FLIP ROBO

Project Report On

Micro Credit Defaulter Project

Submitted By: OLIVER RAMAN | Internship 26

ACKNOWLEDGEMENT

The following research papers helped me understand micro credit, the various factors affecting micro credit, different types of Microfinance Institutions (MFI) and their services The Microfinance services (MFS) provided by MFI are Group Loans, Agricultural Loans, Individual Business Loans and so on & finally, helped me in my model building & predictions:

1."Predicting Credit Default among Micro Borrowers in Ghana"- Kwame Simpe Ofori, Eli Fianu, Osaretin Kayode Omoregie, Nii Afotey Odai

Microfinance institutions play a major role in economic development in many developing countries. However many of these microfinance institutions are faced with the problem of default because of the non-formal nature of the business and individuals they lend money to. This study seeks to find the determinants of credit default in microfinance institutions. With data on 2631 successful loan applicants from a microfinance institution with braches all over the country we proposed a Binary logistic regression model to predict the probability of default. We found the following variables significant in determining default: Age, Gender, Marital Status, Income Level, Residential Status, Number of Dependents, Loan Amount, and Tenure. We also found default to be more among the younger generation and in males. We however found Loan Purpose not to be significant in determining credit default. Microfinance institutions could use this model to screen prospective loan applicants in order to reduce the level of default.

2. “A Machine Learning Approach for Micro-Credit Scoring” - Apostolos Ampountolas, Titus Nyarko Nde, Paresh Date, Corina Constantinescu

: In micro-lending markets, lack of recorded credit history is a significant impediment to assessing individual borrowers’ creditworthiness and therefore deciding fair interest rates. This research compares various machine learning algorithms on real micro-lending data to test their efficacy at classifying borrowers into various credit categories. We demonstrate that off-the-shelf multi-class classifiers such as random forest algorithms can perform this task very well, using readily available data about customers (such as age, occupation, and location). This presents inexpensive and reliable means to micro-lending institutions around the developing world with which to assess creditworthiness in the absence of credit history or central credit databases.

3. “What Happens to Microfinance Clients who Default?” - Jami Solli, Laura Galindo, Alex Rizzi, Elisabeth Rhyne, Nadia van de Walle

These are often the ways lenders characterize non-paying clients as they dispatch collections officers to do whatever they can to get the money back. Lenders with conscience find this part of the business unsettling, but unavoidable. Many a lender who started out softhearted soon realized that having flexible, or ‘soft’ strategies led to rising risk. The loss of any single loan is a small threat, but many defaults can destroy the business. When word gets out that a lender is lenient, mass default can infect the whole market. We've all seen it happen. The defaulter's perspective may be completely different. Most serious defaulters are in financial distress and often in the midst of other life crises. A sick child, spouse, or parent needs expensive medical treatment. Another borrower's entire crop was lost due to drought. Another elderly borrower was robbed of her goods while exiting from a two-hour bus ride to the market.

4. "Rural Micro Credit Assessment using Machine Learning in a Peruvian microfinance institution" - Henry IvanCondori-Alejo, Miguel RomilioAceituno-Rojo, Guina Sotomayor Alzamora

Microcredits are an important component in the development of the peruvian rural economy, which are granted by microfinance institutions, the assessment process for the rural an poor people has a high risk index which is traditionally controlled by the business rural advisor, whose main tasks are the evaluation and verification of the clients requesting these microcredits. This research proposes a model that presents the best level of assertiveness for microcredits assessment process based on determination analysis of rural variables based on the specialized literature in the area. This model serves as a decision-support tool for business rural advisor in order to reduce the credit risk of the rural microfinance institution, The most representative variables of the financial and microfinance segment have been evaluated; the data has been pre-processed; Machine Learning models have been selected, trained, validated and evaluated through different metrics. The most assertive model in the assessment process of the granting of rural microcredit, based on the variables and data used by the analyzed entity, are: the Artificial Neural Network (93.72), on Logistic Regression (86.07), Random Forest (66.35), Support Vector Machine (84.44), Decision Tree (88.80) and k-Nearest Neighbor (65.98). Finally, the level of assertiveness achieved by ANN model 93.72% is better than the entity traditional methodology 76.81%, showing an improvement of 16.91% in the index of default customers.

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.

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

Conceptual Background of the Domain Problem

Microfinance is a banking service provided to unemployed or low-income individuals or groups who otherwise would have no other access to financial services. Indonesia is renowned for its large-scale microfinance sector, with a range of commercial banks. Some rural communities in Indonesia have no choice but to seek out loans from unregulated moneylenders.

Micro lenders, particularly those operating under Indonesian banks, as well as social enterprise start-ups, are also targeting these communities through their high mobile penetration rates and are developing the right digital platforms to reach out to them.

Generally, Credit Scores play a vital role for loan approvals, and is very important in today's financial analysis for an individual. Most of the loan lending vendors rely heavily on it, so in our case users has 5 days' time to pay back the loan or else they are listed as defaulters which will impact the loan the credit score heavily, so there are few thing to lookout in this dataset as users who are taking extensive loans, user who have most frequent recharges in their main account have a good chance of 100% payback rate, and user who never recharged their main account for them loan should have never been approved as there is high chance for single user or default user taking multiple connections in name or documents of the family members

Review of Literature

Provenzano et al. (2020) introduced machine learning models to compose credit rating and default prediction estimation. They used financial instruments, such as historical balance sheets, bankruptcy statutes, and macroeconomic variables of a Moody's dataset. Using machine learning models, the authors observed excellent out-of-sample performance results to reduce the bankruptcy probability or improve credit rating.

Petropoulos et al. (2019) studied a dataset of loan-level data of the Greek economy examining credit quality performance and quantification of probability default for an evaluating period of 10 years. The authors used an extended example of classifications of the incorporated machine learning models against traditional methods, such as logistic regression. Their results identified that machine learning models had demonstrated superior performance and forecasting accuracy through the financial credit rating cycle.

In Addo et al. (2018) the authors examined credit risk scoring by employing various machine and deep learning techniques. The authors used binary classifiers in modeling loan default probability (DP) estimations by incorporating ten key features to test the classifiers' stability by evaluating performance on separate data. Their results indicated that the models such as the logistic regression, random forest, and gradient boosting modeling generated more accurate results than the models based on the neural network approach incorporating various technicalities.

Zhao et al. (2015) examined a multi-layer perceptron (MLP) neural network's accuracy regarding estimating credit scores efficiently. The authors used a German credit dataset to train and estimate the model's accuracy. Their results indicated an MLP model containing nine hidden units achieved a classification accuracy of 87%, higher than other similar experiments. Their study results proved the trend of MLP models' scoring accuracy by increasing the number of hidden units.

Yap et al. (2011) used historical payment data from a recreational club and established credit scoring techniques to identify potential club member subscription defaulters. The study results demonstrated that no model outperforms the others among a credit scorecard model, logistic regression, and a decision tree model. Each model generated almost identical accuracy figures.

Motivation for the Problem Undertaken

The main objective of this study is to investigate which method from a chosen set of machine learning techniques performs the best default prediction. This project was highly motivated project as it includes the real time problem for Microfinance Institution (MFI), and to the poor families in remote areas with low income, and it is related to financial sectors, as I believe that with growing technologies and Idea can make a difference, there are so much in the financial market to explore and analyze and with Data Science the financial world becomes more interesting.

The project gives an insight to identify major factors that lead to credit risk portfolio in microfinance banks and provide recommendations aimed at mitigating credit risks in microfinance banks. With the help of independent variables available in the dataset we need to model the micro credit defaulters' level in the micro finance institution. This model will help the management to understand how the users are considered as defaulter or non-defaulter based on the attributes available.

ANALYTICAL PROBLEM FRAMING

Mathematical/Analytical Modeling of the Problem

We need to build a Machine Learning 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 the dataset, the 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.

Clearly, it is a binary classification problem where we need to use classification algorithms to predict the results. There were no null values in the dataset. There were some unwanted entries like more than 90% of zero values present in some of the columns which means these customers have no loan history so, I have dropped those columns. I found some negative values while summarizing the statistics of the dataset, I have converted them into positive. To get better insights on features I have used some plots like pie plot, count plot, bar plot, distribution plot, box plots etc. There were lots of skewness and outliers present in our dataset which needed to be cleaned using appropriate techniques and balanced the data. At last, I have built many classification models to predict the defaulter level at the institution

Data Sources & 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 had 209593 rows and 36 columns including target. In this particular dataset 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

Data preprocessing done

- Importing necessary libraries and loading the dataset as a data frame.
- Used pandas to display maximum columns ensuring not to find any truncated information.
- Checked some statistical information like shape, number of unique values present, info, finding zero values etc.
- Checked for null values and did not find any null values.
- Dropped some unwanted columns like Unnamed:0, pcircle, msisdn as they are of no use for prediction.
- Dealt with zero values by verifying the percentage of zero values in each column and decided to discard the columns having more than 90% of zero values.
- Converted the time variable “pdate” from object into datetime and extracted Day, Month and Year for better understanding. Checked value counts for each and dropped Year column as it contains unique value throughout the dataset.
- Checked unique values and value counts of target variables.
- Converted the data having values other than 6, 12 & 0 into 0 in the column maxamnt_loans30. As it is specified in the problem statement that we should have only 0, 6 & 12 values. Also, discarded some rows in the column amnt_loans90 as it gives the sum of loans taken by the user in 90 days
- While checking the statistical summary of the dataset, I found some columns having negative values which were invalid and unrealistic so I decided to convert negative values into positive using absolute command.
- Visualized each feature using seaborn and matplotlib libraries by plotting several categorical and numerical plots like pie plot, count plot, bar plot, distribution plot, box plots etc.

- Identified outliers using box plots and I tried to remove them using both Z Score and IQR method and got huge data loss of around 18% and 62% respectively, so removed outliers using percentile method.
- Checked for skewness and removed skewness in numerical columns using power transformation method (yeo-johnson).
- Used Pearson's correlation coefficient to check the correlation between label and features. With the help of heatmap, correlation bar graph was able to understand the Feature vs Label relativity and insights on multicollinearity amongst the feature columns.
- Separated feature and label data and feature scaling is performed using the MinMaxScalar method to avoid any kind of data biases.
- Since the dataset was imbalanced. Label '1' had approximately 87.5% records, while label '0' had approximately 12.5% records. So, I performed Oversampling using SMOTE to balance the data.
- Checked for the best random state to be used on our Machine Learning model pertaining to the feature importance details.
- Finally created a classification model along with evaluation metrics.

Data Inputs- Logic- Output Relationships

The dataset consists of a label and features. The features are independent and the label is dependent as the values of our independent variables change as our label varies.

- Since we had only numeric columns, I checked the distribution of skewness using dist plots as a part of univariate analysis.
- To analyze the relation between features and label I have used many plotting techniques where I found some of the columns having strong relation with label.
- The visualization helped me to understand that maximum distribution is for non-defaulter for all the features & maximum defaulter list are from people who have Average payback time in days over last 30 & 90 days, also frequency of recharge done in the main account since last 90 days. So, the features, which I have kept after dropping, had some kind of relationship with the output.
- I have checked the correlation between the label and features using heat map and bar plot. Where I got the positive correlation between the label and features and there was not much relation.

Hardware and Software Requirements & Tools used

To build the machine learning projects it is important to have the following hardware and software requirements and tools.

Hardware required:

- Processor: core i5 or above
- RAM: 8 GB or above
- ROM/SSD: 250 GB or above

Software required:

- Anaconda - language used Python 3

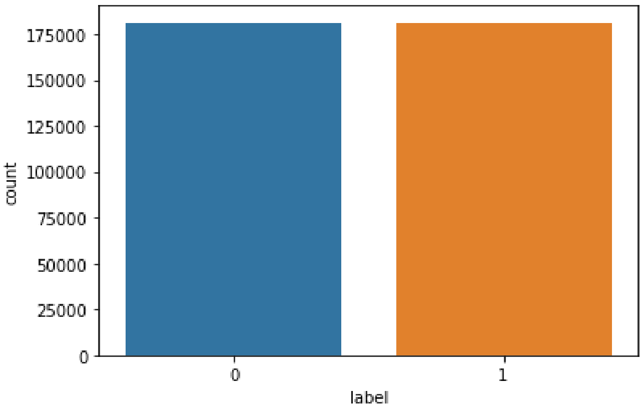
Libraries Used:

- import pandas as pd
- import numpy as np
- import seaborn as sns
- import matplotlib.pyplot as plt
- import warnings
  warnings.filterwarnings('ignore')
- from scipy import stats
- from scipy.stats import z score
- from sklearn.preprocessing import PowerTransformer
  scaler = PowerTransformer(method='yeo-johnson')
- from sklearn.preprocessing import MinMaxScaler
- from imblearn.over_sampling import SMOTE
- from sklearn.metrics import accuracy_score
- from sklearn.metrics import confusion_matrix,classification_report
- from sklearn.model_selection import train_test_split
- from sklearn.preprocessing import StandardScaler
- from sklearn.linear_model import LogisticRegression
- from sklearn.metrics import accuracy_score
- from sklearn.metrics import confusion_matrix,classification_report
- from sklearn.tree import DecisionTreeClassifier
- from sklearn.ensemble import RandomForestClassifier
- from sklearn.ensemble import ExtraTreesClassifier
- from sklearn.ensemble import BaggingClassifier
- from xgboost import XGBClassifier
- from sklearn.model_selection import cross_val_score
- from sklearn.model_selection import GridSearchCV
- from sklearn.metrics import plot_roc_curve

MODEL BUILDING & EVALUATION

Identification of possible problem-solving approaches (methods)

I have used both statistical and analytical approaches to solve the problem which mainly includes the pre-processing of the data and also used EDA techniques and heat map to check the correlation of independent and dependent features. Also, before building the model, I made sure that the input data was cleaned and scaled before it was fed into the machine learning models. The data mainly had class imbalancing issues which look like below.



From the above we can see that the data set is highly imbalanced, so applied the SMOTE method to balance the dataset.

For this particular project we need to predict whether the user paid back the credit loan amount within 5 days of issuing the loan. In this dataset, the label is the target variable, which consists of two categories, defaulters and non-defaulters. Which means our target column is categorical in nature so this is a classification problem.

I have used many classification algorithms and got the prediction results. By doing various evaluations I have selected Extra Trees Classifier as the best suitable algorithm to create our final model as it is giving the least difference in accuracy score and cross validation score among all the algorithms used.

In order to get good performance and to check whether my model is getting overfitting and under-fitting I have made use of the cross validation and then hyper parameter tuning on the best model. Then I saved my final model and loaded the same for predictions.

Testing of Identified Approaches (Algorithms)

Since the label is my target variable which is categorical in nature, from this I can conclude that it is a classification type problem hence I have used the following classification algorithms.

After the pre-processing and data cleaning I left with 27 columns including target and with the help of feature importance bar graph I used these independent features for model building and prediction.

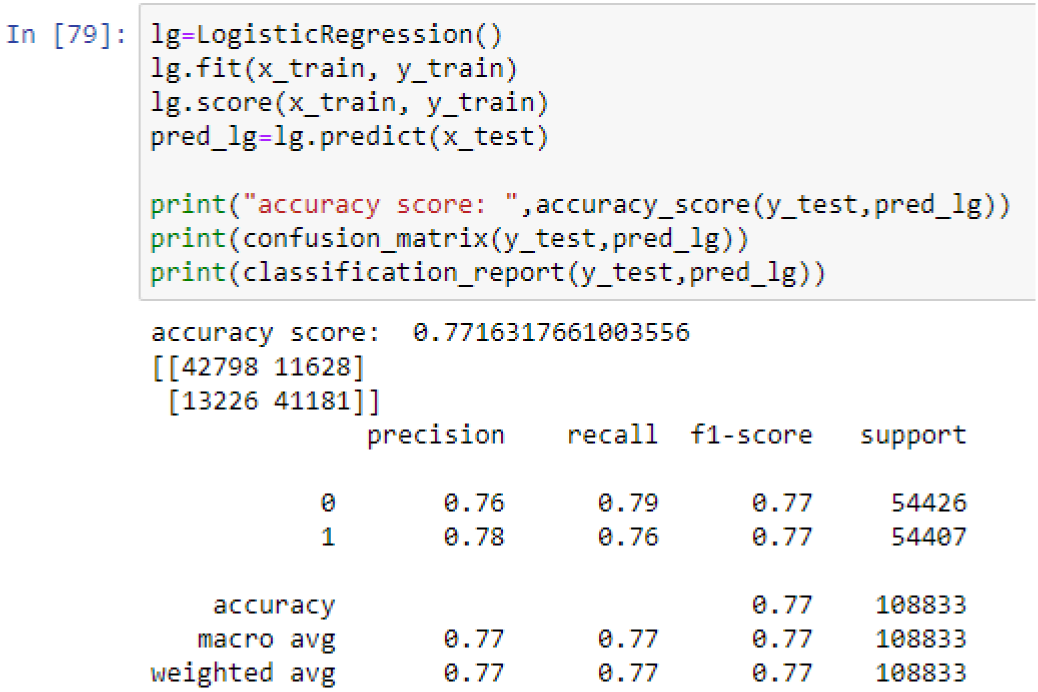
The algorithms used on training the data are as follows:

- Logistic Regression Model
- Decision Tree Classifier
- Random Forest Classifier
- Extra Trees Classifier
- Bagging Classifier
- XGBoost Classifier

Run and Evaluate selected models

<u>Classification Models:</u>

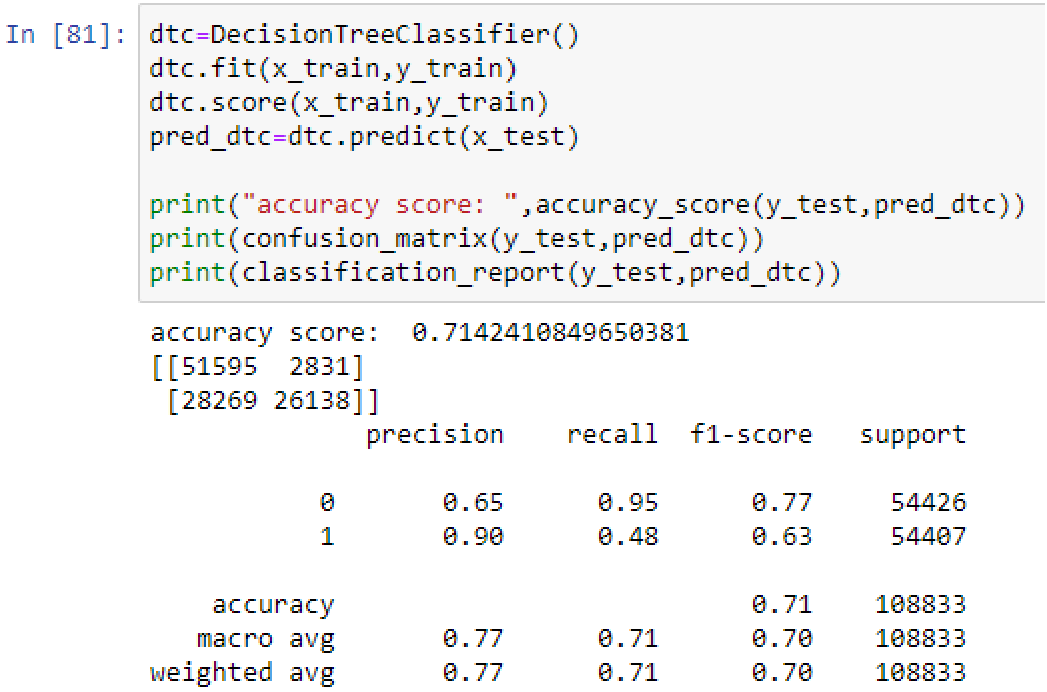
LOGISTIC REGRESSION:



```
In [79]: lg=LogisticRegression()
         lg.fit(x_train, y_train)
         lg.score(x_train, y_train)
         pred_lg=lg.predict(x_test)

         print("accuracy score: ",accuracy_score(y_test,pred_lg))
         print(confusion_matrix(y_test,pred_lg))
         print(classification_report(y_test,pred_lg))

         accuracy score:  0.7716317661003556
         [[42798 11628]
          [13226 41181]]
                       precision    recall  f1-score   support

                    0       0.76      0.79      0.77     54426
                    1       0.78      0.76      0.77     54407

             accuracy                           0.77    108833
            macro avg       0.77      0.77      0.77    108833
         weighted avg       0.77      0.77      0.77    108833
```

The Logistic Regression model gave us an Accuracy Score of 77.16 %.

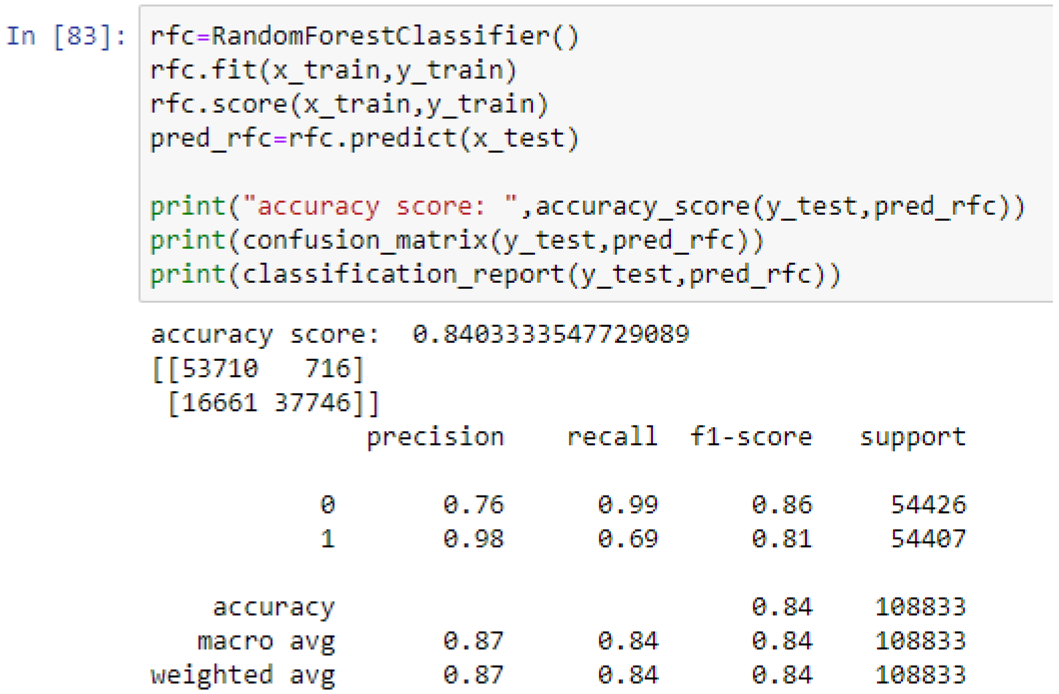
DECISION TREE CLASSIFIER:



```
In [81]: dtc=DecisionTreeClassifier()
         dtc.fit(x_train,y_train)
         dtc.score(x_train,y_train)
         pred_dtc=dtc.predict(x_test)

         print("accuracy score: ",accuracy_score(y_test,pred_dtc))
         print(confusion_matrix(y_test,pred_dtc))
         print(classification_report(y_test,pred_dtc))

         accuracy score:  0.7142410849650381
         [[51595  2831]
          [28269 26138]]
                       precision    recall  f1-score   support

                    0       0.65      0.95      0.77     54426
                    1       0.90      0.48      0.63     54407

             accuracy                           0.71    108833
            macro avg       0.77      0.71      0.70    108833
         weighted avg       0.77      0.71      0.70    108833
```

The Decision Tree Classifier Model gave us an Accuracy Score of 71.42 %.

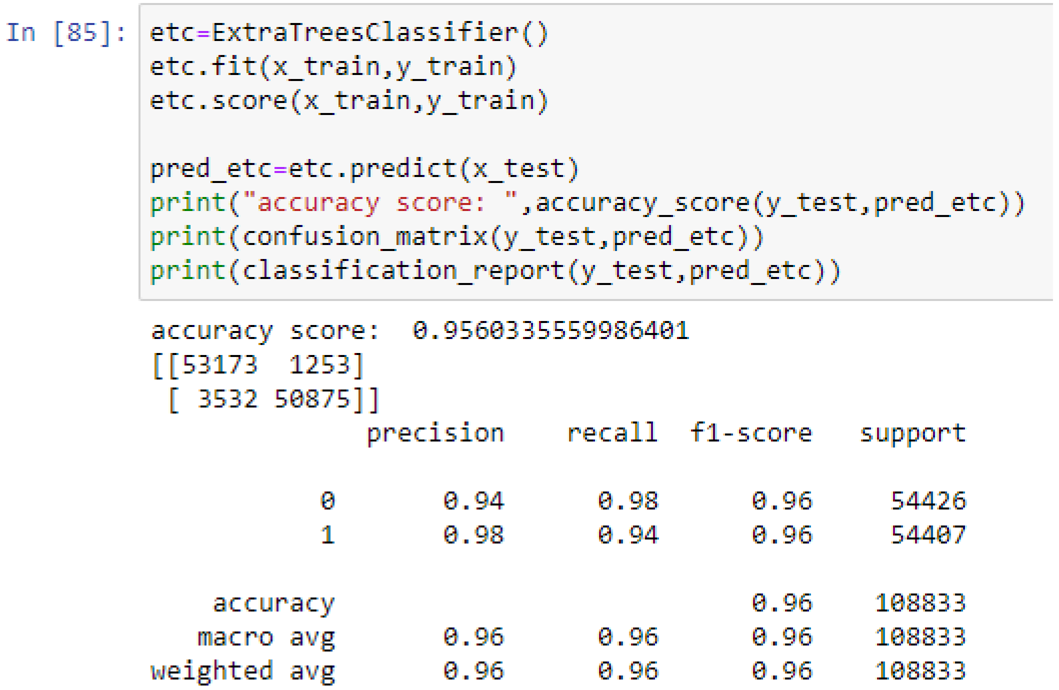
RANDOM FOREST CLASSIFIER:



```
In [83]: rfc=RandomForestClassifier()
         rfc.fit(x_train,y_train)
         rfc.score(x_train,y_train)
         pred_rfc=rfc.predict(x_test)

         print("accuracy score: ",accuracy_score(y_test,pred_rfc))
         print(confusion_matrix(y_test,pred_rfc))
         print(classification_report(y_test,pred_rfc))

         accuracy score:  0.8403333547729089
         [[53710   716]
          [16661 37746]]
                       precision    recall  f1-score   support

                    0       0.76      0.99      0.86     54426
                    1       0.98      0.69      0.81     54407

             accuracy                           0.84    108833
            macro avg       0.87      0.84      0.84    108833
         weighted avg       0.87      0.84      0.84    108833
```

The Random Forest Classifier model gave us an Accuracy Score of 84.03 %.

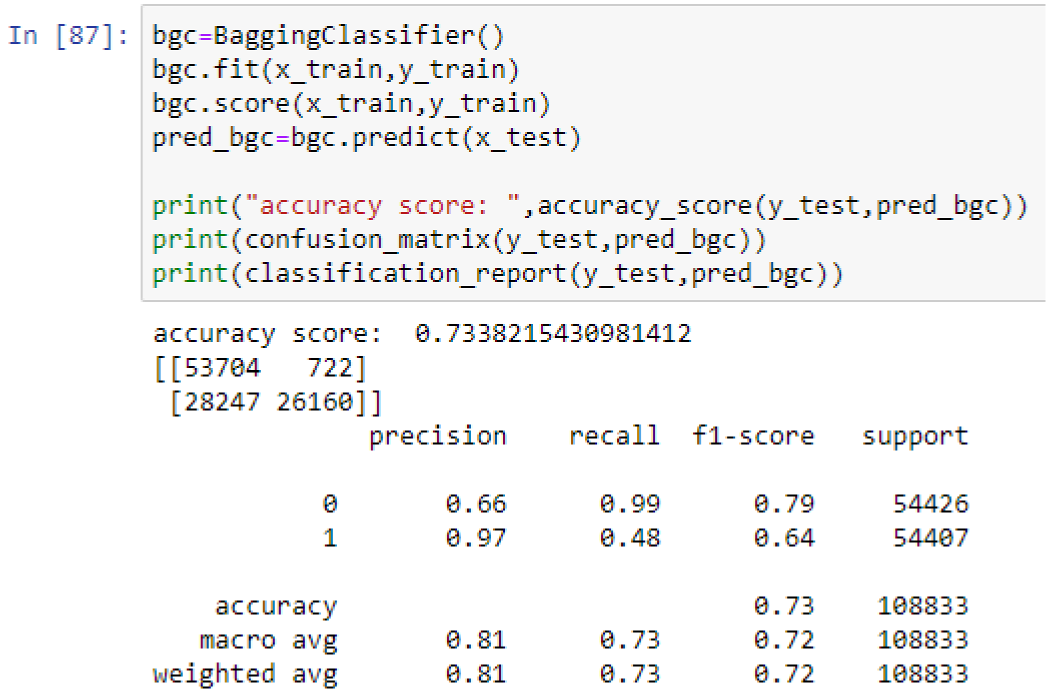
EXTRA TREES CLASSIFIER:



```
In [85]: etc=ExtraTreesClassifier()
         etc.fit(x_train,y_train)
         etc.score(x_train,y_train)

         pred_etc=etc.predict(x_test)
         print("accuracy score: ",accuracy_score(y_test,pred_etc))
         print(confusion_matrix(y_test,pred_etc))
         print(classification_report(y_test,pred_etc))

         accuracy score:  0.9560335559986401
         [[53173  1253]
          [ 3532 50875]]
                       precision    recall  f1-score   support

                    0       0.94      0.98      0.96     54426
                    1       0.98      0.94      0.96     54407

             accuracy                           0.96    108833
            macro avg       0.96      0.96      0.96    108833
         weighted avg       0.96      0.96      0.96    108833
```

The Extra Trees Classifier model gave us an Accuracy Score of 95.60 %.

BAGGING CLASSIFIER:

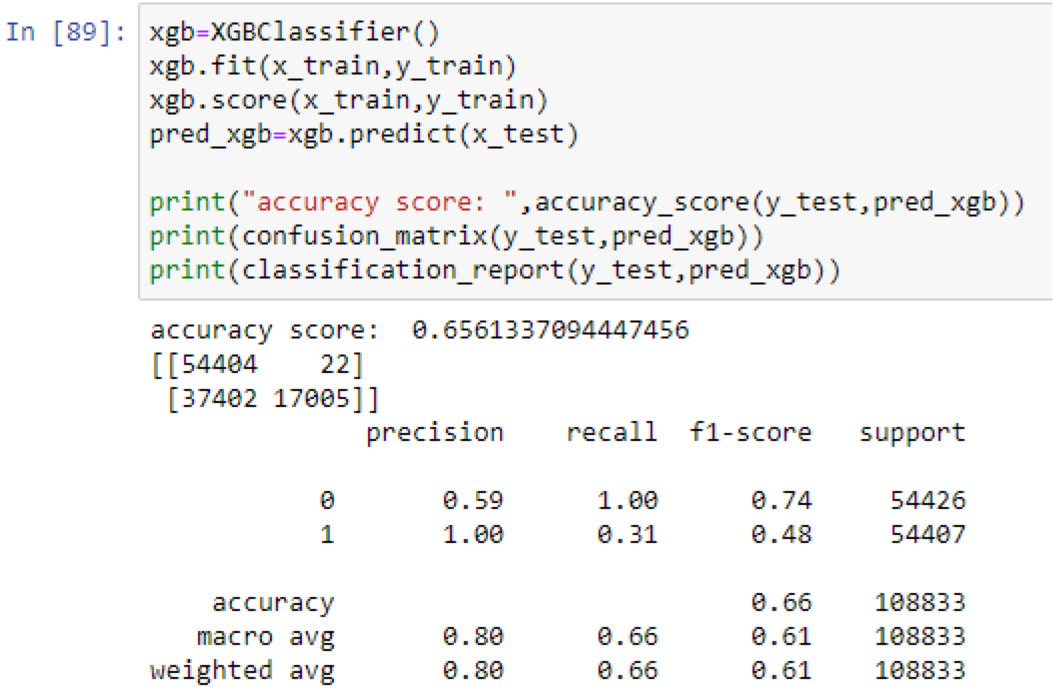


```
In [87]: bgc=BaggingClassifier()
         bgc.fit(x_train,y_train)
         bgc.score(x_train,y_train)
         pred_bgc=bgc.predict(x_test)

         print("accuracy score: ",accuracy_score(y_test,pred_bgc))
         print(confusion_matrix(y_test,pred_bgc))
         print(classification_report(y_test,pred_bgc))
```

```
accuracy score:  0.7338215430981412
[[53704   722]
 [28247 26160]]
              precision    recall  f1-score   support

           0       0.66      0.99      0.79     54426
           1       0.97      0.48      0.64     54407

    accuracy                           0.73    108833
   macro avg       0.81      0.73      0.72    108833
weighted avg       0.81      0.73      0.72    108833
```

The Bagging Classifier model gave us an Accuracy Score of 73.38 %.

XGBOOST CLASSIFIER:



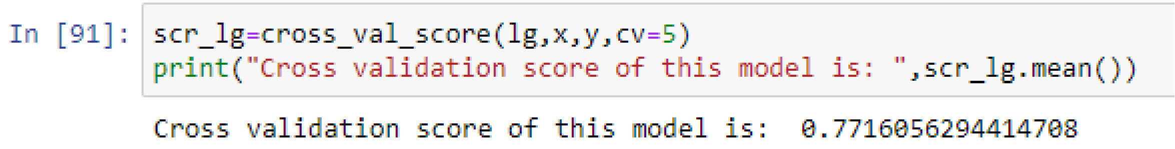
```
In [89]: xgb=XGBClassifier()
         xgb.fit(x_train,y_train)
         xgb.score(x_train,y_train)
         pred_xgb=xgb.predict(x_test)

         print("accuracy score: ",accuracy_score(y_test,pred_xgb))
         print(confusion_matrix(y_test,pred_xgb))
         print(classification_report(y_test,pred_xgb))
```

```
accuracy score:  0.6561337094447456
[[54404    22]
 [37402 17005]]
              precision    recall  f1-score   support

           0       0.59      1.00      0.74     54426
           1       1.00      0.31      0.48     54407

    accuracy                           0.66    108833
   macro avg       0.80      0.66      0.61    108833
weighted avg       0.80      0.66      0.61    108833
```

The XGBoost Classifier model gave us an Accuracy Score of 65.61 %.

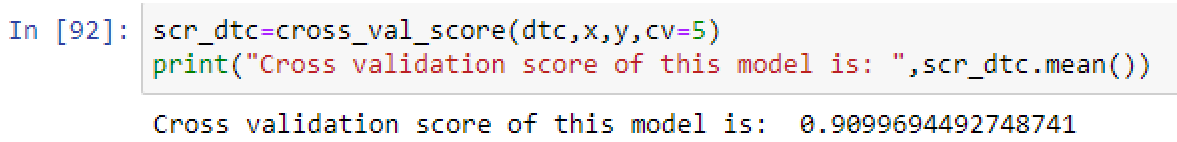
From the above Classification Models, the highest accuracy score belongs to Extra Trees Classifier.
Next, the Random Forest Classifier, followed by Logistic Regression Model & Bagging Classifier.
After that, the Decision Tree Classifier. Lastly, the XGBoost Classifier.

Cross Validation Scores:



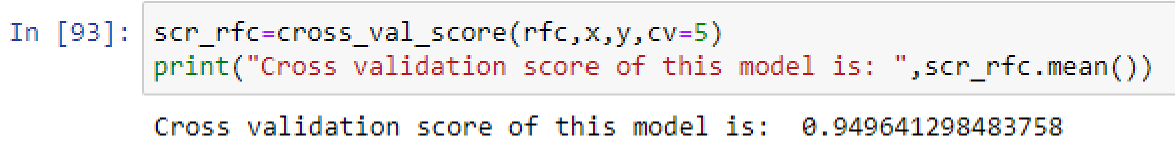
```
In [91]: scr_lg=cross_val_score(lg,x,y,cv=5)
         print("Cross validation score of this model is: ",scr_lg.mean())

         Cross validation score of this model is:  0.7716056294414708
```

The cross validation score of the Logistic Regression Model is 77.16 %.



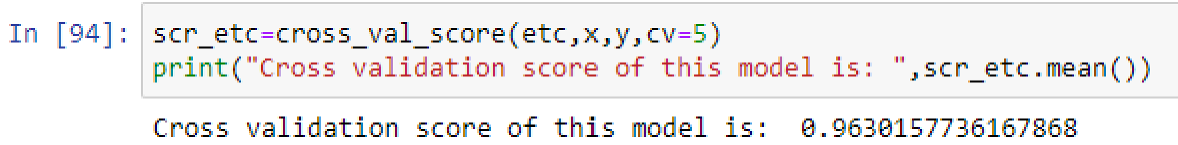
```
In [92]: scr_dtc=cross_val_score(dtc,x,y,cv=5)
         print("Cross validation score of this model is: ",scr_dtc.mean())

         Cross validation score of this model is:  0.9099694492748741
```

The cross validation score of the Decision Tree Classifier Model is 90.99 %.



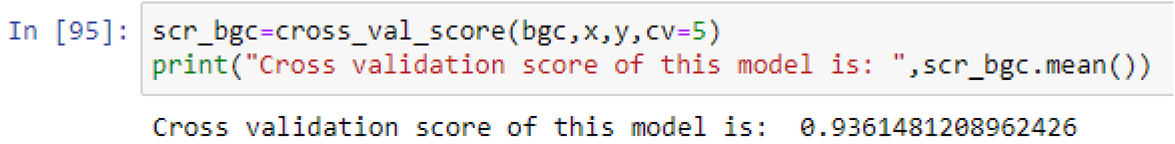
```
In [93]: scr_rfc=cross_val_score(rfc,x,y,cv=5)
         print("Cross validation score of this model is: ",scr_rfc.mean())

         Cross validation score of this model is:  0.949641298483758
```

The cross validation score of the Random Forest Classifier Model is 94.96 %.



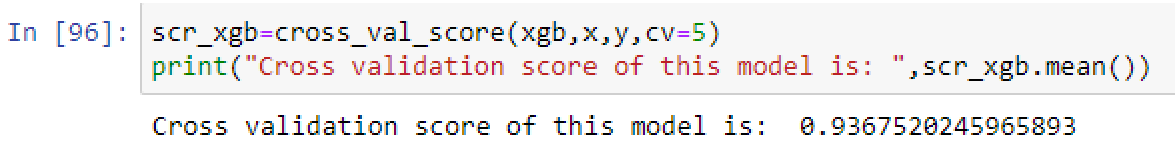
```
In [94]: scr_etc=cross_val_score(etc,x,y,cv=5)
         print("Cross validation score of this model is: ",scr_etc.mean())

         Cross validation score of this model is:  0.9630157736167868
```

The cross validation score of the Extra Trees Classifier Model is 96.60 %.



```
In [95]: scr_bgc=cross_val_score(bgc,x,y,cv=5)
         print("Cross validation score of this model is: ",scr_bgc.mean())

         Cross validation score of this model is:  0.9361481208962426
```

The cross validation score of the Bagging Classifier Model is 93.61 %.



```
In [96]: scr_xgb=cross_val_score(xgb,x,y,cv=5)
         print("Cross validation score of this model is: ",scr_xgb.mean())

         Cross validation score of this model is:  0.9367520245965893
```

The cross validation score of the XGBoost Classifier Model is 93.67 %.

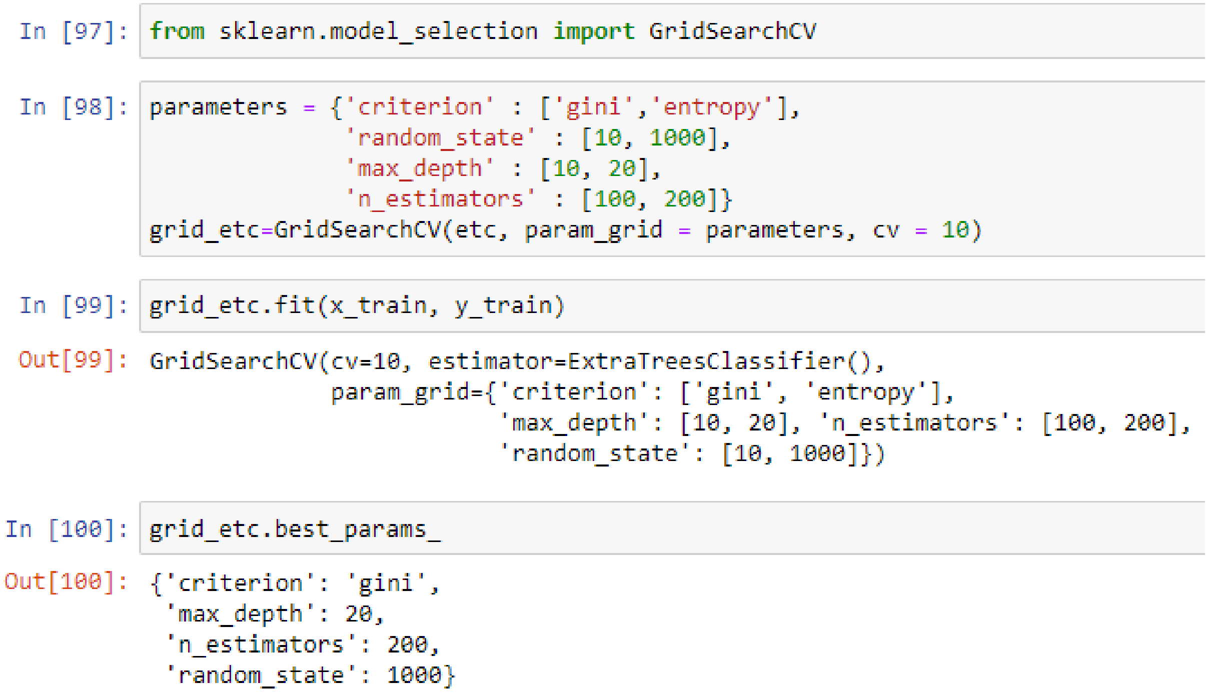
The highest cross validation score belongs to the Extra Trees Classifier, followed by the Random Forest Classifier. Next the XGBoost Classifier and the Bagging Classifier.

After that, the Decision Tree Classifier. Lastly, the lowest Cross Validation Score belongs to the Logistic Regression Model.

Hyper Parameter Tuning:

Since the Accuracy score and the cross validation score of the Extra Trees Classifier are the highest we shall consider it for hyper parameter tuning.

We shall use GridSearchCV for Hyper Parameter Tuning.

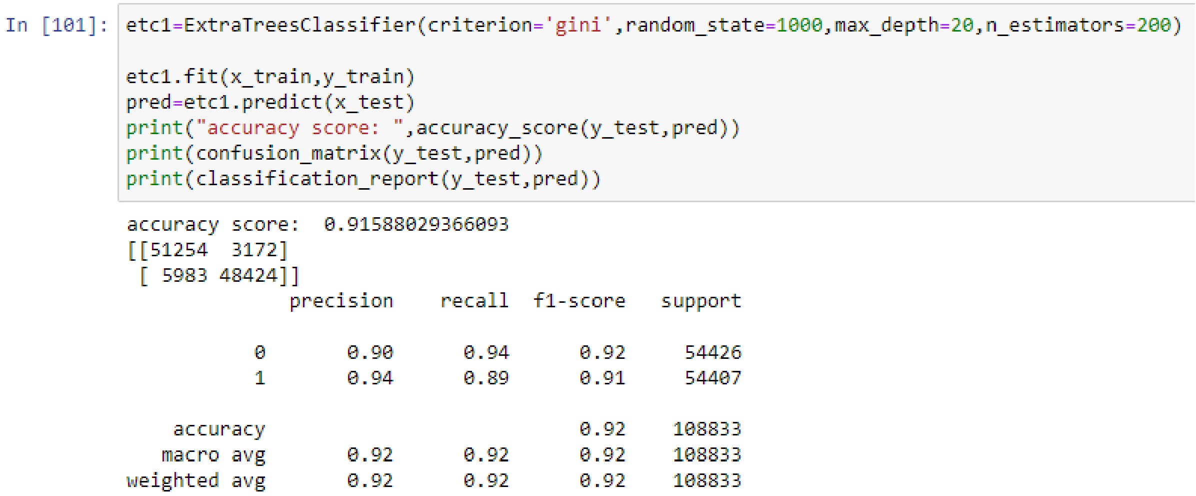


```
In [97]: from sklearn.model_selection import GridSearchCV
```

```
In [98]: parameters = {'criterion' : ['gini','entropy'],
                       'random_state' : [10, 1000],
                       'max_depth' : [10, 20],
                       'n_estimators' : [100, 200]}
         grid_etc=GridSearchCV(etc, param_grid = parameters, cv = 10)
```

```
In [99]: grid_etc.fit(x_train, y_train)

Out[99]: GridSearchCV(cv=10, estimator=ExtraTreesClassifier(),
                      param_grid={'criterion': ['gini', 'entropy'],
                                  'max_depth': [10, 20], 'n_estimators': [100, 200],
                                  'random_state': [10, 1000]})
```

```
In [100]: grid_etc.best_params_

Out[100]: {'criterion': 'gini',
           'max_depth': 20,
           'n_estimators': 200,
           'random_state': 1000}
```

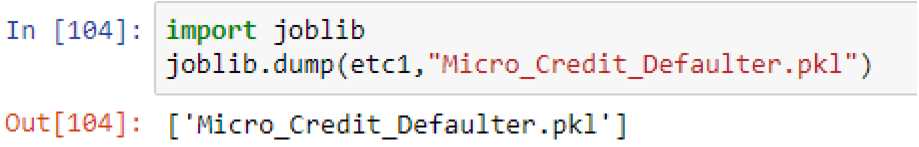


```
In [101]: etc1=ExtraTreesClassifier(criterion='gini',random_state=1000,max_depth=20,n_estimators=200)

          etc1.fit(x_train,y_train)
          pred=etc1.predict(x_test)
          print("accuracy score: ",accuracy_score(y_test,pred))
          print(confusion_matrix(y_test,pred))
          print(classification_report(y_test,pred))

          accuracy score:  0.91588029366093
          [[51254  3172]
           [ 5983 48424]]
                        precision    recall  f1-score   support

                     0       0.90      0.94      0.92     54426
                     1       0.94      0.89      0.91     54407

              accuracy                           0.92    108833
             macro avg       0.92      0.92      0.92    108833
          weighted avg       0.92      0.92      0.92    108833
```

After Hyper Parameter Tuning, we have got an accuracy score of 91.58 %.

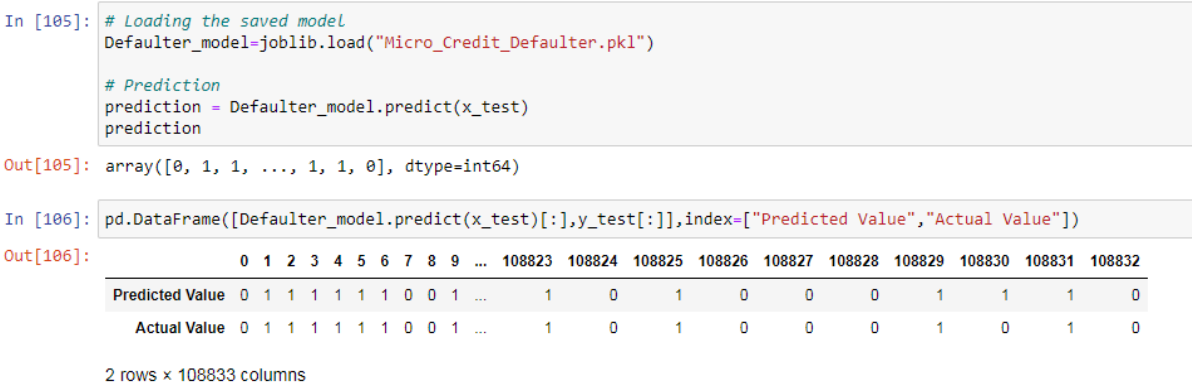
Saving the final model and predicting the saved model

Now we shall save the best model.



```
In [104]: import joblib
          joblib.dump(etc1,"Micro_Credit_Defaulter.pkl")

Out[104]: ['Micro_Credit_Defaulter.pkl']
```

Predictions from the saved model.



```
In [105]: # Loading the saved model
          Defaulter_model=joblib.load("Micro_Credit_Defaulter.pkl")

          # Prediction
          prediction = Defaulter_model.predict(x_test)
          prediction

Out[105]: array([0, 1, 1, ..., 1, 1, 0], dtype=int64)

In [106]: pd.DataFrame([Defaulter_model.predict(x_test)[:],y_test[:]],index=["Predicted Value","Actual Value"])

Out[106]:
```

	0	1	2	3	4	5	6	7	8	9	...	108823	108824	108825	108826	108827	108828	108829	108830	108831	108832
Predicted Value	0	1	1	1	1	1	1	0	0	1	...	1	0	1	0	0	0	1	1	1	0
Actual Value	0	1	1	1	1	1	1	0	0	1	...	1	0	1	0	0	0	1	0	1	0

2 rows × 108833 columns

The actual and predicted values are almost similar.

Key Metrics for success in solving problem under consideration

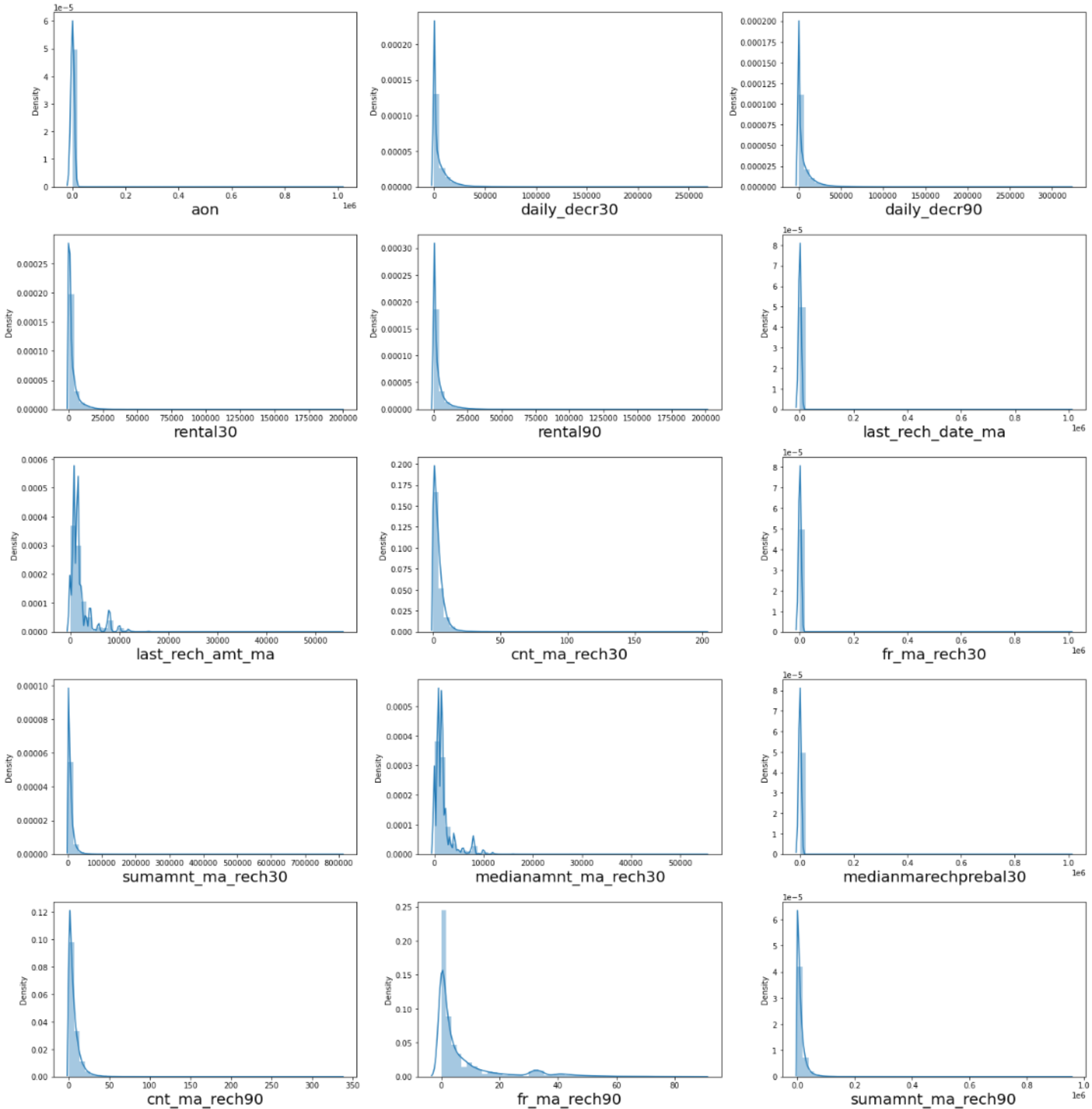
I have used the following metrics for evaluation:

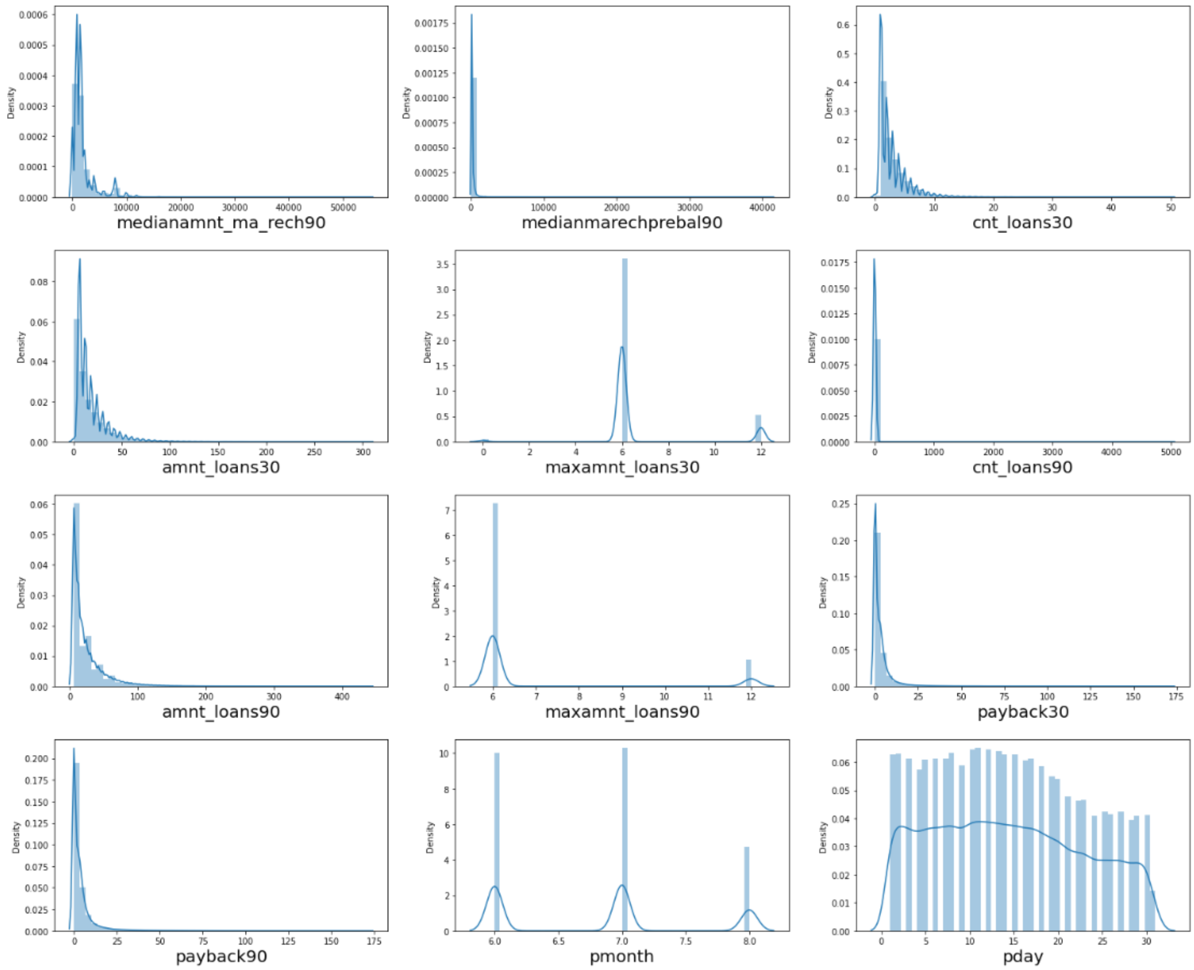
- Accuracy score, which is used when the True Positives and True negatives are more important.
- Confusion Matrix, which is a matrix used to determine the performance of the classification models for a given set of test data.
- Classification report, which is used to measure the quality of predictions from a classification algorithm.
- Cross Validation Score, which is a technique for evaluating ML models by training several ML models on subsets of the available input data and evaluating them on the complementary subset of the data.
- Precision is the degree to which repeated measurements under the same conditions are unchanged. It is the amount of information that is conveyed by a value. It refers to the data that is correctly classified by the classification algorithm.
- Recall is how many of the true positives were recalled (found). Recall refers to the percentage of data that is relevant to the class.
- F1 Score is used to express the performance of the machine learning model (or classifier). It gives the combined information about the precision and recall of a model. This means a high F1-score indicates a high value for both recall and precision.
- Roc Auc Curve, the Receiver Operator Characteristic (ROC) curve, is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values. The Area Under Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

Visualizations

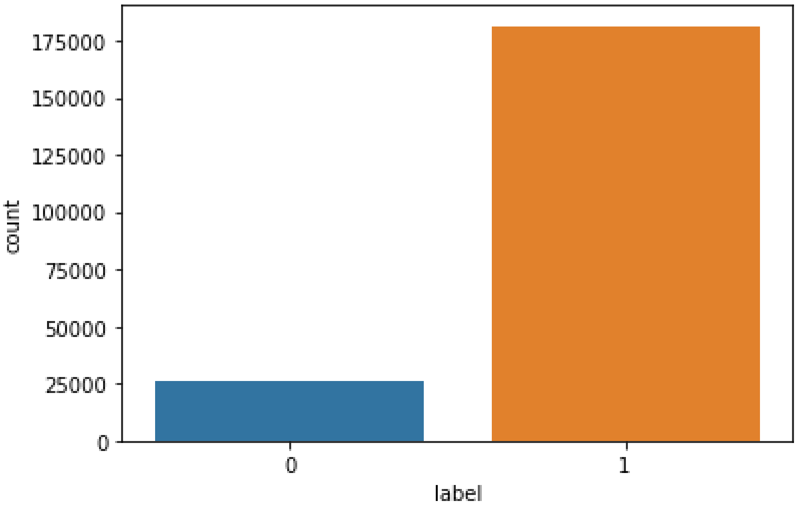
I have used bar plots to see the relation of numerical features with target and I have used 2 types of plots for numerical columns one is disp plot for univariate and bar plot for bivariate analysis.

UNIVARIATE ANALYSIS:



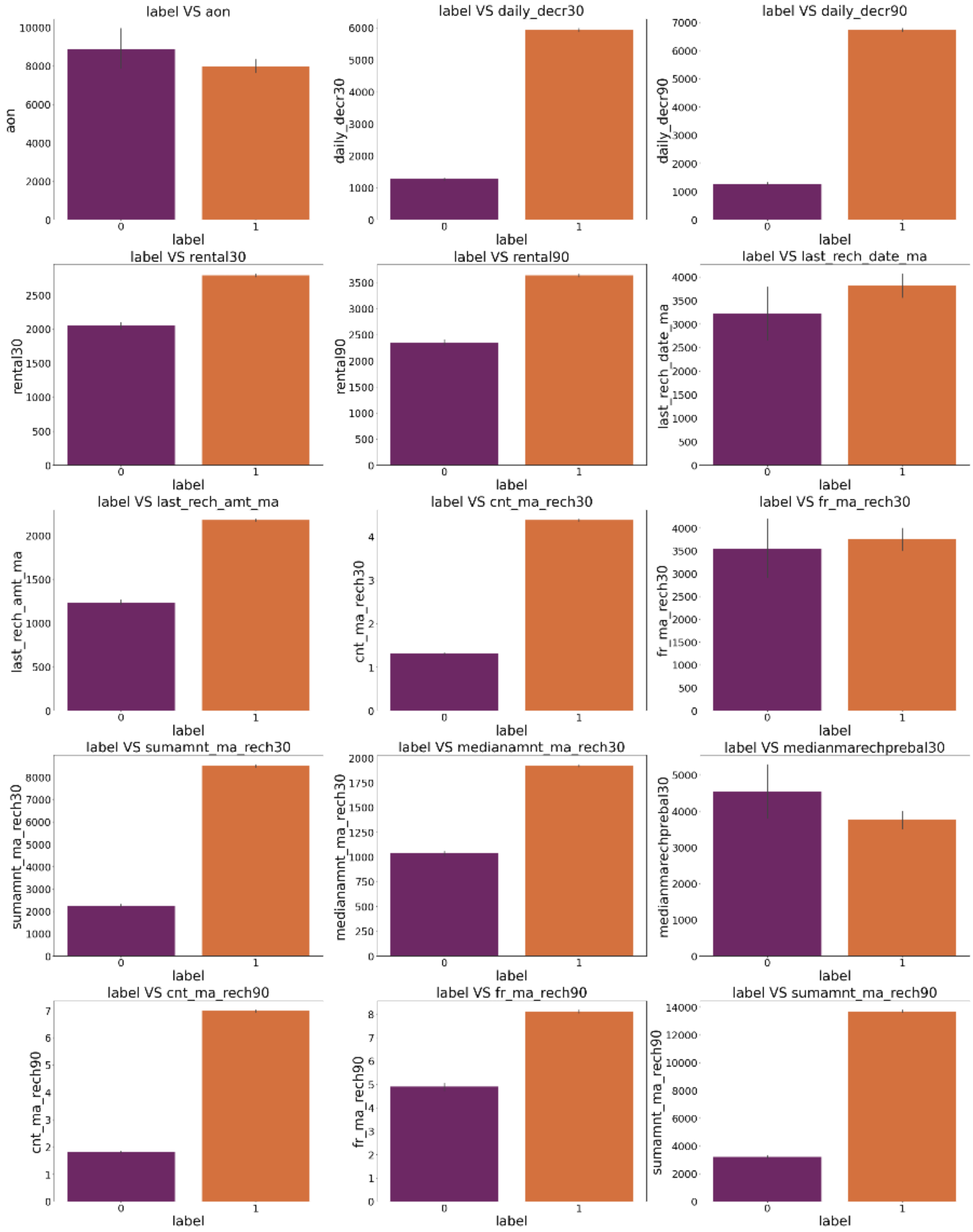


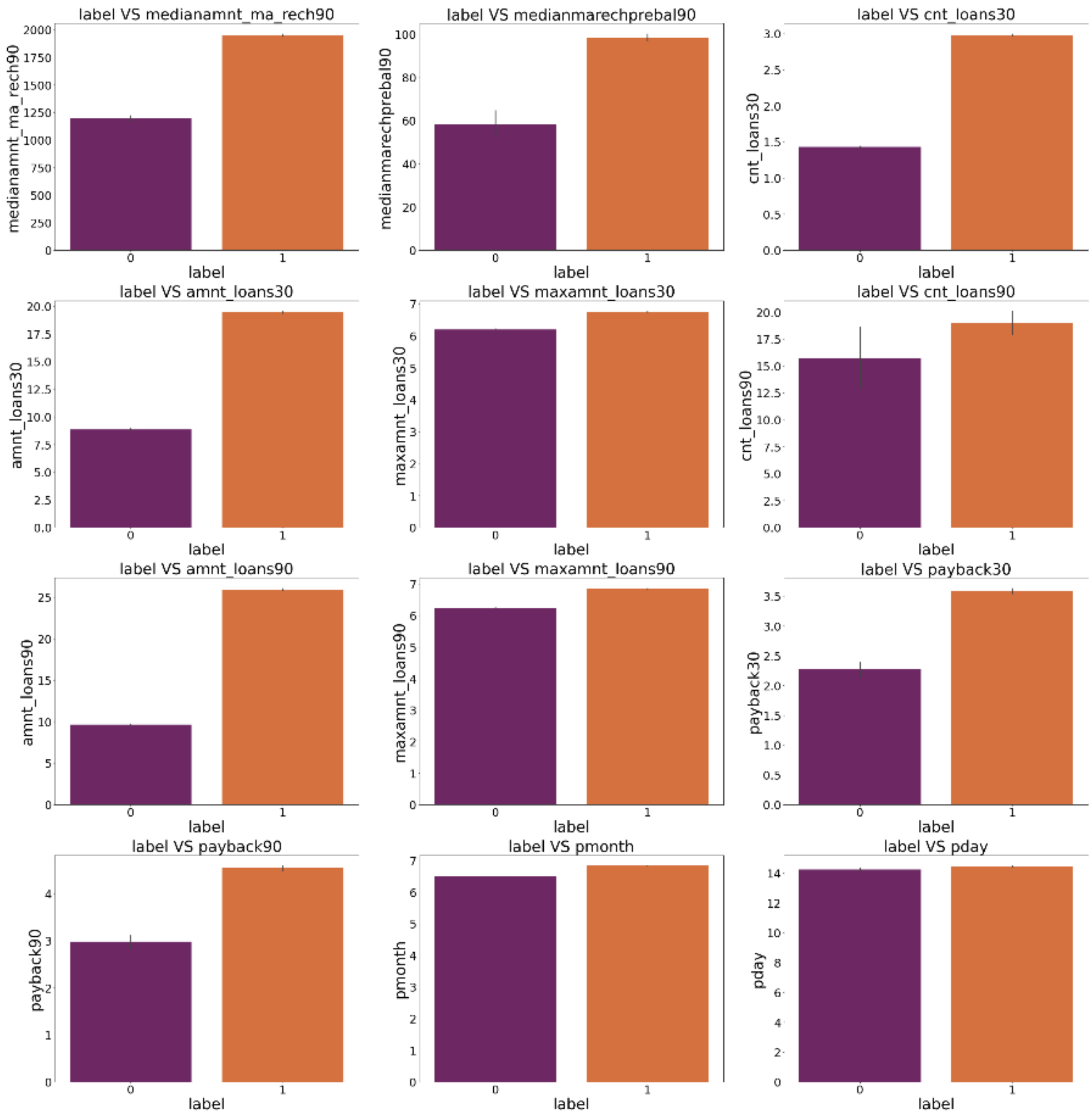
OBSERVATION: We can see skewness present in most of the columns above, we will have to treat skewness before model building.



OBSERVATION: We can see that there are more successes (1) compared to failures (0) in repayment of the loan.

BIVARIATE ANALYSIS:





OBSERVATIONS :

I. Customers with higher Age on cellular network in days(aon) are maximum defaulters (who have not paid their loan amount-0).

II. Customers with a high value of Daily amount spent from the main account, averaged over the last 30 days (in Indonesian Rupiah)(daily_decr30) are maximum. Non-defaulters(who have paid their loan amount-1).

III. Customers with a high value of Daily amount spent from the main account, averaged over the last 90 days (in Indonesian Rupiah)(daily_decr90) are maximum. Non-defaulters (who have paid their loan amount-1).

IV. Customers with a high value of Average main account balance over the last 30 days(rental30) are maximum. Non-defaulters(who have paid their loan amount-1).

V. Customers with a high value of Average main account balance over the last 90 days(rental90) are maximum. Non-defaulters(who have paid their loan amount-1).

VI. Customers with a high Number of days till the last recharge of the main account(last_rech_date_ma) are maximum. Non-defaulters(who have paid their loan amount-1).

VII. Customers with a high value of the last recharge of the main account (in Indonesian Rupiah)(last_rech_amt_ma) are maximum. Non-defaulters(who have paid their loan amount-1).

VIII. Customers with a high value of Number of times the main account got recharged in the last 30 days(cnt_ma_rech30) are maximum. Non-defaulters(who have paid their loan amount-1).

IX. Customers with high value of Frequency of main account recharged in last 30 days(fr_ma_rech30) are maximum. Non-defaulters(who have paid their loan amount-1) and also the count is high for defaulters comparatively. Non-defaulters are more in number.

X. Customers with a high value of Total amount of recharge in the main account over the last 30 days (in Indonesian Rupiah)(sumamnt_ma_rech30) are maximum. Non-defaulters(who have paid their loan amount-1).

XI. Customers with a high value of Median of the amount of recharges done in the main account over the last 30 days at user level (in Indonesian Rupiah)(medianamnt_ma_rech30) are maximum. Non-defaulters(who have paid their loan amount-1).

XII. Customers with a high value of Median of main account balance just before recharge in the last 30 days at user level (in Indonesian Rupiah)(medianmarechprebal30) are maximum defaulters(who have not paid their loan amount-0).

XIII. Customers with a high value of Number of times the main account got recharged in the last 90 days(cnt_ma_rech90) are maximum. Non-defaulters(who have paid their loan amount-1).

XIV. Customers with high value of Frequency of main account recharged in last 90 days(fr_ma_rech90) are maximum. Non-defaulters(who have paid their loan amount-1).

XV. Customers with a high value of Total amount of recharge in the main account over the last 90 days (in Indonesian Rupiah)(sumamnt_ma_rech90) are maximum. Non-defaulters(who have paid their loan amount-1).

XVI. Customers with a high value of Median of the amount of recharges done in the main account over the last 90 days at user level (in Indonesian Rupiah)(medianamnt_ma_rech90) are maximum. Non-defaulters(who have paid their loan amount-1).

XVII. Customers with a high value of Median of main account balance just before recharge in the last 90 days at user level (in Indonesian Rupiah)(medianmarechprebal90) are maximum. Non-defaulters(who have paid their loan amount-1).

XVIII. Customers with a high value of Number of loans taken by the user in the last 30 days(cnt_loans30) are maximum. Non-defaulters(who have paid their loan amount-1).

XIX. Customers with a high value of Total amount of loans taken by the user in the last 30 days(amnt_loans30) are maximum. Non-defaulters(who have paid their loan amount-1).

XX. Customers with a high value of the maximum amount of loan taken by the user in the last 30 days(maxamnt_loans30) are maximum. Non-defaulters(who have paid their loan amount-1).

XXI. Customers with a high value of Number of loans taken by the user in the last 90 days(cnt_loans90) are maximum. Non-defaulters(who have paid their loan amount-1).

XXII. Customers with a high value of Total amount of loans taken by the user in the last 90 days(amnt_loans90) are maximum. Non-defaulters(who have paid their loan amount-1).

XXIII. Customers with a high value of maximum amount of loan taken by the user in the last 90 days(maxamnt_loans90) are maximum. Non-defaulters(who have paid their loan amount-1).

XXIV. Customers with a high value of Average payback time in days over the last 30 days(payback30) are maximum. Non-defaulters(who have paid their loan amount-1).

XXV. Customers with a high value of Average payback time in days over the last 90 days(payback90) are maximum. Non-defaulters(who have paid their loan amount-1).

XXVI. In between 6th and 7th month maximum customers both defaulters and Non-defaulters have paid their loan amount.

XXVII. By the 14th of each month all the customers have paid their loan amount.

Interpretation of the Results

Visualizations: I have used distribution plots to visualize the numerical variables. Used bar plots to check the relation between label and the features. The heat map and bar plot helped me to understand the correlation between dependent and independent features. Also, heat maps helped to detect the multicollinearity problem and feature importance. Detected outliers and skewness with the help of box plots and distribution plots respectively. And I found some of the features skewed to the right. I got to know the count of each column using bar plots

Model building: After cleaning and processing data, I performed a train test split to build the model. I have built multiple classification models to get the accurate accuracy score, and evaluation metrics like precision, recall, confusion matrix, f1 score. I got the Extra Trees Classifier as the best model which gives a 95.60% accuracy score. I checked the cross-validation score ensuring there will be no overfitting. After tuning the best model Extra Trees Classifier multiple times with different attributes, after multiple tries with hyper parameter tuning, the highest accuracy score obtained was 91.58 % and we also got an increment in AUC-ROC curve. Finally, I saved my final model and got the good predictions results for defaulters.

CONCLUSION

Key Findings and Conclusions of the Study

- This case study aims to give an idea of applying EDA in a real business scenario. In this case study, apart from applying the techniques that we have learnt in the EDA module, we will also develop a basic understanding of risk analytics in banking and financial services and understand how data is used to minimize the risk of losing money while lending to customers.
- From this dataset we were able to understand that the selection of customers for the credit to know whether they are defaulters or non-defaulters are done on the basis of different features.

- In this study, we have used multiple machine learning models to predict the micro credit defaulters' rate.
- We have gone through the data analysis by performing feature engineering, finding the relation between features and the label through visualizations. And got the important features and we used these features to predict the defaulters' rate by building ML models.
- After training the model we checked CV score to overcome the overfitting issue.
- Performed hyper parameter tuning, on the best model. We have also got good prediction results.

Learning Outcomes of the Study in respect of Data Science

While working on this project I learned many things about the micro credit loan banks and organizations and how the machine learning models have helped to predict the defaulters' rate which provides greater understanding into the many causes of loan defaults in Microfinance Banks.

I found that the project was quite interesting as the dataset contains several types of data. I used several types of plotting to visualize the relation between target and features. This graphical representation helped me to understand which features are important and how these features describe the defaulter and non-defaulter rates in the banks. Data cleaning was one of the important and crucial things in this project where I dealt with features having zero values, negative statistical summary and time variables.

Limitations of this work and Scope for Future Work

LIMITATIONS:

- The dataset contains the data of only 2016 year belonging to the telecom industry, if we get the data of other years along with other telecom companies then the dataset would be quite more interesting to handle and predict on varied scenarios.
- In the dataset our data is not properly distributed in some of the columns many of the values in the columns are 0's and negative values which are not realistic. Because I have seen in some of the columns even the person didn't take out a loan but the label says that he paid back the loan amount, which is not correct.
- So, because of that data our models may not make the right patterns and the performance of the model also reduces. So those issues need to be taken care of.
- Due to the presence of huge outliers, we ensure that our model is going to perform well on the dataset.
- Due to the class imbalance we had to balance the class defaulter (0). This might also have some effect on the model.

FUTURE WORK:

- The potential future work for this project will be a further development of the model by deepening analysis on variables used in the models as well as creating new variables in order to make better predictions.