : Amount of last recharge of main account (in Indonesian Rupiah)
11. cnt_ma_rech30 : Number of times main account got recharged in last 30 days
12. fr_ma_rech30 : Frequency of main account recharged in last 30 days
13. sumamnt_ma_rech30 : Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)
14. medianamnt_ma_rech30 : Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)
15. medianmarechprebal30 : Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)
16. cnt_ma_rech90 : Number of times main account got recharged in last 90 days
17. fr_ma_rech90 : Frequency of main account recharged in last 90 days
18. sumamnt_ma_rech90 : Total amount of recharge in main account over last 90 days (in Indonesian Rupiah)
19. medianamnt_ma_rech90 : Median of amount of recharges done in main account over last 90 days at user level (in Indonesian Rupiah)
20. medianmarechprebal90 : Median of main account balance just before recharge in last 90 days at user level (in Indonesian Rupiah)
21. cnt_da_rech30 : Number of times data account got recharged in last 30 days
22. fr_da_rech30: Frequency of data account recharged in last 30 days
23. cnt_da_rech90 : Number of times data account got recharged in last 90 days
24. fr_da_rech90 : Frequency of data account recharged in last 90 days
25. cnt_loans30 : Number of loans taken by user in last 30 days
26. amnt_loans30: Total amount of loans taken by user in last 30 days
27. maxamnt_loans30 : maximum amount of loan taken by the user in last 30 days
28. medianamnt_loans30 : Median of amounts of loan taken by the user in last 30 days
29. cnt_loans90 : Number of loans taken by user in last 90 days
30. amnt_loans90 : Total amount of loans taken by user in last 90 days

31. maxamnt_loans90 : maximum amount of loan taken by the user in last 90 days
32. medianamnt_loans90 : Median of amounts of loan taken by the user in last 90 days
33. payback30 : Average payback time in days over last 30 days
34. payback90 : Average payback time in days over last 90 days
35. pcircle : telecom circle
36. pdate : date

2.3 Data Pre-processing Done

1. As a first step I have imported required libraries and I have imported the dataset which was in csv format.
2. Then I did all the analysis like checking shape, nunique, value counts, info ,duplicates value null values etc.....
3. Then while looking into the value counts I found some columns with more than 90% zero values this creates skewness in the model and there are chances of getting model bias so I have dropped those columns with more than 90% zero values.
4. While checking for null values I found no null values in the dataset.
5. I have also dropped Unnamed:0, msisdn and pcircle column as I found they are useless.
6. Next as a part of feature extraction I converted the pdate column to pmonth and pday. Thinking that this data will help us more than pdate.
7. In some columns I found negative values which were unrealistic so I have converted those negative values to positive using abs() command.
8. Also, I have converted all the float values in maxamnt_loans90 to zero as it is specified in the problem statement we can have only 0,6,12 as maximum amount of loan taken by the user in last 30 days. As well I have dropped all the data with amnt_loans90=0 as it gives the persons who have not taken any loans.
- 9.

2.4 Data Inputs- Logic- Output Relationships

- ✓ Since I had all numerical columns I have plotted dist plot to see the distribution of each column data.
- ✓ I have used box plot for each pair of categorical features that shows the relation between label and independent features. Also we can observe

whether the person pays back the loan within the date based on features.

- ✓ In maximum features relation with target I observed Non-defaulter count is high compared to defaulters.

2.6 Hardware and Software Requirements and Tools Used

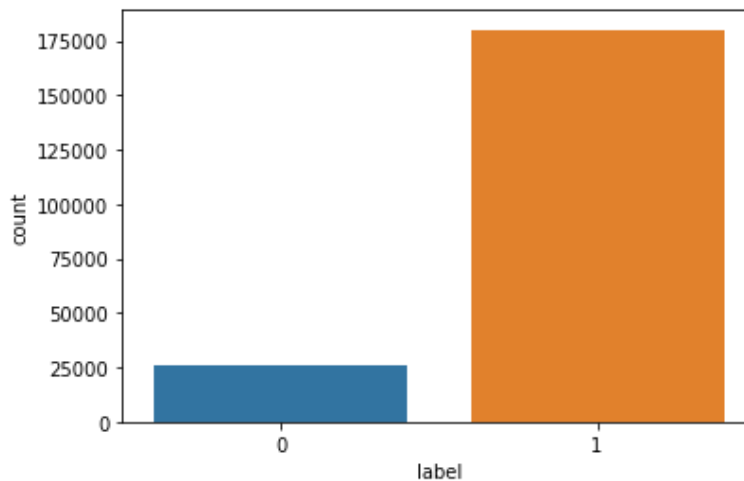
- ✓ Device name: HP Pavilion
- ✓ Processor: AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz
- ✓ RAM: 8.00 GB
- ✓ System type: 64-bit operating system, x64-based processor
- ✓ Jupyter NoteBooks Version : 6.4.3
- ✓ Python3 version : 3.8.8

❖ Libraries required :-

- ✓ Import pandas as pd
- ✓ Import numpy as np
- ✓ Import matplotlib.pyplot as plt
- ✓ Import seaborn as sns
- ✓ Import warnings
- ✓ From sklearn.model_selection import train_test_split, GridSearchCV
- ✓ from sklearn.preprocessing import MinMaxScaler
- ✓ from sklearn.ensemble import RandomForestClassifier
- ✓ from sklearn.ensemble import AdaBoostClassifier
- ✓ from sklearn.ensemble import GradientBoostingClassifier
- ✓ from sklearn.ensemble import BaggingClassifier
- ✓ from sklearn.tree import DecisionTreeClassifier
- ✓ from linear_model import LogisticRegression, SGDClassifier
- ✓ from xgboost import XGBClassifier
- ✓ from sklearn.metrics import classification_report
- ✓ from sklearn.metrics import accuracy_score
- ✓ from sklearn.model_selection import cross_val_score

Data Analysis and Visualization

3.1 Identification of possible problem-solving approaches (methods)



- From the value counts observation I find imbalance dataset in which defaulter values is less and Non defaulter values is high. About to 15% and 85% respectively
- To remove outliers I have used percentile method. And to remove skewness I have used yeo-johnson method. We have dropped all the unnecessary columns in the dataset according to our understanding. Use of Pearson's correlation coefficient to check the correlation between dependent and independent features. Also I have used Normalization to scale the data. After scaling we have to balance the target column using oversampling. Then followed by model building with all Classification algorithms. I have used oversampling (SMOTETomek) to get rid of data imbalancing. The balanced output looks like this.

3.2 Testing of Identified Approaches (Algorithms)

Since label was my target and it was a classification column with 0-defaulter and 1-Non-defaulter, so this particular problem was Classification problem. And I have used all Classification algorithms to build my model. By looking into the difference of accuracy score and cross validation score I found RandomForestClassifier as a best model. Also to get the best model we have to run through multiple models and to avoid the confusion of overfitting we have

go through cross validation. Below are the list of classification algorithms I have used in my project.

- LogisticRegression
- SGDClassifier
- XGBClassifier
- DecisionTreeClassifier
- RandomForestClassifier
- GradientBoostingClassifier
- BaggingClassifier
- AdaBoostClassifier

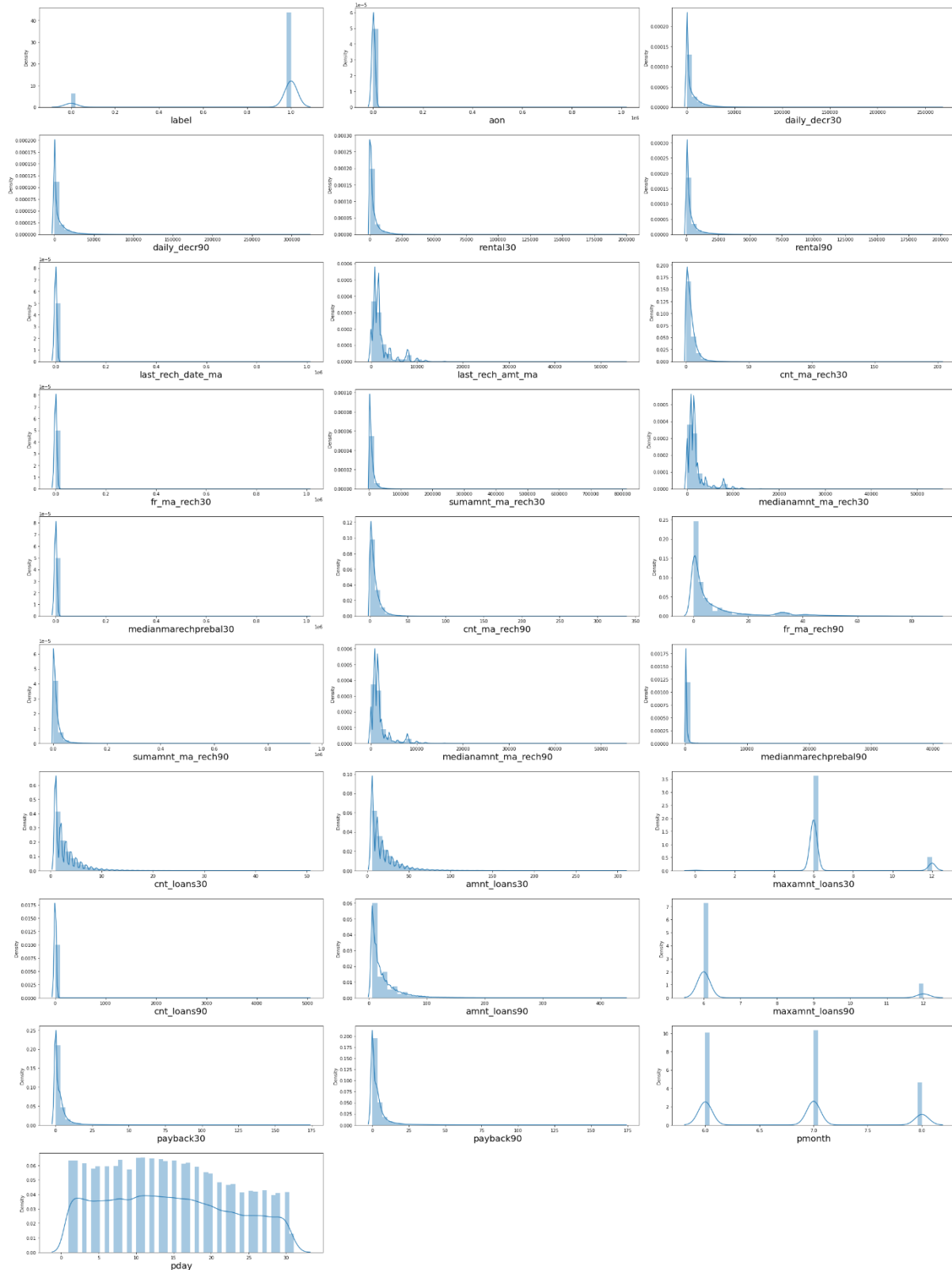
3.3 Key Metrics for success in solving problem under consideration

I have used the following metrics for evaluation:

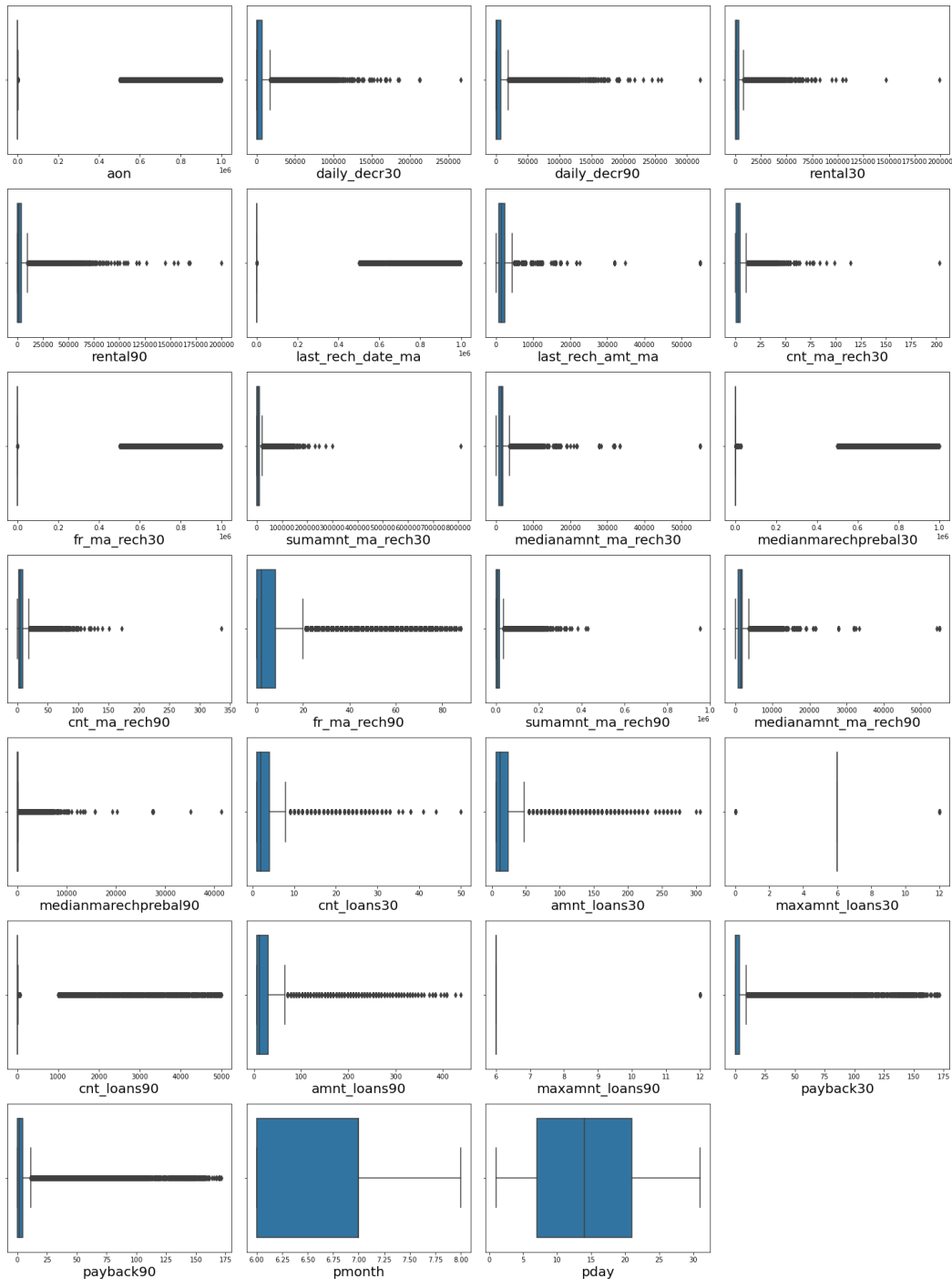
- **Precision** can be seen as a measure of quality, higher precision means that an algorithm returns more relevant results than irrelevant ones.
- **Recall** is used as a measure of quantity and high recall means that an algorithm returns most of the relevant results.
- **Accuracy score** is used when the True Positives and True negatives are more important. Accuracy can be used when the class distribution is similar.
- **F1-score** is used when the False Negatives and False Positives are crucial. While F1-score is a better metric when there are imbalanced classes.
- **Cross_val_score**: To run cross-validation on multiple metrics and also to return train scores, fit times and score times. Get predictions from each split of cross-validation for diagnostic purposes. Make a scorer from a performance metric or loss function.
- **AUC_ROC_score**: ROC curve. It is a plot of the false positive rate (x-axis) versus the true positive rate (y-axis) for a number of different candidate threshold values between 0.0 and 1.0
- I have used accuracy_score since I have balanced my data using oversampling.

3.4 Visualizations

I have used bar plots to see the relation of numerical feature with target and I have used 2 types of plots for numerical columns one is distribution plot for univariate and bar plot for bivariate analysis.



- ✓ We can clearly see that there is skewness in most of the columns so we have to treat them using suitable methods.



- ✓ We can clearly see that there is Outliers in most of the columns so we have to treat them using suitable methods.



Observations:

- ✓ 1. Customers with high value of Age on cellular network in days(aon) are maximum defaulters(who have not paid there loan amount-0).
- ✓ 2. Customers with high value of Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)(daily_decr30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 3. Customers with high value of Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)(daily_decr90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 4. Customers with high value of Average main account balance over last 30 days(rental30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 5. Customers with high value of Average main account balance over last 90 days(rental90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 6. Customers with high Number of days till last recharge of main account(last_rech_date_ma) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 7. Customers with high value of Amount of last recharge of main account (in Indonesian Rupiah)(last_rech_amt_ma) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 8. Customers with high value of Number of times main account got recharged in last 30 days(cnt_ma_rech30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 9. Customers with high value of Frequency of main account recharged in last 30 days(fr_ma_rech30) are maximum Non-defaulters(who have paid there loan amount-1) and also the count is high for defaulters comparitively Non-defaulters are more in number.
- ✓ 10. Customers with high value of Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)(sumamnt_ma_rech30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 11. Customers with high value of Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)(medianamnt_ma_rech30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 12. Customers with high value of Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)(medianmarechprebal30) are maximum defaulters(who have not paid there loan amount-0).

- ✓ 13. Customers with high value of Number of times main account got recharged in last 90 days(cnt_ma_rech90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 14. Customers with high value of Frequency of main account recharged in last 90 days(fr_ma_rech90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 15. Customers with high value of Total amount of recharge in main account over last 90 days (in Indonesian Rupiah)(sumamnt_ma_rech90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 16. Customers with high value of Median of amount of recharges done in main account over last 90 days at user level (in Indonesian Rupiah)(medianamnt_ma_rech90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 17. Customers with high value of Median of main account balance just before recharge in last 90 days at user level (in Indonesian Rupiah)(medianmarechprebal90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 18. Customers with high value of Number of loans taken by user in last 30 days(cnt_loans30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 19. Customers with high value of Total amount of loans taken by user in last 30 days(amnt_loans30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 20. Customers with high value of maximum amount of loan taken by the user in last 30 days(maxamnt_loans30) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 21. Customers with high value of Number of loans taken by user in last 90 days(cnt_loans90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 22. Customers with high value of Total amount of loans taken by user in last 90 days(amnt_loans90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 23. Customers with high value of maximum amount of loan taken by the user in last 90 days(maxamnt_loans90) are maximum Non-defaulters(who have paid there loan amount-1).
- ✓ 24. Customers with high value of Average payback time in days over last 30 days(payback30) are maximum Non-defaulters(who have paid there loan amount-1).

- ✓ 25. Customers with high value of Average payback time in days over last 90 days (payback90) are maximum Non-defaulters (who have paid their loan amount-1).
- ✓ 26. In between 6th and 7th month maximum customers both defaulters and Non-defaulters have paid their loan amount.
- ✓ 27. Below 14th of each month all the customers have paid their loan amount.

3.5 Run and evaluate select models

1). Decision Tree Classifier

```
0]: classifiers(dtc)
```

```
DecisionTreeClassifier()
99.99916245100339
```

```
cross value score 90.35606485292354
```

```
ACCURACY SCORE:
```

```
87.23283953409477
```

```
ROC AUC SCORE:
```

```
76.0634084497515
```

```
CONFUSION MATRIX:
```

```
[[ 4771  3034]
```

```
 [ 4869 49227]]
```

```
CLASSIFICATION REPORT:
```

	precision	recall	f1-score	support
0	0.49	0.61	0.55	7805
1	0.94	0.91	0.93	54096
accuracy			0.87	61901
macro avg	0.72	0.76	0.74	61901
weighted avg	0.89	0.87	0.88	61901

2). SGD Classifier

```
In [61]: classifiers(sgd)

SGDClassifier()
76.58422392709974

cross value score 76.67007303000362
ACCURACY SCORE:
78.54315762265554
ROC AUC SCORE:
75.74579671018338
CONFUSION MATRIX:
[[ 5620 2185]
 [11097 42999]]
CLASSIFICATION REPORT:

```

	precision	recall	f1-score	support
0	0.34	0.72	0.46	7805
1	0.95	0.79	0.87	54096
accuracy			0.79	61901
macro avg	0.64	0.76	0.66	61901
weighted avg	0.87	0.79	0.81	61901

3). XGBoost Classifier

```
] : classifiers(xgb)

XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
colsample_bynode=1, colsample_bytree=1, eval_metric='logloss',
gamma=0, gpu_id=-1, importance_type='gain',
interaction_constraints='', learning_rate=0.300000012,
max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
monotone_constraints='()', n_estimators=100, n_jobs=8,
num_parallel_tree=1, random_state=0, reg_alpha=0, reg_lambda=1,
scale_pos_weight=1, subsample=1, tree_method='exact',
validate_parameters=1, verbosity=None)

95.41441924352576

cross value score 93.17612183631748
ACCURACY SCORE:
91.65118495662429
ROC AUC SCORE:
78.72836846768342
CONFUSION MATRIX:
[[ 4796 3009]
 [ 2159 51937]]
CLASSIFICATION REPORT:

```

	precision	recall	f1-score	support
0	0.69	0.61	0.65	7805
1	0.95	0.96	0.95	54096
accuracy			0.92	61901
macro avg	0.82	0.79	0.80	61901
weighted avg	0.91	0.92	0.91	61901

4). Random Forest Classifier

```
RandomForestClassifier()
score: 99.99874367650507
F1 score: 94.97291121512549
Accuracy scoer: 91.18592591395938

Cross value score 94.81768924048592
Confusion_matrix:
[[ 4907  2898]
 [ 2558 51538]]
Classification report:
              precision    recall  f1-score   support

     0           0.66       0.63       0.64         7805
     1           0.95       0.95       0.95        54096

 accuracy          0.91         61901
 macro avg          0.80         61901
weighted avg          0.91         61901
```

5). AdaBoost Classifier

```
AdaBoostClassifier()
score: 85.24029280712921
F1 score: 90.26398288945533
Accuracy scoer: 83.89525209608891

Cross value score 85.01877552374847
Confusion_matrix:
[[ 5720  2085]
 [ 7884 46212]]
Classification report:
              precision    recall  f1-score   support

     0           0.42       0.73       0.53         7805
     1           0.96       0.85       0.90        54096

 accuracy          0.84         61901
 macro avg          0.69         61901
weighted avg          0.89         61901
```

6). Gradient Boosting Classifier

```
GradientBoostingClassifier()
score: 90.12027203591411
F1 score: 92.52774588595484
Accuracy scoer: 87.38307943328863

Cross value score 89.71701223790622
Confusion_matrix:
[[ 5736 2069]
 [ 5741 48355]]
Classification report:
      precision    recall  f1-score   support

     0       0.50      0.73      0.59       7805
     1       0.96      0.89      0.93      54096

 accuracy          0.87      61901
 macro avg          0.73      0.81      0.76      61901
 weighted avg          0.90      0.87      0.88      61901
```

7). Bagging Classifier

```
BaggingClassifier()
score: 99.64697309792624
F1 score: 94.0376569037657
Accuracy scoer: 89.68675788759471

Cross value score 93.24854009207438
Confusion_matrix:
[[ 5173 2632]
 [ 3752 50344]]
Classification report:
      precision    recall  f1-score   support

     0       0.58      0.66      0.62       7805
     1       0.95      0.93      0.94      54096

 accuracy          0.90      61901
 macro avg          0.76      0.80      0.78      61901
 weighted avg          0.90      0.90      0.90      61901
```

8). Hyper Parameter Tuning of select best model

Hyperparameter Tuning for select model

```
In [68]: n_estimators=[int(x) for x in np.linspace(start=0, stop=100, num=10)]
max_features=['auto', 'sqrt', 'log2']
criterion=['gini', 'entropy']
max_depth=[2,4]
min_samples_split=[1,2]
bootstrap=[True, False]
```

```
In [69]: para_grid={'n_estimators':n_estimators,
                  'max_features':max_features,
                  'criterion':criterion,
                  'max_depth':max_depth,
                  'min_samples_split':min_samples_split,
                  'bootstrap':bootstrap}
```

```
In [71]: grid=GridSearchCV(RandomForestClassifier(),para_grid,n_jobs=-1)
grid.fit(X_train_sm,Y_train_sm)
grid.best_params_
```

```
Out[71]: {'bootstrap': False,
          'criterion': 'gini',
          'max_depth': 4,
          'max_features': 'sqrt',
          'min_samples_split': 2,
          'n_estimators': 77}
```

9). Result after Hyper parameter tuning

```
In [90]: rf=RandomForestClassifier(bootstrap=False,criterion='gini',max_depth=None,max_features='sqrt',min_samples_split=7,n_estimators=77)
rf.fit(X_train_sm,Y_train_sm)
pred_rf=rf.predict(x_test)
print(accuracy_score(y_test,pred_rf))
print(confusion_matrix(y_test,pred_rf))
print(classification_report(y_test,pred_rf))
```

```
0.9136847546889388
[[ 4704  3101]
 [ 2242 51854]]
      precision    recall  f1-score   support

     0       0.68      0.60      0.64       7805
     1       0.94      0.96      0.95      54096

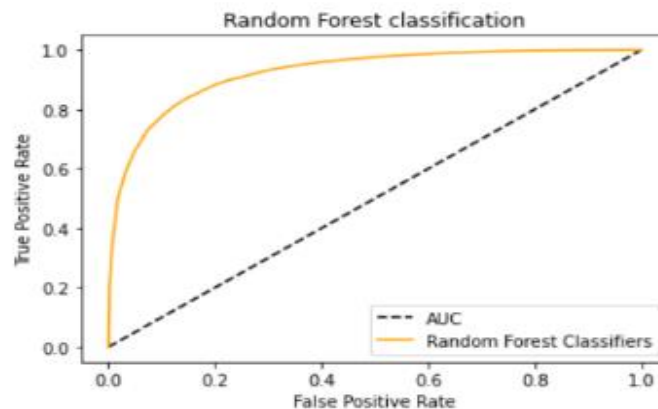
 accuracy          0.91      61901
 macro avg          0.81      0.78      0.79      61901
 weighted avg          0.91      0.91      0.91      61901
```

- I have choosed all parameters of Random Forest Classifier, after tuning the model with best parameters I have increased my model accuracy from 91.18% to 91.36%.

10). AUC ROC SCORE AND GRAPTH PLOT

```
In [115]: #Ploting ROC curve for final model
plt.plot([0,1],[0,1], 'k--', label='AUC')
plt.plot(fpr, tpr, label='Random Forest Classifiers', color='orange',)
print("ROC AUC SCORE: ", roc_auc_score(y_test, pred_rf)*100)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Random Forest classification')
plt.legend()
plt.show()
```

ROC AUC SCORE: 78.06228720772769



11). Saving Model for Prediction

=> I have saved my best model using .obj as follows.

```
In [116]: # Saving best model
import joblib
joblib.dump(rf, 'microcrediloan.obj')
```

Out[116]: ['microcrediloan.obj']

```
In [119]: # Loading the saved model
model=joblib.load("microcrediloan.obj")

#Prediction
prediction = model.predict(x_test)
prediction
```

Out[119]: array([1, 1, 1, ..., 1, 1, 1], dtype=int64)

```
In [122]: pd.set_option('max_columns',6200)
pd.DataFrame([model.predict(x_test)[:],y_test[:]],index=["Predicted","Actual"])
```

Out[122]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
Predicted	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	
Actual	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	

2 rows x 61901 columns

- Above frame is prediction by model of Actual and Predicted values

3.6 Interpretation of the Results

- ✓ The dataset was very challenging to handle it had 37 features with 30days and 90days information of customers.
- ✓ Firstly, the datasets were not having any null values.
- ✓ But there was huge number of zero entries in maximum columns so we have to be careful while going through the statistical analysis of the datasets.
- ✓ And proper plotting for proper type of features will help us to get better insight on the data. I found maximum numerical columns in the dataset so I have chosen bar plot to see the relation between target and features.
- ✓ I notice a huge amount of outliers and skewness in the data so we have choose proper methods to deal with the outliers and skewness. If we ignore this outliers and skewness we may end up with a bad model which has less accuracy.
- ✓ Then scaling dataset has a good impact like it will help the model not to get biased. Since we have not removed outliers and skewness completely from the dataset so we have to choose Normalization.
- ✓ We have to use multiple models while building model using dataset as to get the best model out of it.
- ✓ And we have to use multiple metrics like F1_score, precision, recall and accuracy_score which will help us to decide the best model.
- ✓ I found Random forest Classifier as the best model with 91.18% accuracy_score..
- ✓ At last I have predicted wheather the loan is paid back or not using saved model. It was good!! that I was able to get the predictions near to actual values.

CONCLUSION

4.1 Key Findings and Conclusions of the Study

In this project report, we have used machine learning algorithms to predict the micro credit defaulters. We have mentioned the step by step procedure to analyze the dataset and finding the correlation between the features. Thus we can select the features which are correlated to each other and are independent in nature. These feature set were then given as an input to four algorithms and a hyper parameter tuning was done to the best model and the accuracy has been improved. Hence we calculated the performance of each model using different performance metrics and compared them based on these metrics. Then we have also saved the best model and predicted the label. It was good the the predicted and actual values were almost same.

4.2 Limitations of this work and Scope for Future Work

- ✓ First draw back is the length of the dataset it is very huge and hard to handle.
- ✓ Followed by more number of outliers and skewness these two will reduce our model accuracy.
- ✓ Also, we have tried best to deal with outliers, skewness and zero values. So it looks quite good that we have achieved a accuracy of 91.32% even after dealing all these drawbacks.
- ✓ Also, this study will not cover all Classification algorithms instead, it is focused on the chosen algorithm, starting from the basic ensembling techniques to the advanced ones.