Micro-Credit Defaulter Model

Submitted by:

VANISREE P G

ACKNOWLEDGMENT

First I would like to thank the Almighty for his wonderful presence with me throughout this project and helped me to make it as a successful one.

For my internship I had the pleasure of working at FILP ROBO Was a great chance for acquired knowledge, personal and Professional development.

I extend whole hearted thanks to FILP ROBO under whom I worked and learned a lot and for enlightening me with their knowledge and experience to grow with the corporate working.

This is a great pleasure to express my deep sense of gratitude and thanks to SME for his valuable ideas, instantaneous help, effective support and continued encouragement which enabled for the successful completion of the project. I also like to thank the data trained mentors and Technical team members for helping me with technical queries.

And these are the following website which I referred for the reference

1. https://www.kaggle.com/
2. https://scikit-learn.org/
3. www.stackoverflow.com
4. www.google.com
5. www.geeksforgeeks.org

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 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.

❖Motivation for the Problem Undertaken

Credit score models have been successfully applied in a traditional credit card industry and by mortgage firms to determine defaulting customer from the non-defaulting customer. In the light of growing competition in the microfinance industry, over-indebtedness and other factors, the industry has come under increased regulatory supervision. Our study provides evidence from a large microfinance institutions (MFI) in India, and we have applied both the credit scoring method and neural network (NN) method and compared the results. In this article, we demonstrate the capability of credit scoring models for an Indian-based microfinance firm in terms of predicting default probability as well the relative importance of each of its associated drivers. A logistic regression model and NN have been used as the predictive analytic tools for sifting the key drivers of default.

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.

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 payed i.e. Non- defaulter, while, Label ‘0’ indicates that the loan has not been payed i.e. defaulter.

❖ Review of Literature

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.

Microfinance is a category of financial services targeting individuals and small businesses who lack access to conventional banking and related services. Microfinance includes microcredit the provision of small loans to poor clients; savings and checking accounts microinsurance and payment systems, among other services. Microfinance services are designed to reach excluded

customers, usually poorer population segments, possibly socially marginalized, or geographically more isolated, and to help them become self-sufficient.

Microfinance initially had a limited definition: the provision of microloans to poor entrepreneurs and small businesses lacking access to credit. The two main mechanisms for the delivery of financial services to such clients were: relationship-based banking for individual entrepreneurs and small businesses; and group-based models, where several entrepreneurs come together to apply for loans and other services as a group. Over time, microfinance has emerged as a larger movement whose object is: "a world in which as everyone, especially the poor and socially marginalized people and households have access to a wide range of affordable, high quality financial products and services, including not just credit but also savings, insurance, payment services, and fund transfers.

Proponents of microfinance often claim that such access will help poor people out of poverty, including participants in the Microcredit Summit Campaign. For many, microfinance is a way to promote economic development, employment and growth through the support of micro-entrepreneurs and small businesses; for others it is a way for the poor to manage their finances more effectively and take advantage of economic opportunities while managing the risks. Critics often point to some of the ills of micro-credit that can create indebtedness. Many studies have tried to assess its impacts.

Microfinance, also called microcredit, is a type of banking service provided to unemployed or low-income individuals or groups who otherwise would have no other access to financial services.

While institutions participating in the area of microfinance most often provide lending—microloans can range from as small as $100 to as large as $25,000—

many banks offer additional services such as checking and savings accounts as well as micro-insurance products, and some even provide financial and business education. The goal of microfinance is to ultimately give impoverished people an opportunity to become self-sufficient.

- Microfinance is a banking service provided to unemployed or low-income individuals or groups who otherwise would have no other access to financial services.
- Microfinance allows people to take on reasonable small business loans safely, and in a manner that is consistent with ethical lending practices.
- The majority of micro financing operations occur in developing nations, such as Uganda, Indonesia, Serbia, and Honduras.1234
- Like conventional lenders, micro financiers charge interest on loans and institute specific repayment plans.
- The World Bank estimates that more than 500 million people have benefited from microfinance-related operations.5

❖ Motivation for the Problem Undertaken

This project includes the real time problem for Microfinance Institution (MFI), 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 analyse and with Data Science the financial world becomes more interesting. The objective of the project is to prepare a model based on the sample dataset that classifies all loan defaulters and help our client in further investment and improvement in selection of customers. 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.

To understand real world problems where Machine Learning and Data Analysis can be applied to help to predict to make better decisions with the help of which they can gain profit or can be escaped from any loss which otherwise could be possible without the study of data.

Analytical Problem Framing

❖Mathematical/ Analytical Modelling of the Problem

In the whole research process various mathematical, statistical and analytics modelling has been done. There has been reduction of the columns because few of them was not necessary for the problem solving. And few of them was removed due to very less correlation with dependent variable. Since the dataset contains a lot of features hence feature selection has been also done.

In machine learning, several algorithms are applied to forecast the prices of flight tickets. The algorithms are: Decision tree, SVM, KNeighborsClassifier and Random Forest Algorithm. These models have been implemented using the python library Sklearn. The parameters like accuracy_score , confusion_matrix are considered to check the efficiency of these models

After uploading the data I get to know the data by the data.describe() so many information the min value , max value, SD the 25 percentile the 50th percentile the 75 percentile of the data. Then by the help of .skew() I get to know the skewness of the data. Then by the help of correlation function I get to know the correlation of each columns with each other. From the heatmap I can visualized to see them clearly that they are positive correlated or the negative correlated the dark side is show the negative correlation among each other the lighter side represent the positive correlation among the each other.

❖Data Sources and their formats

The most critical aspect of this project is the accumulation of knowledge. To prepare the models, The data set comes from my internship company – Fliprobo technologies in excel format.

There are 37 columns and 209593 rows in this dataset. The different features in dataset are as below:

- label : Flag indicating whether the user paid back the credit amount within 5 daysof issuing the loan {1:success, 0:failure}
- msisdn : mobile number of user
- aon : age on cellular network in days
- daily_decr30 : Daily amount spent from main account, averaged over last 30 days(in Indonesian Rupiah)
- daily_decr90 : Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)

- rental30 : Average main account balance over last 30 days
- rental90 : Average main account balance over last 90 days
- last_rech_date_ma : Number of days till last recharge of main account
- last_rech_date_da: Number of days till last recharge of data account
- last_rech_amt_ma : Amount of last recharge of main account (in Indonesian Rupiah)
- cnt_ma_rech30 : Number of times main account got recharged in last 30 days
- fr_ma_rech30 : Frequency of main account recharged in last 30 days
- sumamnt_ma_rech30 : Total amount of recharge in main account over last 30 days(in Indonesian Rupiah)
- medianamnt_ma_rech30 : Median of amount of recharges done in main accountover last 30 days at user level (in Indonesian Rupiah)
- medianmarechprebal30 : Median of main account balance just before recharge inlast 30 days at user level (in Indonesian Rupiah)
- cnt_ma_rech90 : Number of times main account got recharged in last 90 days
- fr_ma_rech90 : Frequency of main account recharged in last 90 days
- sumamnt_ma_rech90: Total amount of recharge in main account over last 90 days(in Indonesian Rupiah)
- medianamnt_ma_rech90: Median of amount of recharges done in main accountover last 90 days at user level (in Indonesian Rupiah)
- medianmarechprebal90: Median of main account balance just before recharge inlast 90 days at user level (in Indonesian Rupiah)
- cnt_da_rech30 : Number of times data account got recharged in last 30 days
- fr_da_rech30: Frequency of data account recharged in last 30 days
- cnt_da_rech90 : Number of times data account got recharged in last 90 days
- fr_da_rech90 : Frequency of data account recharged in last 90 days
- cnt_loans30 : Number of loans taken by user in last 30 days
- amnt_loans30 : Total amount of loans taken by user in last 30 days

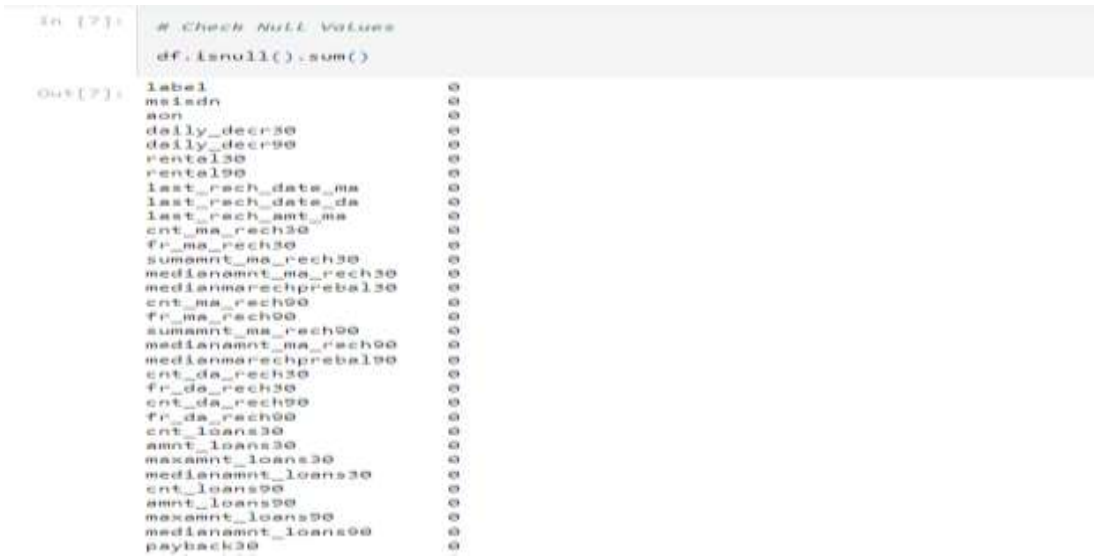
- maxamnt_loans30 : maximum amount of loan taken by the user in last 30 days
- medianamnt_loans30 : Median of amounts of loan taken by the user in last 30 days
- cnt_loans90 : Number of loans taken by user in last 90 days
- amnt_loans90 : Total amount of loans taken by user in last 90 days
- maxamnt_loans90 : maximum amount of loan taken by the user in last 90 days
- medianamnt_loans90 : Median of amounts of loan taken by the user in last 90 days
- payback30: Average payback time in days over last 30 days
- payback90: Average payback time in days over last 90 days
- pcircle: telecom circle
- pdate: date

❖Data Pre-processing

Before building model, the data should be properly pre processed and converted to quality, clean data even the resulting machine learning model will be of great quality .The data pre-processing includes three main parts that is data integration, data cleaning, data transformation. In data integration the data collected from various sources are integrated. In data cleaning process the data containing the null values, unnecessary rows with null values are being cleared. The data transformation includes the feature scaling , categorical data, etc to set the certain range of data.

The raw data is taken and performed various steps to reduce skewness, outlier, class imbalance and scaling. There were null value was present and removed the values from the dataset. Many outlier removal and skewness removal methods are tested and best method is chosen in order to prevent data loss.

a) Checking missing value from the data set.



```
In [7]: # Check Null Values
        df.isnull().sum()

Out[7]: label                    0
        msisdn                   0
        aon                      0
        daily_decr30             0
        daily_decr90             0
        rental30                 0
        rental90                 0
        last_rech_date_ma        0
        last_rech_date_da        0
        last_rech_amt_ma         0
        cnt_ma_rech30            0
        fr_ma_rech30             0
        sumamnt_ma_rech30        0
        medianamnt_ma_rech30     0
        medianmarechprebal30     0
        cnt_ma_rech90            0
        fr_ma_rech90             0
        sumamnt_ma_rech90        0
        medianamnt_ma_rech90     0
        medianmarechprebal90     0
        cnt_da_rech30            0
        fr_da_rech30             0
        cnt_da_rech90            0
        fr_da_rech90             0
        cnt_loans30              0
        amnt_loans30             0
        maxamnt_loans30          0
        medianamnt_loans30       0
        cnt_loans90              0
        amnt_loans90             0
        maxamnt_loans90          0
        medianamnt_loans90       0
        payback30                0
```

There were no null value was present in the dataset and there is no outliers are present in the data.

- **Data error and correction in maxamnt_loans30 column**
- **Feature Engineering on 'pdate' column**

Simple feature engineering operation perform on ‘pdate’ to extract day, month and year column. At last Unnamed :0, PCircle , msisdn columns are drop as they are unnecessary for further investigation

In [12]:

```
# Converting Date datatypes and spliting date into date, month and year.
df['pdate']=pd.to_datetime(df['pdate'])
df['Day']=df['pdate'].apply(lambda x:x.day)
df['Month']=df['pdate'].apply(lambda x:x.month)
df['Year']=df['pdate'].apply(lambda x:x.year)
df.head()
```

- **Converting all negative values to positive values.**

In [15]:

```
#Converting all negative values to positive values in above columns
df['aon']=abs(df['aon'])
df['daily_decr30']=abs(df['daily_decr30'])
df['daily_decr90']=abs(df['daily_decr90'])
df['rental30']=abs(df['rental30'])
df['rental90']=abs(df['rental90'])
df['last_rech_date_ma']=abs(df['last_rech_date_ma'])
df['last_rech_date_da']=abs(df['last_rech_date_da'])
```

- **Outliers Detection and removal**

Outliers detected in boxplot. In order Removing outliers using zscore is not suggested since we loose more than 20% of the data which we cannot afford. After performing this droping the outliers methord creates Nan values. so now we can replace null values by median data.Based on this observationwe decided.

Removing outliers using zscore is not suggested since we loose more than 20% of the data

In [47]:

```
#Dropping the outlier rows with standard deviation
factor = 3
upper_lim = df2.mean () + df2.std () * factor
lower_lim = df2.mean () - df2.std () * factor

data = df2[(df2 < upper_lim) & (df2 > lower_lim)]
```

After performing this droping the outliers methord creates Nan values. so now we can replace null values by median data.

- **Skewness in dataset**

Considerable amount of skewness found in most features by skew () function. Power transformer from sklearn.preprocessing library used to transform skewness in features.

❖ Data Inputs- Logic- Output Relationship

- The input data contains 209593 rows and 36 columns.
- Label (target variable) depends on all the features of 30 or 90 daysmobile loan/repay .
- Pcricle,Pdate and msisdn are removed from the table since that is not much effective in predicting the target variable.

❖ Hardware and Software Requirements and Tools Used

Hardware used for doing the project is a ‘Laptop’ with high end specification and stable internet connection .while coming to the software part I had used ‘python jupyter notebook’ for do my python program and data analysis.

Excel file and Microsoft excel are required for the data handling. In jupyter notebook I had imported lot of python libraries are carried to this project.

1.Pandas-a library which is used to read the data ,visualisation and analysis of data.

2.Numpy-used for working with array and various mathematical operations in python.

3.Seaborn- visualization for plotting different type of plot.

4.Matplotlib- It provides an object-oriented API for embedding plots into applications .

Model/s Development and Evaluation

❖ Identification of possible problem-solving approaches (methods)

Classification Model with following algorithms

- KneighborsClassifier
- LogisticRegression
- DecisionTreeClassifier
- GaussianNB
- SVC

Evaluation metrics

- Accuracy score
- Precison, recall
- AUC,ROC
- F1 score

In this case, Label '1' indicates that the loan has been payed i.e. Non- defaulter, while, Label '0' indicates that the loan has not been payed i.e. defaulter. . Our objective is to predict whether customer is defaulter or not. This becomes binary classification problem which can be solved using various classification algorithms. In order to gain high accuracy of model we will train model with different classification model and select final model among them. To enhance performance of best model will employ hyper parameter tuning over it. At end we will save our final model using joblib.

❖ Testing of Identified Approaches (Algorithms)

Listing down all the algorithms used for the training and testing.

- DT=DecisionTreeRegressor()
- rf=RandomForestRegressor()
- svr=SVR()
- R=Ridge()
- KNN=KNeighborsClassifier()

❖ KEY METRICS FOR SUCCESS IN SOLVING PROBLEM UNDER CONSIDERATION

▪ The precision is the ratio tp / (tp + fp) where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier to not label a sample as positive if it is negative.

▪ The recall is the ratio tp / (tp + fn) where tp is the number of true positives and fn the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples. 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 alsoto 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.

- **roc _auc _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

❖ Run and Evaluate selected models

Describe all the algorithms used along with the snapshot of their code and what were the results observed over different evaluation metrics. Train-test data splits were conducted. In this situation, we split the data into training and test sets, then fit candidate models on the training set, evaluate and select them on the test set.

1. Random Forest Classifier

```
In [68]: clf= RandomForestClassifier(n_estimators=700).fit(x_train, y_train)
         predRFC= clf.predict(x_test)
         randomforest_accu=accuracy_score(y_test,predRFC)
         randomforest_accu
         #print(confusion_matrix(y_test, predRFC))
         print(classification_report(y_test, predRFC))
```

```
              precision    recall  f1-score   support

         0.0       0.72      0.58      0.64      5313
         1.0       0.94      0.97      0.95     36606

    accuracy                           0.92     41919
   macro avg       0.83      0.78      0.80     41919
weighted avg       0.91      0.92      0.91     41919
```

```
In [69]: randomforest_accu
```

```
Out[69]: 0.9180323958109687
```

cross validation score

```
In [70]: from sklearn.model_selection import cross_val_score
         randomforest_cv= cross_val_score(clf,x,y,scoring='accuracy', cv = 3).mean()
         randomforest_cv
```

```
Out[70]: 0.9212235161437053
```

2.SVM

```
In [71]: from sklearn.svm import SVC
         from sklearn.model_selection import cross_val_score
         from sklearn import svm
         from sklearn.svm import LinearSVC
         #svc=SVC(kernel='rbf')
         #svc.fit(x_train,y_train)

         svc = LinearSVC(random_state=0, tol=1e-5)
         svc.fit(x_train, y_train.ravel())


         svc.score(x_train,y_train)
         predsvc=svc.predict(x_test)
         svc_acc=accuracy_score(y_test,predsvc)
         print(svc_acc)
         print(confusion_matrix(y_test,predsvc))
         print(classification_report(y_test,predsvc))
```

```
0.7500178916481787
[[ 4071  1242]
 [ 9237 27360]]
              precision    recall  f1-score   support

         0.0       0.31      0.77      0.44      5313
         1.0       0.96      0.75      0.84     36606

    accuracy                           0.75     41919
   macro avg       0.63      0.76      0.64     41919
weighted avg       0.87      0.75      0.79     41919
```

cross validation score

```
In [72]: from sklearn.model_selection import GridSearchCV, cross_val_score
         svc_cv=cross_val_score(svc,x,y,scoring='accuracy', cv = 3).mean()
         svc_cv
```

```
Out[72]: 0.8891899986556714
```

3.Decision Tree Regression

```
In [73]: from sklearn.tree import DecisionTreeClassifier

         # creating model
         model_dt = DecisionTreeClassifier()

         # feeding the training set into the model
         model_dt.fit(x_train, y_train)

         # predicting the results for the test set
         y_pred = model_dt.predict(x_test)

         # calculating the training and testing accuracies
         print("Training accuracy :", model_dt.score(x_train, y_train))
         print("Testing accuracy :", model_dt.score(x_test, y_test))
         # classification report
         print(classification_report(y_test, y_pred))

         # confusion matrix
         print(confusion_matrix(y_test, y_pred))
         print('accuracy_score :' ,accuracy_score(y_test,y_pred))
```

```
Training accuracy : 0.9999829720260538
Testing accuracy : 0.8825115102936616
              precision    recall  f1-score   support

         0.0       0.54      0.53      0.53      5313
         1.0       0.93      0.93      0.93     36606

    accuracy                           0.88     41919
   macro avg       0.73      0.73      0.73     41919
```

```
weighted avg       0.88      0.88      0.88     41919

[[ 2825  2488]
 [ 2437 34169]]
accuracy_score : 0.8825115102930616
```

```
In [74]: from sklearn.model_selection import cross_val_score
         dtc_cv=cross_val_score(model_dt,x,y,scoring='accuracy', cv = 5).mean()
         dtc_cv
```

```
Out[74]: 0.8840323067213048
```

4.GaussianNB

```
In [75]: from sklearn.naive_bayes import GaussianNB
         # creating model
         model_gnb = GaussianNB()

         # feeding the training set into the model
         model_gnb.fit(x_train, y_train)

         # predicting the results for the test set
         y_pred = model_gnb.predict(x_test)

         # calculating the training and testing accuracies
         print("Training accuracy :", model_gnb.score(x_train, y_train))
         print("Testing accuracy :", model_gnb.score(x_test, y_test))
         # classification report
         print(classification_report(y_test, y_pred))

         # confusion matrix
         print(confusion_matrix(y_test, y_pred))
         print('accuracy_score :',accuracy_score(y_test,y_pred))
```

```
Training accuracy : 0.7412838413876792
Testing accuracy : 0.7499048361172738
              precision    recall  f1-score   support

         0.0       0.30      0.72      0.42      5313
         1.0       0.95      0.75      0.84     36606

    accuracy                           0.75     41919
   macro avg       0.62      0.74      0.63     41919

[[ 3809  1504]
 [ 8976 27630]]
accuracy_score : 0.7499048361172738
```

```
In [76]: gnb_cv=cross_val_score(model_gnb,x,y,scoring='accuracy', cv = 5).mean()
         gnb_cv
```

```
Out[76]: 0.7985655486440258
```

5.KNeighborsClassifier

```
In [77]: from sklearn.neighbors import KNeighborsClassifier
         KNN = KNeighborsClassifier()

         KNN.fit(x_train,y_train)

         predKNN = KNN.predict(x_test)

         reportKNN = classification_report(y_test,predKNN, output_dict = True)

         crKNN = pd.DataFrame(reportKNN).transpose()
         knn_acc=accuracy_score(y_test,predKNN)
         print(knn_acc)
         crKNN
```

```
0.8342279157422648
```

```
Out[77]:
```

	precision	recall	f1-score	support
0.0	0.412137	0.722191	0.524790	5313.000000
1.0	0.954736	0.850489	0.899603	36606.000000
accuracy	0.834228	0.834228	0.834228	0.834228
macro avg	0.683437	0.786340	0.712196	41919.000000
weighted avg	0.885965	0.834228	0.852097	41919.000000

```
In [80]: knn_cv=cross_val_score(KNN,x,y,scoring="accuracy", cv = 3)
         knn_cv.mean()
```

```
Out[80]: 0.9048393774433601
```

Evaluating the model accuracy is an essential part of the process of creating machine learning models to describe how well the model is performing in its predictions. The Precision, Recall,Accuracy score ,F1-score metrics and Cross validation Score are mainly used to evaluate the prediction error rates and model performance in regression analysis.

After evaluating the model based on Precision, Recall,Accuracy score ,F1-score metrics and Cross validation Score the best model choose for hyper parameter tuning are RandomForestRegressor, DecisionTreeRegressor and GaussianNB

A. RandomForestRegressor
B. DecisionTreeRegressor
C. GaussianNB

Hyper parametertuning

A. RandomForestRegressor

```
In [87]: from sklearn.model_selection import GridSearchCV
         parameter ={ 'max_depth': np.arange(2,10),'criterion':['gini','entropy']}
         rf=GridSearchCV(RandomForestClassifier(),parameter,cv=3)

In [88]: #rf=RandomForestClassifier(max_depth=2, random_state=0)
         rf.fit(x_train,y_train)

Out[88]: GridSearchCV(cv=3, estimator=RandomForestClassifier(),
                      param_grid={'criterion': ['gini', 'entropy'],
                                  'max_depth': array([2, 3, 4, 5, 6, 7, 8, 9])})

In [89]: rf.best_params_

Out[89]: {'criterion': 'gini', 'max_depth': 9}

In [90]: clf= RandomForestClassifier(criterion='gini',max_depth=9,random_state=71).fit(x_train, y_train)
         fianlRFC= clf.predict(x_test)
         randomforest_accu=accuracy_score(y_test,fianlRFC)
         randomforest_accu
         #print(confusion_matrix(y_test, predRFC))
         print(classification_report(y_test, fianlRFC))

                       precision    recall  f1-score   support

                  0.0       0.86      0.36      0.51      5354
                  1.0       0.91      0.99      0.95     36565
```

```
              accuracy                           0.91     41919
             macro avg       0.89      0.68      0.73     41919
          weighted avg       0.91      0.91      0.89     41919
```

```
In [92]: randomforest_accu

Out[92]: 0.9109234476013264

In [91]: randomforest_cv= cross_val_score(clf,x,y,scoring='accuracy', cv = 3).mean()
         randomforest_cv

Out[91]: 0.9117670935484578
```

B. Decision Tree Regression

```
In [93]: from sklearn.model_selection import GridSearchCV
         parameter ={ 'max_depth': np.arange(2,10),'criterion':['gini','entropy'],'splitter':['best', 'random']}

In [94]: GCV=GridSearchCV(DecisionTreeClassifier(),parameter,cv=5)

In [95]: GCV.fit(x_train,y_train)

Out[95]: GridSearchCV(cv=5, estimator=DecisionTreeClassifier(),
                      param_grid={'criterion': ['gini', 'entropy'],
                                  'max_depth': array([2, 3, 4, 5, 6, 7, 8, 9]),
                                  'splitter': ['best', 'random']})

In [96]: GCV.best_params_

Out[96]: {'criterion': 'gini', 'max_depth': 9, 'splitter': 'best'}

In [97]: # Create the some more parameters list
         parameters_DT = {'max_depth': [6,7,8,9,10],
         'min_samples_leaf': [2,3,4,5,6,7],
         'min_samples_split': [2,3,4,5,6,7]}

In [98]: GCV=GridSearchCV(DecisionTreeClassifier(),parameters_DT,cv=5)

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

Out[99]: GridSearchCV(cv=5, estimator=DecisionTreeClassifier(),
                      param_grid={'max_depth': [6, 7, 8, 9, 10],
                                  'min_samples_leaf': [2, 3, 4, 5, 6, 7],
                                  'min_samples_split': [2, 3, 4, 5, 6, 7]})

In [100]: GCV.best_params_

Out[100]: {'max_depth': 10, 'min_samples_leaf': 5, 'min_samples_split': 5}

In [101]: final_mod=DecisionTreeClassifier(criterion='gini',splitter= 'best',random_state = 1,max_depth=10,min_samples_leaf= 5, min_sample
          final_mod.fit(x_train,y_train)
          pred=final_mod.predict(x_test)
          acc=accuracy_score(y_test,pred)
          print(acc)

          0.9179131181564446

In [102]: from sklearn.model_selection import cross_val_score
          dtc_cv=cross_val_score(final_mod,x,y,scoring='accuracy', cv = 5).mean()
          dtc_cv

Out[102]: 0.9176165219389201
```

c. GaussianNB

```
In [103]: from sklearn.model_selection import GridSearchCV
nb_classifier = GaussianNB()

params_NB = {'var_smoothing': np.logspace(0,-9, num=100)}
gs_NB = GridSearchCV(estimator=nb_classifier,
                     param_grid=params_NB,
                     cv=5,   # use any cross validation technique
                     verbose=1,
                     scoring='accuracy')
gs_NB.fit(x_train, y_train)
```

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
```

```
Out[103]: GridSearchCV(cv=5, estimator=GaussianNB(),
             param_grid={'var_smoothing': array([1.00000000e+00, 8.11130831e-01, 6.57933225e-01, 5.33669923e-01,
       4.32876128e-01, 3.51119173e-01, 2.84803587e-01, 2.31012970e-01,
       1.87381742e-01, 1.51991108e-01, 1.23284674e-01, 1.00000000e-01,
       8.11130831e-02, 6.57933225e-02, 5.33669923e-02, 4.32876128e-02,
       3.51119173e-02, 2.84803587e-02, 2.31...
       1.23284674e-07, 1.00000000e-07, 8.11130831e-08, 6.57933225e-08,
       5.33669923e-08, 4.32876128e-08, 3.51119173e-08, 2.84803587e-08,
       2.31012970e-08, 1.87381742e-08, 1.51991108e-08, 1.23284674e-08,
       1.00000000e-08, 8.11130831e-09, 6.57933225e-09, 5.33669923e-09,
       4.32876128e-09, 3.51119173e-09, 2.84803587e-09, 2.31012970e-09,
       1.87381742e-09, 1.51991108e-09, 1.23284674e-09, 1.00000000e-09])},
             scoring='accuracy', verbose=1)
```

```
In [104]: gs_NB.best_params_
```

```
Out[104]: {'var_smoothing': 1.0}
```

```
In [106]: from sklearn.naive_bayes import GaussianNB
GNB = GaussianNB(var_smoothing = 1.0)

GNB.fit(x_train,y_train)

predGNB = GNB.predict(x_test)

reportGNB = classification_report(y_test, predGNB, output_dict = True)

crGNB = pd.DataFrame(reportGNB).transpose()
gnb_acc=accuracy_score(y_test,predGNB)
print(gnb_acc)
crGNB
```

```
0.8582981464232487
```

```
Out[106]:
```

	precision	recall	f1-score	support
0.0	0.440389	0.404371	0.421618	5354.000000
1.0	0.913818	0.924764	0.919258	36565.000000
accuracy	0.858298	0.858298	0.858298	0.858298
macro avg	0.677103	0.664567	0.670437	41919.000000
weighted avg	0.853352	0.858298	0.855698	41919.000000

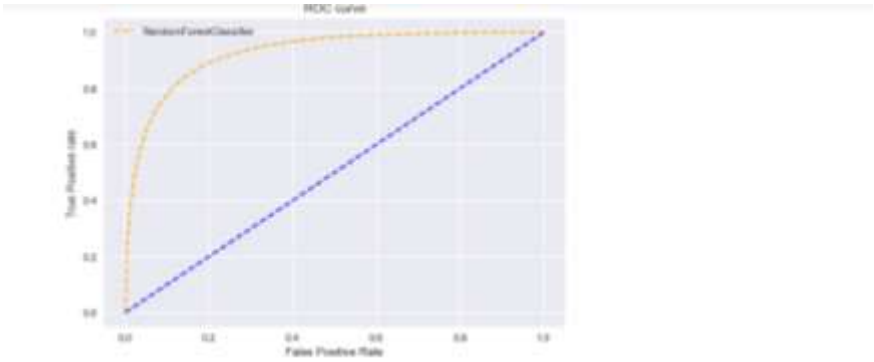
```
In [107]: #Now lets try to do some evaluation for GaussianNB model using cross validation.
gnb_cv = cross_val_score(estimator = final_mod, X = x_train, y = y_train, cv = 10)
gnb_cv.mean()
```

```
Out[107]: 0.8576404371779217
```

The best model after hyper parameter tuning is Random Forest Regressor.

AOC -ROC CURVE OF ML MODELS

```
In [114]: import matplotlib.pyplot as plt
plt.style.use('seaborn')

# plot roc curves
plt.plot(fpr1, tpr1, linestyle='--',color='orange', label='RandomForestClassifier')
plt.plot(p_fpr, p_tpr, linestyle='--', color='blue')
# title
plt.title('ROC curve')
# x label
plt.xlabel('False Positive Rate')
# y label
plt.ylabel('True Positive rate')

plt.legend(loc='best')
plt.savefig('ROC',dpi=300)
plt.show();
```



8.Saving the model

```
In [118]: import joblib
          joblib.dump(fianlRFC,"Micro_Credit_Defaulter_Final.pkl")

Out[118]: ['Micro_Credit_Defaulter_Final.pkl']
```

In our project we had implemented various Machine Learning Algorithms such as Decision Tree Regression, Random Forest Regression ,SVM, GaussianNB and KNeighborsClassifier compared the accuracy of results based on our test data set. Based on the various accuracy levels we find that Random Forest Regression gives the highest accuracy i.e. 91%. Therefore we selected Random Forest Regression and created User Interface based on it.

Visualizations

After cleaning the data, we can visualize data and better understand the relationships between different variables. There are many more visualizations that you can do to learn more about your dataset, like scatterplots, histograms, boxplots.

- The Analysis of Label Distribution.

```
In [21]: sns.countplot(x='label',data=df)
         plt.title('Label Distribution',fontsize=18,fontweight ='bold')
         plt.xlabel("Label Class",fontsize=18,fontweight ='bold')
         plt.ylabel("No. of Customer",fontsize=18,fontweight ='bold')
         plt.xticks(rotation=90)

Out[21]: (array([0, 1]), [Text(0, 0, '0'), Text(1, 0, '1')])
```

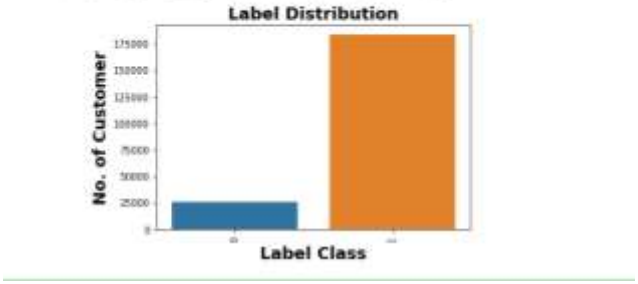


Fig. 1: The Analysis of label

We can see Most of customers are non-defaulter while very few are defaulter. From ML model building point of view target variable is imbalanced which need to balance using balancing techniques.

- Label class 1 represent Non-defaulter while Label class 0 represent Defaulter i.e. Loan not paid
- We can see Most of customers are Non-defaulter while very few are defaulter.
- From ML model building point of view target variable is imbalanced data case.

- The Analysis of label based on month

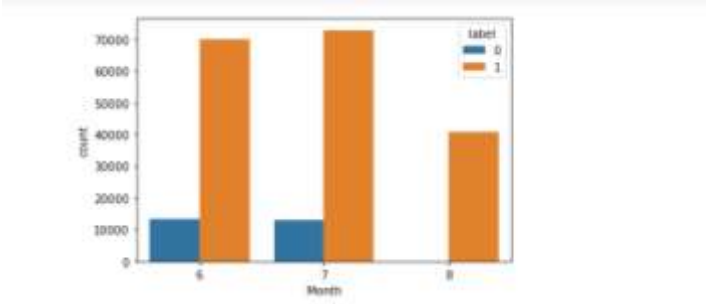


Fig. 2: The Analysis of label based on month

- Most of data belong to month 6 and 7, followed my month 8.
- We can see very few defaulter in month 8.

➢ The Analysis of Number of loans vs Amount of loan

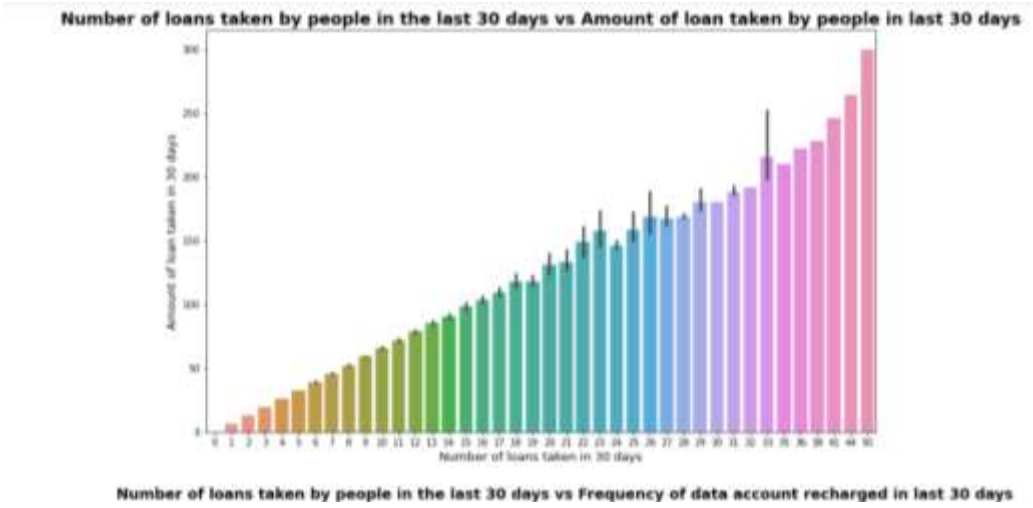


Fig. 3: The Analysis of Number of loans vs Amount of loan

- Maximum number of loans taken by the people is 50 and the Average loan amount is equivalent to 300.
- Minimum number of loans taken by the people is 0.

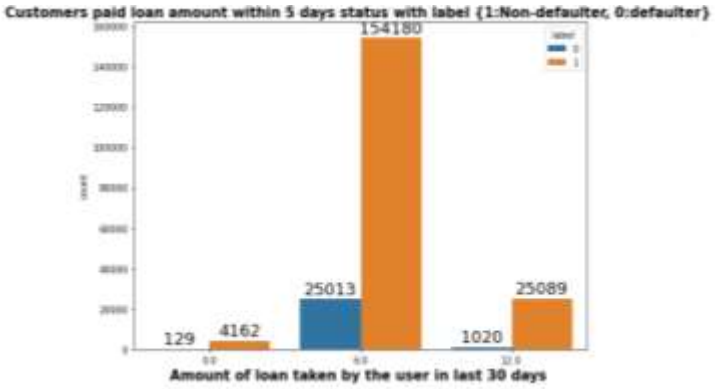


Fig. 4: The Analysis of Amount of loan taken by the user in last 30 days.

➢ Very few defaulter in customers who take loan in amount of 12.

❖ The Analysis of maximum number of loans taken vs Average payback time .

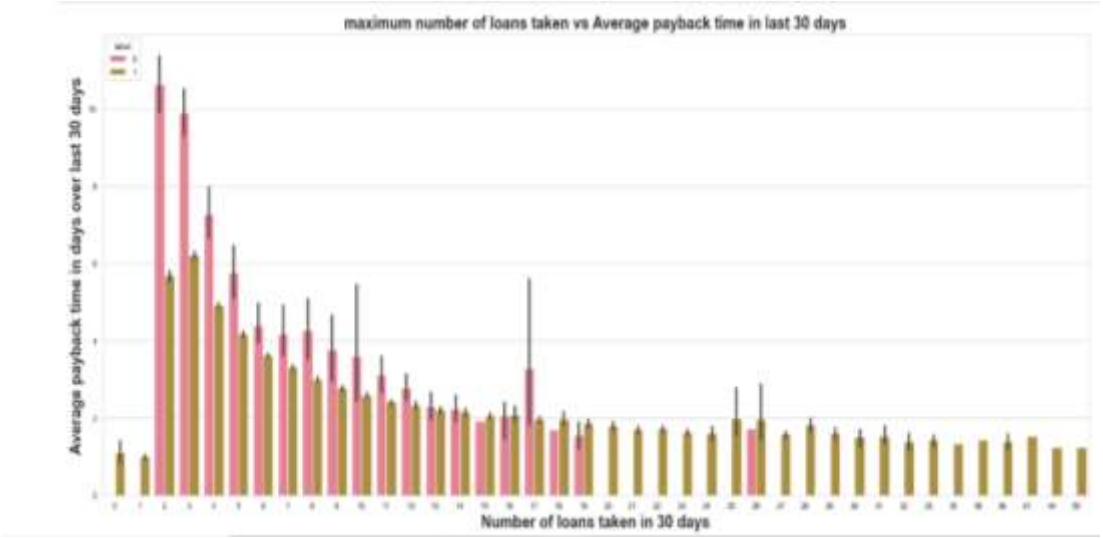


Fig. 5: The Analysis of maximum number of loans taken vs Average payback time .

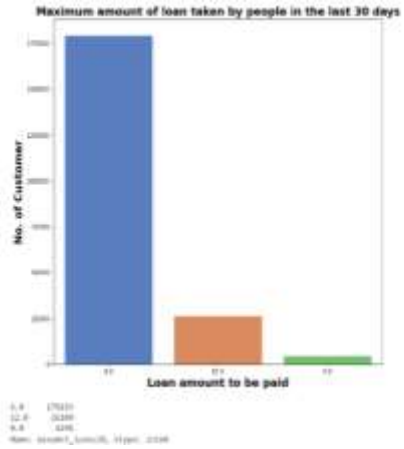
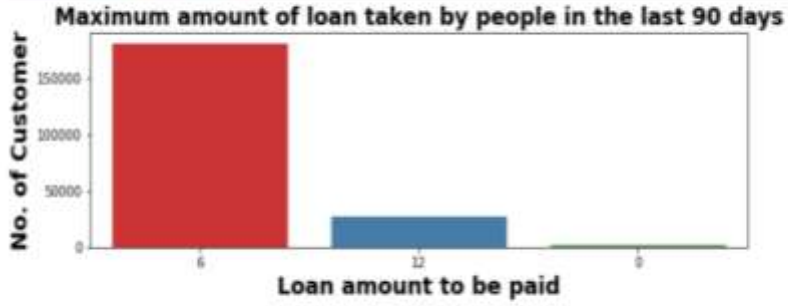


Fig. 6: The Analysis of Maximum amount of loan taken by people in the last 30 days

```
plt.xlabel('Loan amount to be paid',fontsize=18,fontweight ='bold')
plt.ylabel('No. of Customer',fontsize=18,fontweight ='bold')
plt.show()
print(df['maxamnt_loans90'].value_counts())
```




```
6     180945
12     26605
0       2043
Name: maxamnt_loans90, dtype: int64
```

Fig. 7: The Analysis of Maximum amount of loan taken by people in the last 90 day

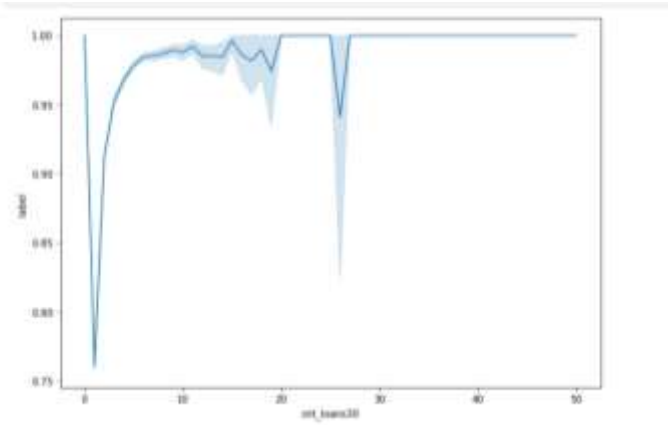
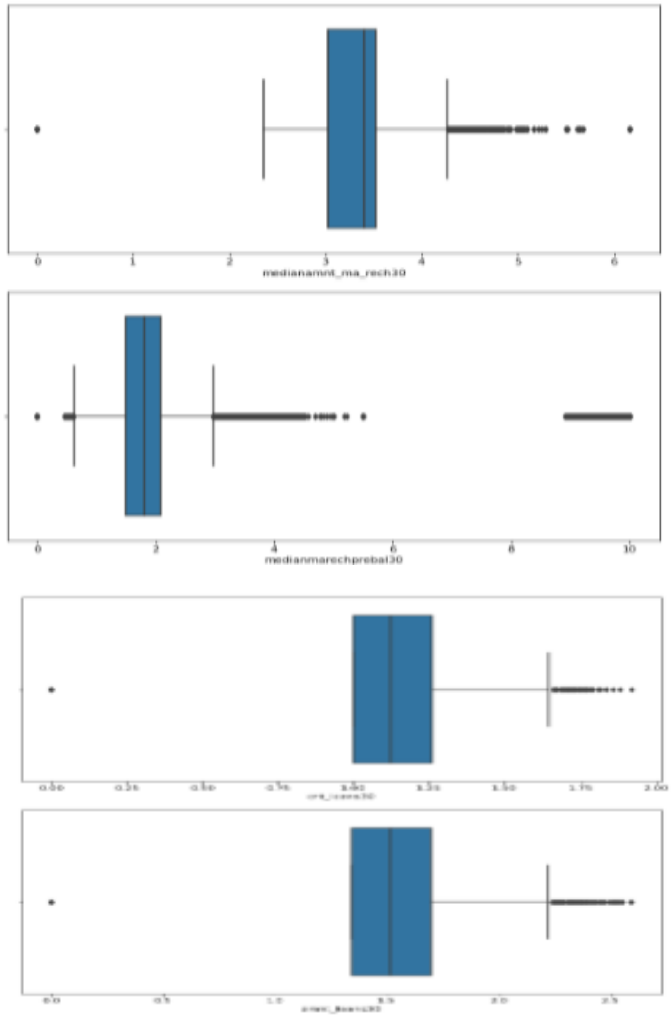
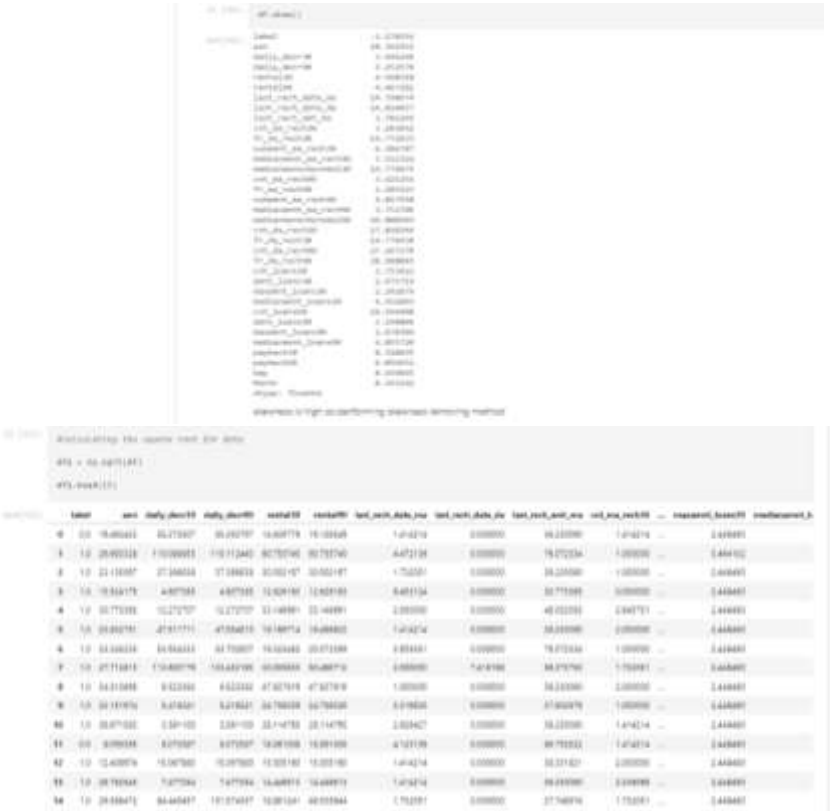


Fig. 8: The Analysis between cnt_loans30 and label

- Checking the outliers



- Checking and Removing Skewness



- since the skewness is not reduced much in square root transformation again performing cube root transformation on same data

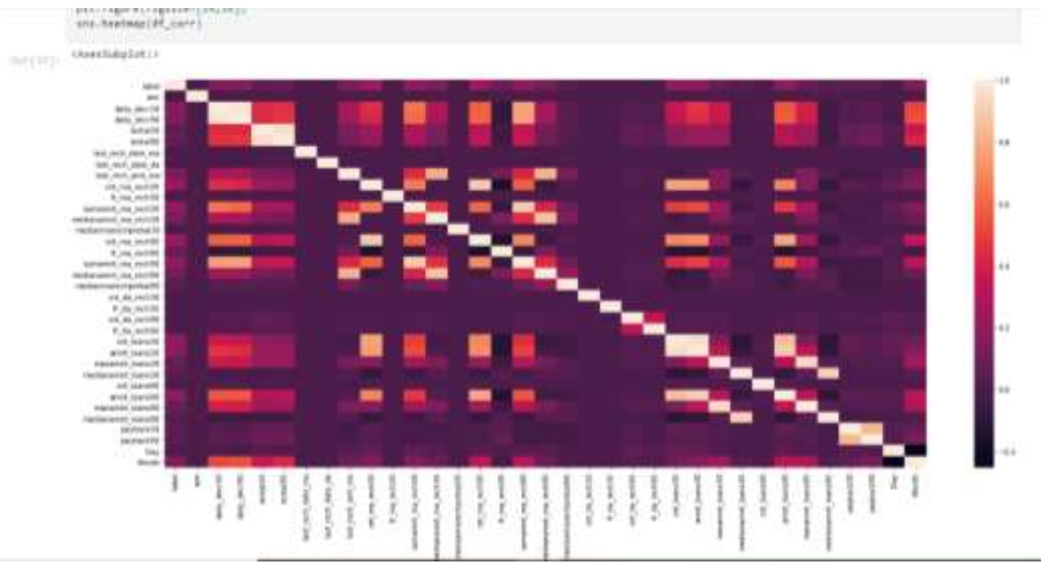
```
In [39]: #calculating the cube root for the column

         df2 = np.cbrt(df1)

         df2.head(15)
         df=df2
```

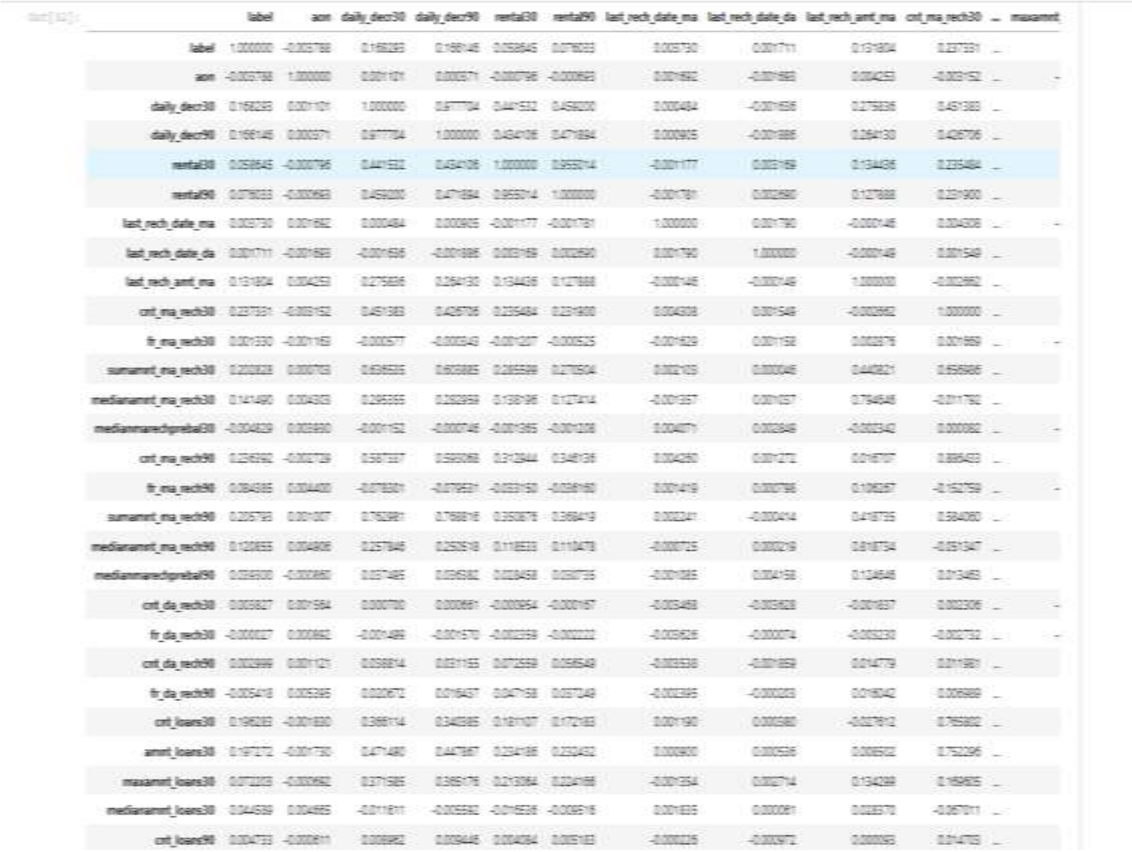
```
In [40]: df2.skew()
```

```
Out[40]: label                  -2.270254
         aon                     6.837774
         daily_decr30           -0.871687
         daily_decr90           -0.824164
         rental30               -0.723701
         rental90               -0.599488
         last_rech_date_ma       6.921141
         last_rech_date_da      11.296889
         last_rech_amt_ma       -1.980510
         cnt_ma_rech30          -1.688780
         fr_ma_rech30            4.464568
         sumamnt_ma_rech30      -1.417171
         medianamnt_ma_rech30   -1.683847
         medianmarechprebal30    2.072666
         cnt_ma_rech90          -1.709576
         fr_ma_rech90           -0.413325
         sumamnt_ma_rech90      -1.462184
         medianamnt_ma_rech90   -1.973927
         medianmarechprebal90   -1.856114
         cnt_da_rech30          12.737370
         fr_da_rech30           13.872648
         cnt_da_rech90           6.120253
         fr_da_rech90           16.042530
         cnt_loans30            -2.387681
         amnt_loans30           -2.157467
         maxamnt_loans30        -5.968672
         medianamnt_loans30      3.474769
         cnt_loans90             3.540770
         amnt_loans90           -1.866361
         maxamnt_loans90        -7.759739
         medianamnt_loans90      3.887399
         payback30               0.179646
         payback90              -0.000861
         Day                    -0.835846
         Month                   0.244902
         dtype: float64
```

➢ CORRELATION

To gain more insight about relationship between input & output heatmap of correlation and bar plot of correlation of label with independents features is plotted.





	label	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cnt_ma_rech30	...	maxamnt
label	1.000000	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
aon	[illegible]	1.000000	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
daily_decr30	[illegible]	[illegible]	1.000000	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
daily_decr90	[illegible]	[illegible]	[illegible]	1.000000	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
rental30	[illegible]	[illegible]	[illegible]	[illegible]	1.000000	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
rental90	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1.000000	[illegible]	[illegible]	[illegible]	[illegible]	...	
last_rech_date_ma	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1.000000	[illegible]	[illegible]	[illegible]	...	
last_rech_date_da	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1.000000	[illegible]	[illegible]	...	
last_rech_amt_ma	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1.000000	[illegible]	...	
cnt_ma_rech30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1.000000	...	
fr_ma_rech30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
sumamnt_ma_rech30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
medianamnt_ma_rech30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
medianmarechprebal30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
cnt_ma_rech90	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
fr_ma_rech90	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
sumamnt_ma_rech90	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
medianamnt_ma_rech90	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
medianmarechprebal90	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
cnt_da_rech30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
fr_da_rech30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
cnt_da_rech90	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
fr_da_rech90	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
cnt_loans30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
amnt_loans30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
maxamnt_loans30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
medianamnt_loans30	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	
cnt_loans90	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	...	

➢ Interpretation of the Results

- Most importantly, the people are paying the amount early or lately and sometimes they might fail to pay within the time frame, but I observed that almost 80% of users are paying the amount within 7-8 days. It is recommended that to extent loan repayment time frame from 5 days to 7 days.
- The collected data is only for one Telecom circle area as per Dataset Documentation so that we had dropped that column.
- Customer who takes a greater number of loans are non-defaulters (i.e., 98% of the category) as they repay the loan within the given timei.e., 5 days.
- So here 'DecisionTreeClassifier Model' is the best model out of all model tested above and by looking this we can conclude that our model is predicting around 92% of correct results for Label ‘0’ indicates that the loan has not been payed i.e. defaulter.
- Tendency to pay loan within 5 days is high among customer who take loan many times within month compare to those who take loan 1-2 times.
- It is recommended that to extent loan repayment time frame from 5 days to 7 days.

CONCLUSION

Key Findings and Conclusions of the Study

From this dataset I get to know that each feature play a very import role to understand the data. Data format plays a very important role in the visualization and Appling the models and algorithms .Importance of removing the skewness and outlier.

With the aim of identifying the determinants of credit default, an attempt was made to judge against defaulters with non-defaulters. Accordingly, it was found to be on average a bit younger with more proportion of them being male, illiterate, and loan diverters. They also receive a smaller credit amounts, earn smaller income, and support more dependents than the non-defaulters. The difference between the two groups was found to be significant in terms of credit/loan diversion and income. The findings of the econometric analysis reveal that, education, number of times borrowed and suitability of repayment period are significant determinants of credit diversion.

Learning Outcomes of the Study in respect of Data Science

Learnt how to process the large number of data. Tried and learnt more about distribution of the data. The power of visualization is helpful for the understanding of data into the graphical representation its help me to understand that what data is trying to say, Data cleaning is one of the most important step to remove missing value or null value fill it by mean median or by mode or by 0.Setting a good parameters is

more important for the model accuracy. Finding a best random state played a vital roll in finding a better model.

Microfinance has been globally accepted as the preferred medium to reach out to the rural and productive poor with banking services which includes micro credit to help alleviate poverty which is one of the United Nations millennium challenge goals. Micro credit default has been identified to be one of the major drawbacks of this laudable initiative as it depletes these revolving funds and reduces investors' confidence. Therefore, it is important to understand the factors that influence a loan beneficiary to default so that appropriate countermeasures can be developed to prevent and reduce the incidents of default.

Limitations of this work and Scope for Future Work

The techniques to increase the speed of the model need to be constructed. In Upcoming days when huge amount of information is accessed as in detailed information in the dataset, the expected results in future are highly correct. For further research anyone desire to expand upon it ought to request different sources of historical data or be a lot of organized in collection knowledge manually over amount of your time to boot.

The techniques to increase the speed of the model need to be constructed. The future model can be constructed with the most co related data with the target variable in order to increase the speed of the model.