FLIP ROBO

NAME OF THE PROJECT

Micro-Credit Defaulter ML

Submitted by:

Yogesh.C.Mudliar

ACKNOWLEDGMENT

It is a genuine pleasure to express my deep sense of thanks and gratitude to my guide, Ms.Khushboo Garg, for allowing me to work on this project. It was a great way to expose myself to the actual research environment.

I thank FLIPROBO for permitting me to work with them.
I take this opportunity to say heartfelt thanks to Dr. Deepika Sharma, VP-learning And development DataTraind for her overall dedication, devotion, and support towards me. I convey my sincere regards to all the DataTraind team thanks for supporting me during academic years of my post-graduation course in data science.

I express my profound sense of gratitude to my mentor Ms.Khushboo Garg, FLIPROBO for her guidance at every step of my research work.

Apart from the project, I learned a lot from her, she gave me valuable thought- "To think"; that I will benefit from, for a long time to come. I am indebted to her more than she knows.

References:

1) https://www.investopedia.com/terms/m/microcredit.asp

2) www.Google.com

3) https://app.grammarly.com/

4) https://www.youtube.com/watch?v=NeOxIiV_ikw

5)https://www.researchgate.net/publication/265161200_Predicting_Credit_Default_among_Micro_Borrowers_in_Ghana

6) https://en.wikipedia.org/

INTRODUCTION

- ## Business Problem Framing

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.

Exercise:

Build a model which can be used to predict in terms of a probability for each loan transaction, whether the customer will be paying back the loaned amount within 5 days of insurance of loan. In this case, Label ‘1’ indicates that the loan has been paid i.e. Non- defaulter, while, Label ‘0’ indicates that the loan has not been paid i.e. defaulter.

- ## Conceptual Background of the Domain Problem

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.

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.

The client wants some predictions that could help them in further investment and improvement in selection of customers.

Exercise:

We are building a model which can be used to predict in terms of a probability for each loan transaction, whether the customer will be paying back the loaned amount within 5 days of insurance of loan. In this case, Label '1' indicates that the loan has been paid i.e. Non-defaulter, while, Label '0' indicates that the loan has not been paid i.e. defaulter.

- ## Review of Literature

Microcredit, also called micro banking or microfinance, a means of e xtending credit , usually in the form of small loans with no collatera, to non-traditional borrowers such as the poor in rural or undeveloped areas.

Microfinance institutions play a major role in economic development in many Developing countries. However many of these microfinance in stitutions are faced with the problem of default because of the non-foral 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 209592 rows and 37 Columns loans from a microfinance institution with braches all over the country we proposed a Random Forest Classifier model to predict the probability of default. Microfinance institutions could use this model to screen prospective loa n applicants in order to reduce the level of default.

- Motivation for the Problem Undertaken

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.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

We are building 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.

There are no null values in the dataset. There may be some customers with no loan history. The dataset is imbalanced. Label '1' has approximately 87.5% records, while, label '0' has approximately 12.5% records.

We have checked the correlation of data. In most of the columns Outliers are present, we have removed the outliers using the zscore method. We have used the various visualization plot for all features. In the distribution plot we observed that there is a skewness present in data. We have removed the skewness using the yeo-johnson method. Then we have built the model, checked their accuracy score, confusion matrix and classification report.

I done Extensive EDA have to be performed to gain relationships of important variable

- Data Sources and their formats

The data was provided by the Flip Robo Technologies and it is in the .csv format. The data is huge it has 209592 rows and 37 columns.

-The below image shows data format

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209592 entries, 0 to 209591
Data columns (total 37 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   Unnamed: 0            209592 non-null  int64
 1   label                 209592 non-null  int64
 2   msisdn                209592 non-null  object
 3   aon                   209592 non-null  float64
 4   daily_decr30          209592 non-null  float64
 5   daily_decr90          209592 non-null  float64
 6   rental30              209592 non-null  float64
 7   rental90              209592 non-null  float64
 8   last_rech_date_ma     209592 non-null  float64
 9   last_rech_date_da     209592 non-null  float64
 10  last_rech_amt_ma      209592 non-null  int64
 11  cnt_ma_rech30         209592 non-null  int64
 12  fr_ma_rech30          209592 non-null  float64
 13  sumamnt_ma_rech30     209592 non-null  float64
 14  medianamnt_ma_rech30  209592 non-null  float64
 15  medianmarechprebal30  209592 non-null  float64
 16  cnt_ma_rech90         209592 non-null  int64
 17  fr_ma_rech90          209592 non-null  int64
 18  sumamnt_ma_rech90     209592 non-null  int64
 19  medianamnt_ma_rech90  209592 non-null  float64
 20  medianmarechprebal90  209592 non-null  float64
 21  cnt_da_rech30         209592 non-null  float64
 22  fr_da_rech30          209592 non-null  float64
 23  cnt_da_rech90         209592 non-null  int64
 24  fr_da_rech90          209592 non-null  int64
 25  cnt_loans30           209592 non-null  int64
 26  amnt_loans30          209592 non-null  int64
 27  maxamnt_loans30       209592 non-null  float64
 28  medianamnt_loans30    209592 non-null  float64
 29  cnt_loans90           209592 non-null  float64
 30  amnt_loans90          209592 non-null  int64
 31  maxamnt_loans90       209592 non-null  int64
 32  medianamnt_loans90    209592 non-null  float64
 33  payback30             209592 non-null  float64
 34  payback90             209592 non-null  float64
 35  pcircle               209592 non-null  object
 36  pdate                 209592 non-null  object
dtypes: float64(21), int64(13), object(3)
memory usage: 59.2+ MB
```

Data Description:

Variable	Definition
label	Flag indicating whether the user paid back the credit amount within 5 days of 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 account over last 30 days at user level (in Indonesian Rupiah)
medianmarechprebal30	Median of main account balance just before recharge in last 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 Indonasian Rupiah)
medianamnt_ma_rech90	Median of amount of recharges done in main account over last 90 days at user level (in Indonasian Rupiah)
medianmarechprebal90	Median of main account balance just before recharge in last 90 days at user level (in Indonasian 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 Preprocessing Done

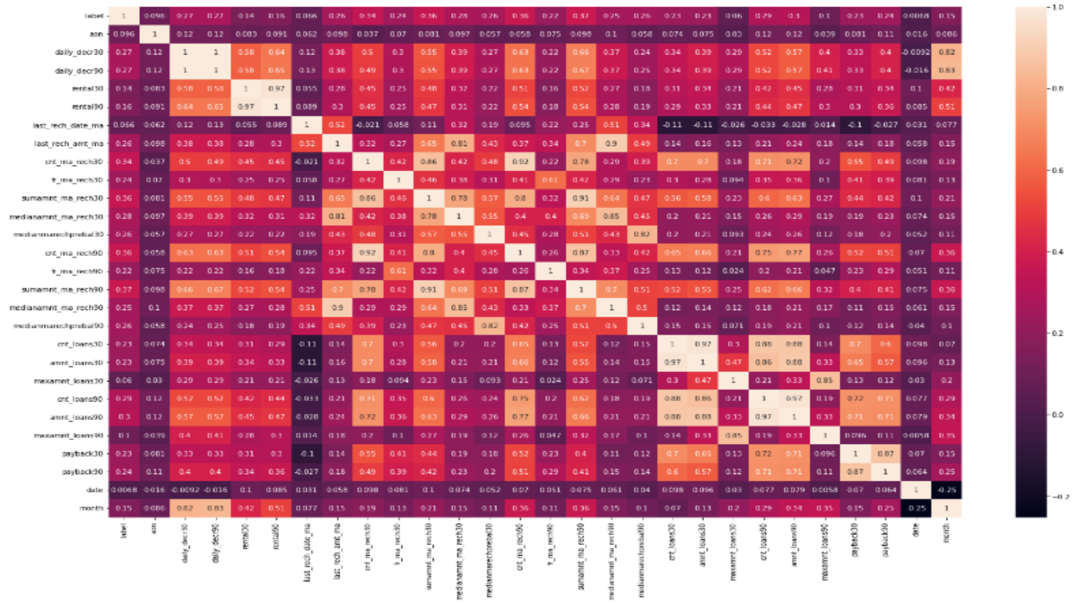
After collecting a data, the next step is to get it ready for analysis. This means cleaning, or ‘scrubbing’ it, and is crucial in making sure that you’re working with high quality data.

- **Removing unwanted data points**—extracting irrelevant observations that have no bearing on intended analysis.
- **Bringing structure to data**—fixing layout issues, which will help to map and manipulate this data more easily.
- **Filling in major gaps**—this data contains null values and I notice that important data are missing. Once we identified gaps, we can go about filling them.

During cleaning the data we have to do EDA(Exploratory Data Analysis (EDA) is an approach to analyse the data using visual techniques. It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations). This helps identify initial trends and characteristics, and can even refine our hypothesis.

- Data Inputs- Logic- Output Relationships

With the correlation plot we can understand the relationship between each feature with the target variables.



There are some inputs which is important to find our outputs like the price of houses with the available independent variables.

'aon', 'daily_decr30', 'daily_decr90', 'rental30', 'rental90', 'last_rech_date_ma', 'last_rech_amt_ma', 'cnt_ma_rech30', 'fr_ma_rech30', 'sumamnt_ma_rech30', 'medianamnt_ma_rech30', 'medianmarechprebal30', 'cnt_ma_rech90', 'fr_ma_rech90', 'sumamnt_ma_rech90', 'medianamnt_ma_rech90', 'medianmarechprebal90', 'cnt_loans30', 'amnt_loans30', 'maxamnt_loans30', 'cnt_loans90', 'amnt_loans90', 'maxamnt_loans90', 'payback30', 'payback90'

- State the set of assumptions (if any) related to the problem under consideration

Default in microfinance is the failure of a client to repay a loan. The default could be in terms of the amount to be paid or the timing of the payment. MFIs can sustain and increase deployment of loans to stimulate the poverty reduction goal if repayment rates are high and consistent. MFIs are able to reduce interest rates and processing fees if repayment rates are high, thus increasing patronage of loans. A high repayment rate is a catalyst for increasing the volume of loan disbursements to various sectors of the economy. Borrowers that do not have formal education are likely to have inadequate knowledge of loan acquisition and management, thereby making them unable to repay the loans given to them. Literate peoples will pay off their loans better than illiterate peoples because they understand the advantages of prompt loan repayment.

- **Hardware and Software Requirements and Tools Used**

  Here is the hardware and software used in the project.
  Processor – core i3
  RAM – 12 GB
  SSD – 250 GB

Software requirements:

numpy : library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

pandas : software library written for the Python programming language for data manipulation and analysis

sklearn: Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines

seaborn: Seaborn is a library that uses Matplotlib underneath to plot graphs. It will be used to visualize random distributions.

matplotlib.pyplot : is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

Worked on these models:

1) Logistic Regression
2) Random Forest Classifier
3) Decision Tree Classifier
4) Gradient Bossting Classifier
5) KNeighbors Classifier

Model/s Development and Evaluation

- **Identification of possible problem-solving approaches (methods)**

Cleaning, or ‘scrubbing’ the data, and is crucial in making sure that we are working with high quality data.

-Removing unwanted data points—extracting irrelevant observations that have no bearing on intended analysis.

-Bringing structure to data—fixing layout issues, which will help to map and manipulate this data more easily.

-Filling in major gaps—This data contains null values and I notice that important data are missing. Once we identified gaps, we can go about filling them.

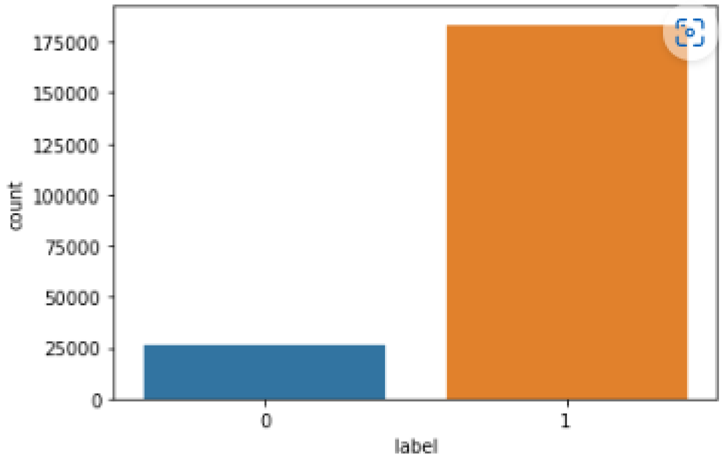
During cleaning the data we have to do EDA (Exploratory Data Analysis (EDA) is an approach to analyse the data using visual techniques. It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations). This helps identify initial trends and characteristics, and can even refine our hypothesis.

I done Predictive analysis (Predictive analytics encompasses a variety of statistical techniques from data mining, predictive modelling, and machine learning that analyse current and historical facts to make predictions about future or otherwise unknown events)

Predictions about which variables are important to predict the price of variable, and how do these variables describe to know the person defaulter or non-defaulter.

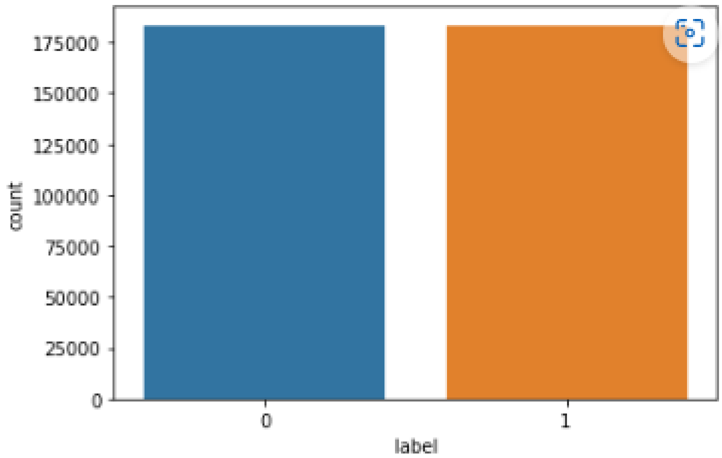
Balancing our target variable:

<AxesSubplot:xlabel='label', ylabel='count'>



↓

<AxesSubplot:xlabel='label', ylabel='count'>



- **Testing of Identified Approaches (Algorithms)**

1) Logistic Regression:

- Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.

2) Random Forest Classifier:

-The random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

3) Decision Tree Classifier:

- The decision tree classifier creates the classification model by building a decision tree. Each node in the tree specifies a test on an attribute, each branch descending from that node corresponds to one of the possible values for that attribute.

4) Gradient Bossting Classifier:

-Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting.

5) KNeighbors Classifier:

- KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

- **Run and Evaluate selected models**

1)Logistic Regression

```
lg=LogisticRegression()
lg.fit(X_train,y_train)
```

```
LogisticRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
lg_pred=lg.predict(X_test)
print("Predicted value:\n",lg_pred)
print("Accuracy Score:",accuracy_score(y_test,lg_pred),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,lg_pred),'\n')
print("Classification Report:\n",classification_report(y_test,lg_pred))
```

```
Predicted value:
 [1 1 1 ... 0 0 1]
Accuracy Score: 0.7721676555302156

Confusion Matrix:
 [[43180 11668]
 [13407 41804]]

Classification Report:
               precision    recall  f1-score   support

           0       0.76      0.79      0.77     54848
           1       0.78      0.76      0.77     55211

    accuracy                           0.77    110059
   macro avg       0.77      0.77      0.77    110059
weighted avg       0.77      0.77      0.77    110059
```

2)Random Forest Classifier

```
rfc=RandomForestClassifier()
rfc.fit(X_train,y_train)
```

```
RandomForestClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
rfc_pred=rfc.predict(X_test)
print("Predicted value:\n",rfc_pred)
print("Accuracy Score:",accuracy_score(y_test,rfc_pred),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,rfc_pred),'\n')
print("Classification Report:\n",classification_report(y_test,rfc_pred))
```

```
Predicted value:
 [1 1 1 ... 0 0 1]
Accuracy Score: 0.952125678045412

Confusion Matrix:
 [[52476  2372]
 [ 2897 52314]]

Classification Report:
               precision    recall  f1-score   support

           0       0.95      0.96      0.95     54848
           1       0.96      0.95      0.95     55211

    accuracy                           0.95    110059
   macro avg       0.95      0.95      0.95    110059
weighted avg       0.95      0.95      0.95    110059
```

3)Decision Tree Classifier

```
dr=DecisionTreeClassifier()
dr.fit(X_train,y_train)
```

```
DecisionTreeClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
dr_pred=dr.predict(X_test)
print("Predicted value:\n",dr_pred)
print("Accuracy Score:",accuracy_score(y_test,dr_pred),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,dr_pred),'\n')
print("Classification Report:\n",classification_report(y_test,dr_pred))
```

```
Predicted value:
 [1 1 0 ... 0 0 1]
Accuracy Score: 0.9152454592536731

Confusion Matrix:
 [[50545  4303]
 [ 5025 50186]]

Classification Report:
               precision    recall  f1-score   support

           0       0.91      0.92      0.92     54848
           1       0.92      0.91      0.91     55211

    accuracy                           0.92    110059
   macro avg       0.92      0.92      0.92    110059
weighted avg       0.92      0.92      0.92    110059
```

4)Gradient Bossting Classifier

```
gbc=GradientBoostingClassifier()
gbc.fit(X_train,y_train)
```

```
GradientBoostingClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
gbc_pred=gbc.predict(X_test)
print("Predicted value:\n",gbc_pred)
print("Accuracy Score:",accuracy_score(y_test,gbc_pred),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,gbc_pred),'\n')
print("Classification Report:\n",classification_report(y_test,gbc_pred))
```

```
Predicted value:
 [1 1 1 ... 0 0 1]
Accuracy Score: 0.901898072851834

Confusion Matrix:
 [[50227  4621]
 [ 6176 49035]]

Classification Report:
               precision    recall  f1-score   support

           0       0.89      0.92      0.90     54848
           1       0.91      0.89      0.90     55211

    accuracy                           0.90    110059
   macro avg       0.90      0.90      0.90    110059
weighted avg       0.90      0.90      0.90    110059
```

5)KNeighbors Classifier

```
knn=KNeighborsClassifier()
knn.fit(X_train,y_train)
```

```
KNeighborsClassifier()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
knn_pred=knn.predict(X_test)
print("Predicted value:\n",knn_pred)
print("Accuracy Score:",accuracy_score(y_test,knn_pred),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,knn_pred),'\n')
print("Classification Report:\n",classification_report(y_test,knn_pred))
```

```
Predicted value:
 [1 1 0 ... 0 0 1]
Accuracy Score: 0.9019525890658646

Confusion Matrix:
 [[54376   472]
 [10319 44892]]

Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.99      0.91     54848
           1       0.99      0.81      0.89     55211

    accuracy                           0.90    110059
   macro avg       0.92      0.90      0.90    110059
weighted avg       0.92      0.90      0.90    110059
```

- **Key Metrics for success in solving problem under consideration**

During cleaning the data we have to do EDA (Exploratory Data Analysis (EDA) is an approach to analyse the data using visual techniques. It is used to discover trends, patterns, or to check assumptions with the help of statistical summary and graphical representations). This helps identify initial trends and characteristics, and can even refine our hypothesis.

I done Predictive analysis (Predictive analytics encompasses a variety of statistical techniques from data mining, predictive modelling, and machine learning that analyse current and historical facts to make predictions about future or otherwise unknown events)

Predictions about which variables are important to predict the price of variable, and how do these variables describe to know the person defaulter or non-defaulter.

Hyperparameter tuning:

Consists of finding a set of optimal hyperparameter values for a learning algorithm while applying this optimized algorithm to any data set. That combination of hyperparameters maximizes the model's performance, minimizing a predefined loss function to produce better results with fewer errors.

Hyper parameter Tuning

```
from sklearn.model_selection import GridSearchCV
```

```
params={'n_estimators':[50,60],
        'criterion':['gini','entropy'],
        'max_features':['auto','log2']}
```

```
grid_search=GridSearchCV(estimator=rfc,param_grid=params,cv=3,verbose=3)
```

```
grid_search.fit(X_train,y_train)
```

```
Fitting 3 folds for each of 8 candidates, totalling 24 fits
[CV 1/3] END criterion=gini, max_features=auto, n_estimators=50;, score=0.946 total time=  49.1s
[CV 2/3] END criterion=gini, max_features=auto, n_estimators=50;, score=0.945 total time=  48.4s
[CV 3/3] END criterion=gini, max_features=auto, n_estimators=50;, score=0.944 total time=  43.5s
[CV 1/3] END criterion=gini, max_features=auto, n_estimators=60;, score=0.946 total time=  55.5s
[CV 2/3] END criterion=gini, max_features=auto, n_estimators=60;, score=0.945 total time=  54.9s
[CV 3/3] END criterion=gini, max_features=auto, n_estimators=60;, score=0.945 total time= 1.0min
[CV 1/3] END criterion=gini, max_features=log2, n_estimators=50;, score=0.945 total time=  38.9s
[CV 2/3] END criterion=gini, max_features=log2, n_estimators=50;, score=0.944 total time=  41.1s
[CV 3/3] END criterion=gini, max_features=log2, n_estimators=50;, score=0.944 total time=  39.6s
[CV 1/3] END criterion=gini, max_features=log2, n_estimators=60;, score=0.946 total time=  45.6s
[CV 2/3] END criterion=gini, max_features=log2, n_estimators=60;, score=0.945 total time=  43.1s
[CV 3/3] END criterion=gini, max_features=log2, n_estimators=60;, score=0.945 total time=  47.8s
[CV 1/3] END criterion=entropy, max_features=auto, n_estimators=50;, score=0.946 total time= 1.0min
[CV 2/3] END criterion=entropy, max_features=auto, n_estimators=50;, score=0.945 total time=  58.6s
[CV 3/3] END criterion=entropy, max_features=auto, n_estimators=50;, score=0.945 total time=  58.7s
[CV 1/3] END criterion=entropy, max_features=auto, n_estimators=60;, score=0.947 total time= 1.3min
```

- ## Visualizations

```
1 df.describe()
```

	Unnamed: 0	label	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_re
count	209592.000000	209592.000000	209592.000000	209592.000000	209592.000000	209592.000000	209592.000000	209592.000000	209592.000000	20
mean	104796.818695	0.875181	8112.380576	5381.384257	6082.500332	2692.577747	3483.395263	3755.865701	3712.220632	
std	60504.519227	0.330514	75696.261202	9220.641701	10918.836731	4308.596638	5770.472738	53906.020205	53374.960144	
min	1.000000	0.000000	-48.000000	-93.012667	-93.012667	-23737.140000	-24720.580000	-29.000000	-29.000000	
25%	52398.750000	1.000000	246.000000	42.439500	42.691917	280.417500	300.260000	1.000000	0.000000	
50%	104796.500000	1.000000	527.000000	1469.091834	1500.000000	1083.540000	1334.000000	3.000000	0.000000	
75%	157195.250000	1.000000	982.000000	7244.000000	7802.272500	3356.820000	4201.715000	7.000000	0.000000	
max	209593.000000	1.000000	999860.755200	265926.000000	320630.000000	198926.110000	200148.110000	998650.377700	999171.809400	5

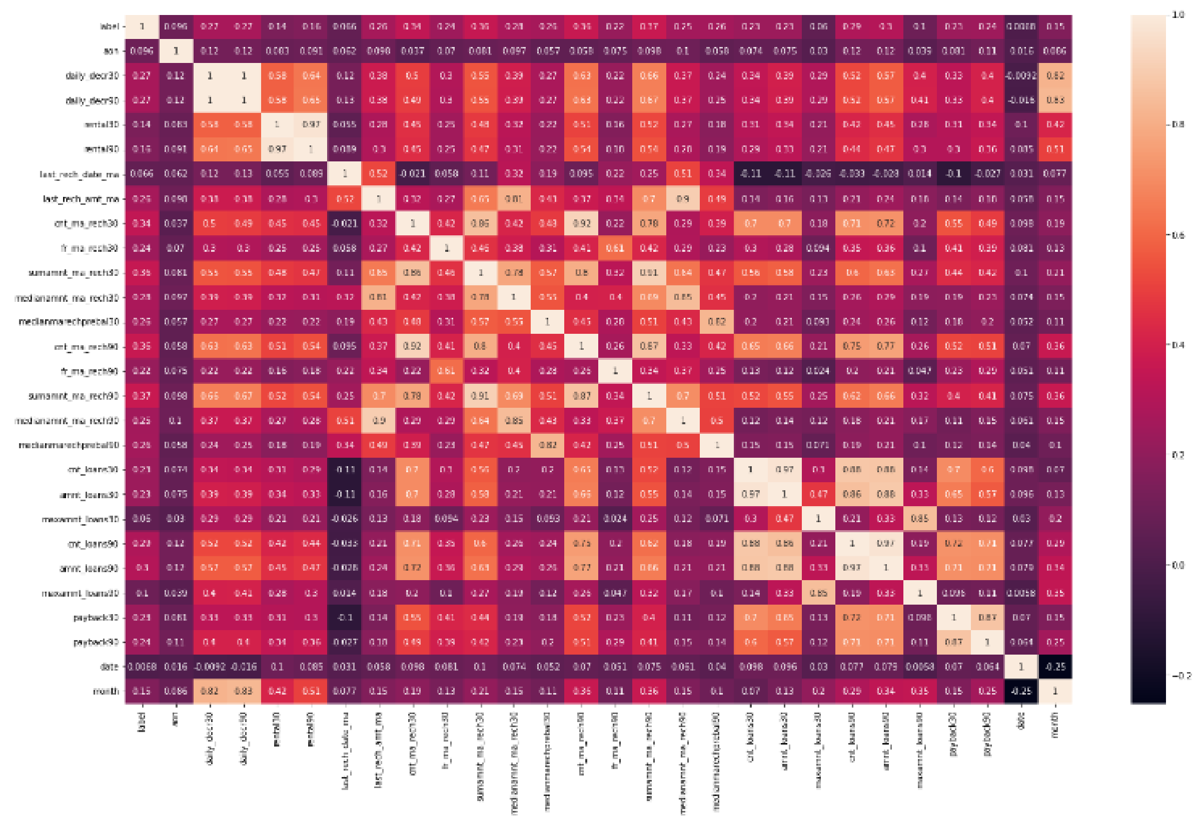
1) We can see the data is imbalanced

2) We have to drop unnecessary columns

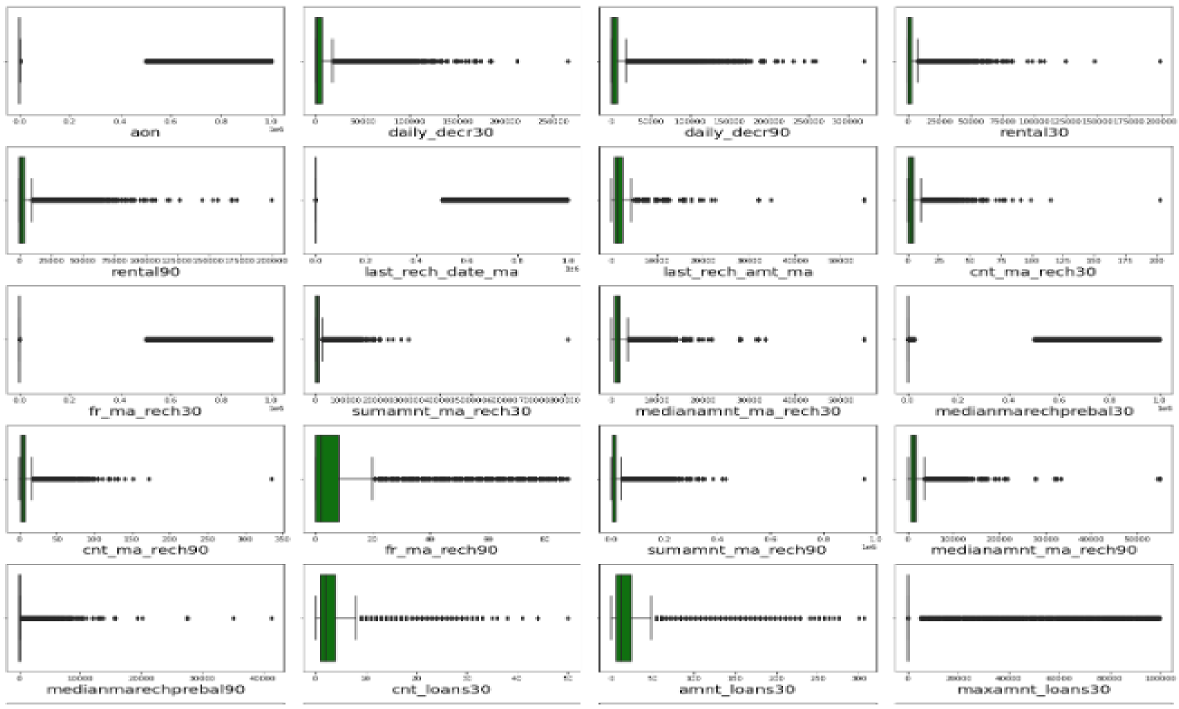
3) The mean value is higher than 50% which shows there is skewness present.

4) These observations suggest that there are outliers in these columns.

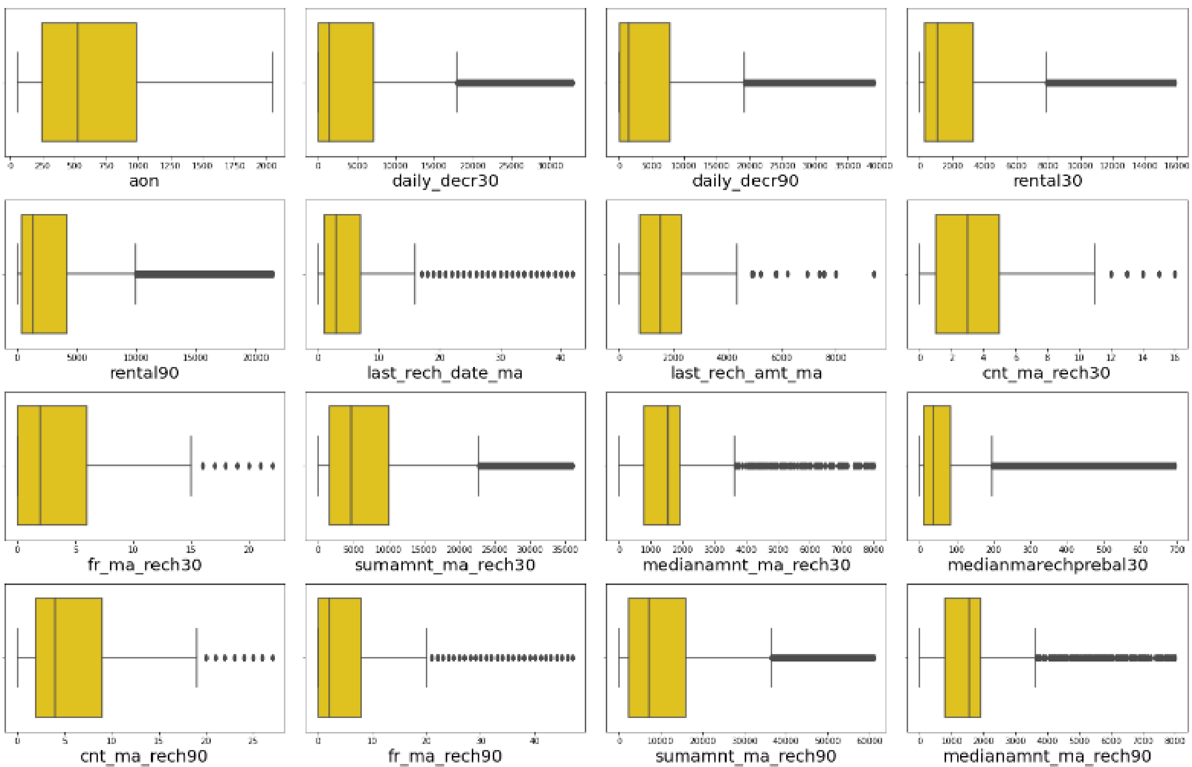
- ## Correlation plot



Removing the outliers from the given datasets



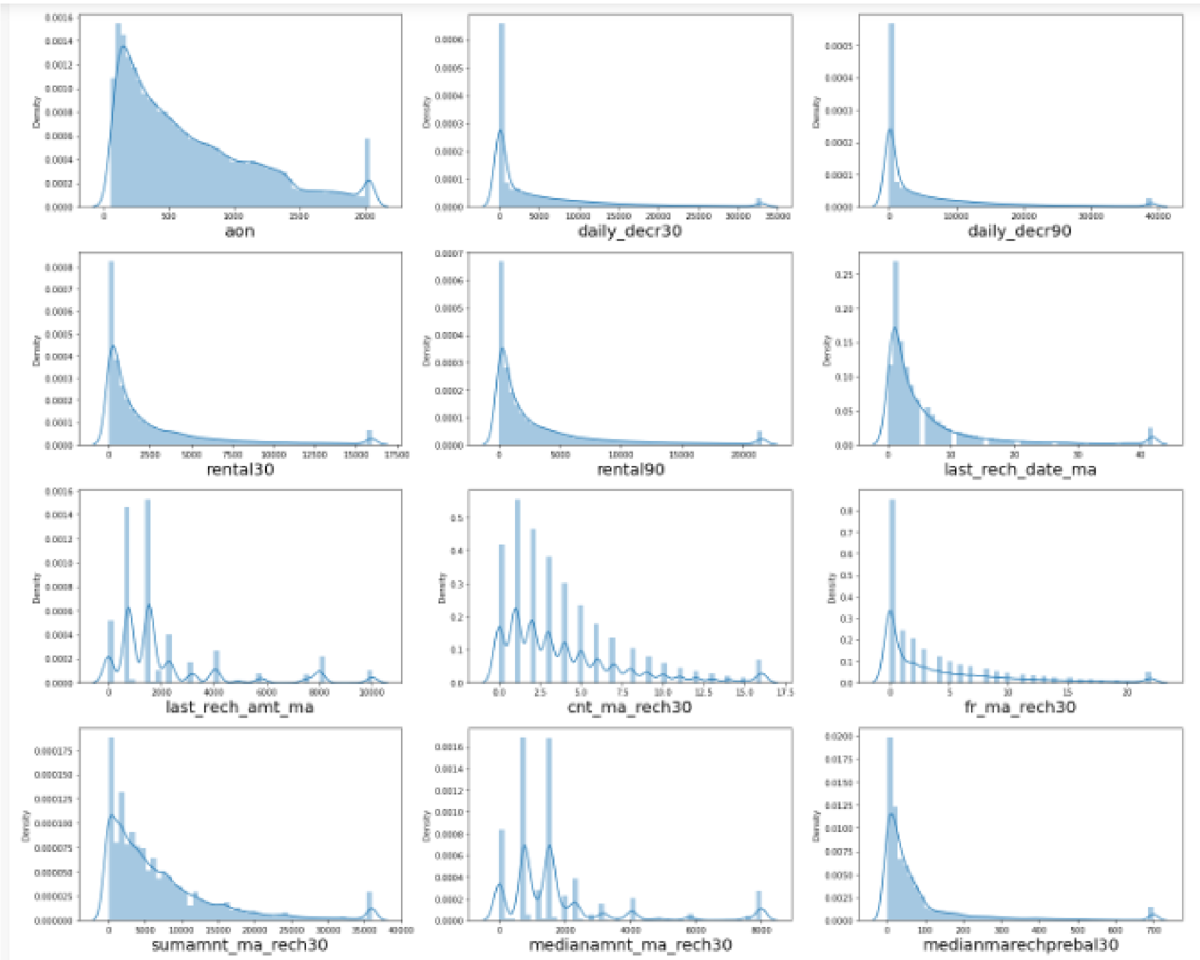
↓



-We can observe the dataset is imbalenced

- Skewness is a measure of the asymmetry of a distribution. A distribution is asymmetrical when its left and right side are not mirror images. A distribution can have right (or positive), left (or negative), or zero skewness.

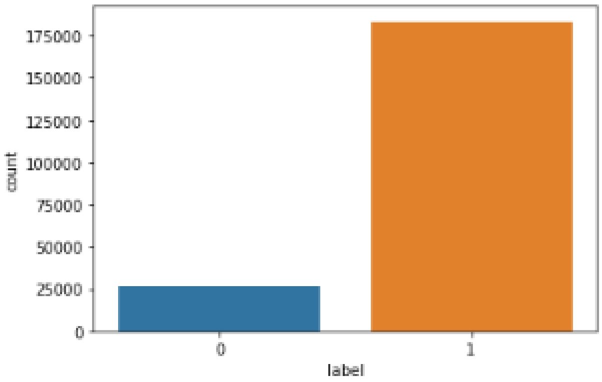
- We have positive skew , the tail of a distribution curve is longer on the right side. This means the outliers of the distribution curve are further out towards the right and closer to the mean on the left. Skewness does not inform on the number of outliers; it only communicates the direction of outliers.



-Balancing the target

```
sns.countplot(y)
```

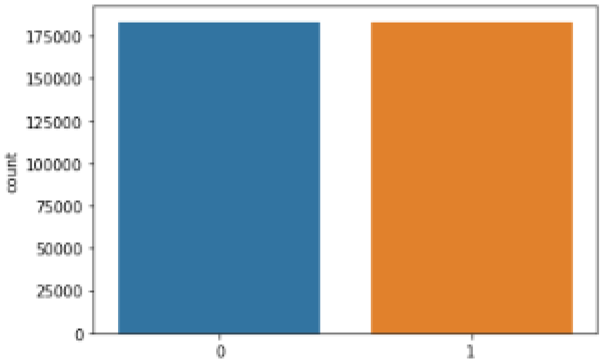
```
<AxesSubplot:xlabel='label', ylabel='count'>
```




```
# We balenced our target column
from imblearn.over_sampling import SMOTE
SM = SMOTE()
X, y = SM.fit_resample(X,y)
```

```
sns.countplot(y)
```

```
<AxesSubplot:xlabel='label', ylabel='count'>
```



- ## Interpretation of the Results

```
#Cross Validation score
from sklearn.model_selection import cross_val_score

print("Cross Validation score for Logistic Regression:",cross_val_score(lg,X,y,cv=5).mean()*100)
print("Cross Validation score for Random Forest Classifier:",cross_val_score(rfc,X,y,cv=5).mean()*100)
print("Cross Validation score for Decision Tree Classifier:",cross_val_score(dr,X,y,cv=5).mean()*100)
print("Cross Validation score for Gradient Bossting Classifier:",cross_val_score(gbc,X,y,cv=5).mean()*100)
print("Cross Validation score for KNeighbors Classifier:",cross_val_score(knn,X,y,cv=5).mean()*100)
```

```
Cross Validation score for Logistic Regression: 77.2175922984819
Cross Validation score for Random Forest Classifier: 94.9981337985202
Cross Validation score for Decision Tree Classifier: 91.08903556587184
Cross Validation score for Gradient Bossting Classifier: 89.80516925204778
Cross Validation score for KNeighbors Classifier: 90.50814759328425
```

- We got Cross Validation score for Random Forest Classifier: 94.998 1337985202

Selecting Best Accuracy Score Model

```
best_model=RandomForestClassifier(criterion='gini',max_features='auto',n_estimators=60)
```

```
best_model.fit(X_train,y_train)
```

```
RandomForestClassifier(max_features='auto', n_estimators=60)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
best_model_pred=best_model.predict(X_test)
print("Predicted value:",best_model_pred)
print("Accuracy Score:",accuracy_score(y_test,best_model_pred),'\n')
print("Confusion Matrix:\n",confusion_matrix(y_test,best_model_pred),'\n')
print("Classification Report:\n",classification_report(y_test,best_model_pred))
```

```
Predicted value: [1 1 1 ... 0 0 1]
Accuracy Score: 0.9523164847945194

Confusion Matrix:
 [[52496  2352]
 [ 2896 52315]]

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.96      0.95     54848
           1       0.96      0.95      0.95     55211

    accuracy                           0.95    110059
   macro avg       0.95      0.95      0.95    110059
weighted avg       0.95      0.95      0.95    110059
```

-I select Random Forest Classifier model for Micro-Credit Defaulter, I also checked AUC-ROC curve.

AUC-ROC CURVE:

A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

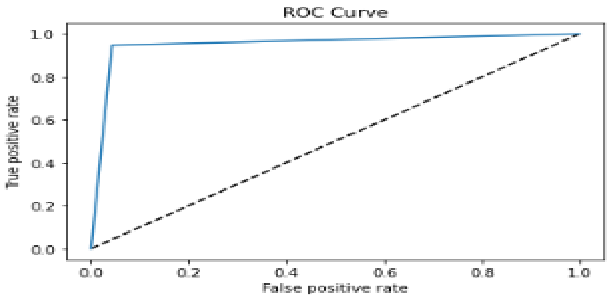
```python
from sklearn.metrics import roc_curve, roc_auc_score
from sklearn.metrics import plot_roc_curve
```

```python
fpr,tpr,threshold=roc_curve(y_test,best_model_pred)
```

```python
print('False positive rate =',fpr)
print('True positive rate = ',tpr)
print('threshold = ',threshold)
```

```
False positive rate = [0.         0.04288215 1.        ]
True positive rate =  [0.         0.94754668 1.        ]
threshold =  [2 1 0]
```

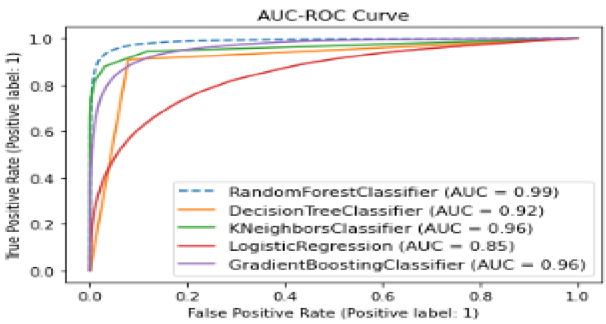
```python
plt.plot([0,1],[0,1],'k--')
plt.plot(fpr,tpr,label='ROC Curve')
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.title('ROC Curve')
plt.show()
```



-I plot graph between all of the models and we found the best model for Micro-Credit Defaulter is Random Forest Classifier with AUC=0.99

```python
dist=plot_roc_curve(rfc,x_test,y_test,linestyle='--')
plot_roc_curve(dr,x_test,y_test,ax=dist.ax_)
plot_roc_curve(knn,x_test,y_test,ax=dist.ax_)
plot_roc_curve(lg,x_test,y_test,ax=dist.ax_)
plot_roc_curve(gbc,x_test,y_test,ax=dist.ax_)
plt.title("AUC-ROC Curve")

plt.legend(prop={'size':11},loc='lower right')
plt.show()
```



The best model for Micro-Credit Defaulter is Random Forest Classifier with AUC=0.99

CONCLUSION

- Key Findings and Conclusions of the Study

Microfinance institutions play a major role in economic development in many developing countries. However many of these microfinance instit utions are faced with the problem of default because of the non-formal nature of the business and individuals they lend money. This study seeks to find the determinants of credit default in microfinance institutions. With data on 209592 rows and 37 Columns we proposed a best model for Micro-Credit Defaulter is **Random Forest Classifier** wi th AUC=0.99 model to predict the probability of default. Microfinance institutions could use this model to screen prospective loan applicants i n order to reduce the level of default.

- Learning Outcomes of the Study in respect of Data Science

➢ First I define the business problem and done EDA(Exploratory data analysis), this dataset having Skewness (measure of the asymmetry of a distribution.) and having outliers(observation that lies an abnormal distance from other values in a random sample from a population.)

➢ I keep all necessary dataset and drop all unnecessary data.

➢ Cleaning the data

Removing unwanted data points—extracting irrelevant observations that have no bearing on intended analysis.

Bringing structure to data—fixing layout issues, which will help to map and manipulate this data more easily.

Filling in major gaps—this data contains null values and I notice that important data are missing. Once we identified gaps, we can go about filling them.

➢ Analysing the dataset

➢ Sharing the predictions

Prediction

```
df=pd.DataFrame([loaded_model.predict(X_test)[:],y_test[:]],index=['Predicted','Actual'])
df
```

	0	1	2	3	4	5	6	7	8	9	...	110049	110050	110051	110052	110053	110054	110055	110056	110057	110058
Predicted	1	1	1	0	1	0	1	0	1	0	...	0	0	0	0	0	1	0	0	0	1
Actual	1	1	1	0	1	0	1	0	1	0	...	0	0	0	0	0	1	0	0	0	1

2 rows × 110059 columns

Saving the Best Model

]:
```
import pickle

filename = 'Micro_Credit_Defaulter_ML.pkl'

pickle.dump(best_model, open(filename,'wb'))

loaded_model = pickle.load(open(filename,'rb'))
```

- Limitations of this work and Scope for Future Work

-The data is interesting, as it contains more feature and the volume is huge. It takes time for visualization of distribution plot of all features.

-I used various visualization plots which helped me to understand the data.

- Dropped the unnecessary columns which having the more zero values. It helps to avoid the Multicollinearity (statistical concept where several independent variables in a model are correlated.)

- I have used the 5-machine learning algorithm to make the prediction for Micro Credit Defaulter project

- We can also analyze the dataset by observing age, gender, gross monthly income, and tenure with the current employer, loan amount, and the tenor of loan, number of dependents, other income, and other deductions were important determinants of default.