**Micro Credit — Defaulter detection using machine learning**

**Introduction**

Microcredit is a common form of microfinance that involves an extremely small loan given to an individual to help them become self-employed or grow a small business. These borrowers tend to be low-income individuals, especially from less developed countries (LDCs). Microcredit is also known as "microlending" or "microloan."

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

**Data Analysis**

In this project, we have a dataset which has the details of Loan taken.

The given dataset contains 209593 rows and 37 columns. The column names like daily\_decr30, rental30, amnt\_loans, payback30,etc.

Unnamed: 0 int64

label int64

msisdn object

aon float64

daily\_decr30 float64

daily\_decr90 float64

rental30 float64

rental90 float64

last\_rech\_date\_ma float64

last\_rech\_date\_da float64

last\_rech\_amt\_ma int64

cnt\_ma\_rech30 int64

fr\_ma\_rech30 float64

sumamnt\_ma\_rech30 float64

medianamnt\_ma\_rech30 float64

medianmarechprebal30 float64

cnt\_ma\_rech90 int64

fr\_ma\_rech90 int64

sumamnt\_ma\_rech90 int64

medianamnt\_ma\_rech90 float64

medianmarechprebal90 float64

cnt\_da\_rech30 float64

fr\_da\_rech30 float64

cnt\_da\_rech90 int64

fr\_da\_rech90 int64

cnt\_loans30 int64

amnt\_loans30 int64

maxamnt\_loans30 float64

medianamnt\_loans30 float64

cnt\_loans90 float64

amnt\_loans90 int64

maxamnt\_loans90 int64

medianamnt\_loans90 float64

payback30 float64

payback90 float64

pcircle object

pdate object

Data Types of the dataset

There is no null values in the dataset.

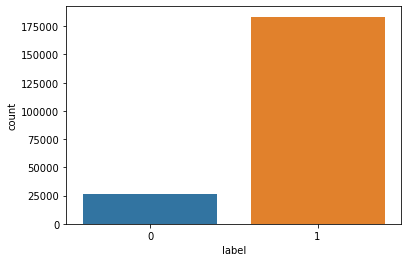
Exploratory data analysis

**Dependent variable:** Exploratory data analysis was conducted starting with the dependent variable, defaulter\_reported. There were 26162 defaulter and 183431 non-defaulter.

1 183431

0 26162

Name: label, dtype: int64



**Correlations among variables:**Heatmap was plotted for variables with at least 0.3 Pearson’s correlation coefficient, including the DV. Daily amount spent from main account, averaged over last 30 days and Daily amount spent from main account, averaged over last 90 days had a correlation of 0.98 ,rental30 and rental90 had a correlation of 0.96 and Total amount of loans taken by user in last 30 days and Number of loans taken by user in last 30 days had a correlation of 0.96 .

**Visualizing variables:**Number of times main account got recharged in last 30 days 1 time for more than 35000 rupees most of the time recharged with less than 5000 rupees..

1 37238

2 31216

0 27979

3 25519

4 20258

...

78 1

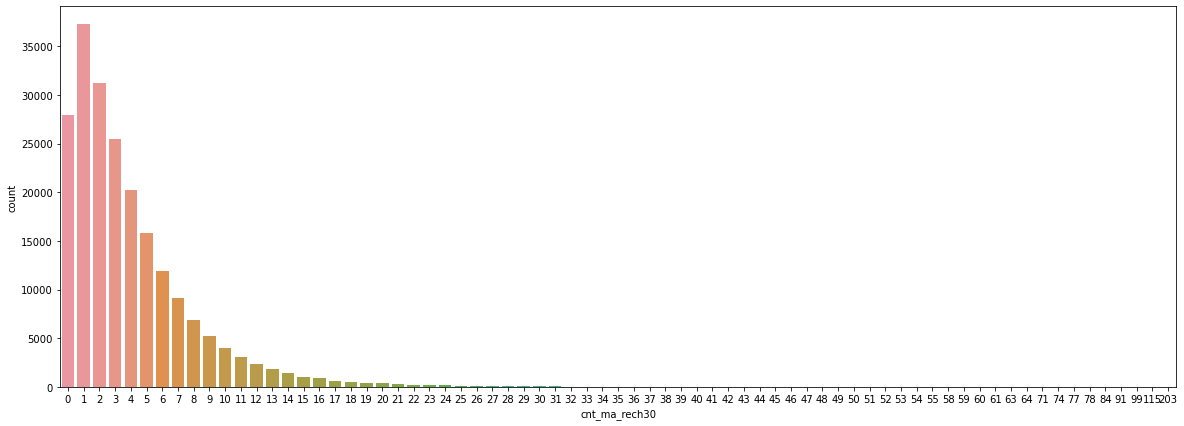
84 1

91 1

99 1

115 1

Name: cnt\_ma\_rech30, Length: 71, dtype: int64



Number of times main account got recharged in last 90 days 1 time for more than 25000 rupees most of the time recharged with less than 5000 rupees..

1 27898

2 24052

0 20950

3 20446

4 17329

...

336 1

151 1

132 1

140 1

127 1

Name: cnt\_ma\_rech90, Length: 110, dtype: int64



Frequency of main account recharged in last 30 days 1 time for more than 2000 rupees and most of the time less than 5000 ruppes*.*

0 65753

1 24373

2 19285

3 13192

4 10021

...

80 7

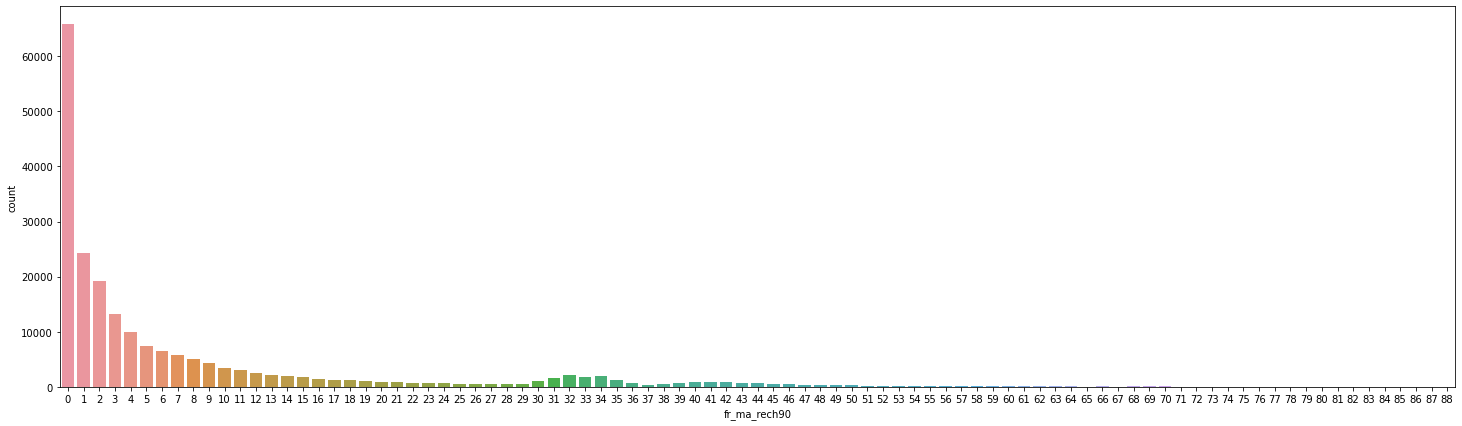
81 7

88 5

84 4

87 1

Name: fr\_ma\_rech90, Length: 89, dtype: int64

****

Number of times data account got recharged in last 90 days 1 time with less 2500 rupees

****

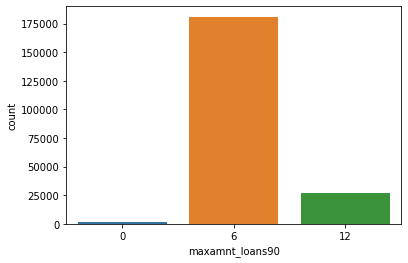
maximum amount of loan taken by the user in last 90 days

6 180945

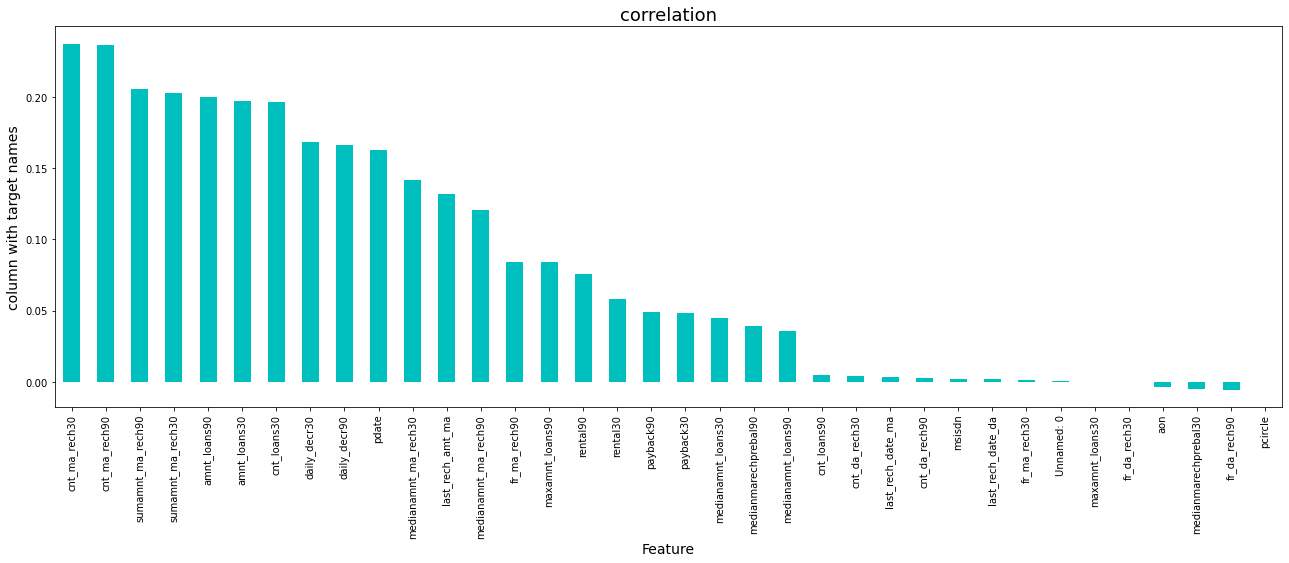
12 26605

0 2043

Name: maxamnt\_loans90, dtype: int64



**Checking correlation between dependent and independent variables**

****

## Pre-processing Pipeline

Data preprocessing is a predominant step in machine learning to yield highly accurate and insightful results. Greater the quality of data, the greater is the reliability of the produced results. **Incomplete, noisy, and inconsistent data** are the inherent nature of real-world datasets. Data preprocessing helps in increasing the quality of data by filling in missing incomplete data, smoothing noise, and resolving inconsistencies.

**Incomplete data** can occur due to many reasons. Appropriate data may not be persisted due to a misunderstanding, or because of instrument defects and malfunctions.

**Noisy data** can occur for a number of reasons (having incorrect feature values). The instruments used for the data collection might be faulty. Data entry may contain human or instrument errors. Data transmission errors might occur as well.

There are many stages involved in data preprocessing.

**Data cleaning** attempts to impute missing values, removing outliers from the dataset.

**Data integration**integrates data from a multitude of sources into a single data warehouse.

**Data transformation**such as normalization, may be applied. For example, normalization may improve the accuracy and efficiency of mining algorithms involving distance measurement.

**Data reduction**can reduce the data size by dropping out redundant features. Feature selection and feature extraction techniques can be used.

**Treating null values**

Sometimes there are certain columns which contain the null value used to indicate missing or unknown values or maybe the value doesn’t exist. There are no null values in our dataset.

There are different ways of replacing null values from the dataset, like replacing with mean , mode or median of the column.

**Converting labels into numeric**

In machine learning, we usually deal with datasets which contain multiple labels in one or more than one column. These labels can be in the form of words or numbers. To make the data understandable or in human readable form, the training data is often labeled in words.

In our data there are columns with categorical values. The columns like cnt\_ma\_rech30, last\_rech\_amt\_ma, cnt\_ma\_rech90, fr\_ma\_rech90, sumamnt\_ma\_rech90, cnt\_da\_rech90, fr\_da\_rech90, cnt\_loans30, amnt\_loans30, amnt\_loans90, maxamnt\_loans90. There are 2 columns with object datatype like pcircle and pdate. These columns have to be treated with one hot encoding or the label encoder.

**Label Encoder**refers to converting the labels into numeric form so as to convert it into the machine readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important preprocessing step for the structured dataset in supervised learning.

Label encoding in python can be imported from the Sklearn library. Sklearn provides a very efficient tool for encoding. Label encoders encode labels with a value between 0 and n\_classes-1.

**Outliers handling**

**Outliers**are data points that are distant from other similar points. They may be due to variability in the measurement or may indicate experimental errors. If possible, outliers should be excluded from the data set. However, detecting that anomalous instance might be very difficult, and is not always possible.

Methods to remove outliers:

**Z-score —**Call scipy.stats.zscore() with the given data-frame as its argument to get a numpy array containing the z-score of each value in a dataframe. Call numpy.abs() with the previous result to convert each element in the data frame to its absolute value. Use the syntax (array < 3).all(axis=1) with array as the previous result to create a Boolean array.

**Balancing our imbalanced data**

There are different algorithms present to balance the target variable. We have used the SMOTE() algorithm to make our data balance.

**NOTE:** SMOTE(Synthetic minority oversampling technique) works by randomly picking a point from the minority class and computing the k-nearest neighbors of this point. The synthetic points are added between the chosen point and its neighbors.

**SMOTE algorithm works in 4 simple steps:**

1. Choose a minority class as input vector.
2. Find its k-nearest neighbors.
3. Choose one of these neighbors and place a synthetic point anywhere on the line joining the point under consideration and its chosen neighbors.
4. Repeat the step until the data is balanced.

from imblearn.over\_sampling import SMOTE

smt=SMOTE()

x,y=smt.fit\_resample(x,y)

The original shape of our data was 183431 for non defaulter and 26162 defaulter. The SMOTE algorithm balances our data with the highest number of values present in it.

**Building machine learning models**

For building machine learning models there are several models present inside the Sklearn module.

Sklearn provides two types of models i.e. regression and classification. Our dataset’s target variable is to predict whether fraud is reported or not. So for this kind of problem we use classification models.

But before fitting our dataset to its model first we have to separate the predictor variable and the target variable, then we pass this variable to the train\_test\_split method to create a random test and train subset.

**What is train\_test\_split**, itis a function in sklearn model selection for splitting data arrays into two subsets for training data and testing data. With this function, you don’t need to divide the dataset manually. By default, sklearn train\_test\_split will make random partitions for the two subsets. However, you can also specify a random state for the operation. It gives four outputs x\_train, x\_test, y\_train and y\_test. The x\_train and x\_test contains the training and testing predictor variables while y\_train and y\_test contains the training and testing target variable.

After performing train\_test\_split we have to choose the models to pass the training variable.

We can build as many models as we want to compare the accuracy given by these models and to select the best model among them.

I have selected 5 models:

* **Logistic Regression from sklearn.linear\_model:** Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is binary, which means there would be only two possible classes 1 (stands for success/yes) or 0 (stands for failure/no). Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

Acuracy\_score 0.7957626068640813

Confusion matrix [[51796 8638]

[16088 44543]]

precision recall f1-score support

0 0.76 0.86 0.81 60434

1 0.84 0.73 0.78 60631

accuracy 0.80 121065

macro avg 0.80 0.80 0.80 121065

weighted avg 0.80 0.80 0.80 121065

* **DecisionTreeClassifier from sklearn.tree:** Decision trees can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions. The two main entities of a tree are decision nodes, where the data is split and leaves, where we get the outcome.

Acuracy\_score 0.9079750547226696

Confusion matrix [[55115 5319]

[ 5822 54809]]

precision recall f1-score support

0 0.90 0.91 0.91 60434

1 0.91 0.90 0.91 60631

accuracy 0.91 121065

macro avg 0.91 0.91 0.91 121065

weighted avg 0.91 0.91 0.91 121065

* **RandomForestClassifier from sklearn.ensemble:** As we know that a forest is made up of trees and more trees means more robust forest. Similarly, a random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Acuracy\_score 0.9447982488745715

Confusion matrix [[56655 3779]

[ 2904 57727]]

precision recall f1-score support

0 0.95 0.94 0.94 60434

1 0.94 0.95 0.95 60631

accuracy 0.94 121065

macro avg 0.94 0.94 0.94 121065

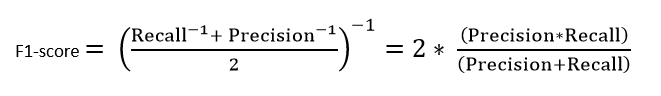
weighted avg 0.94 0.94 0.94 121065

# ****Conclusion from models****

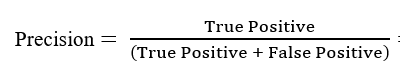
We got our best model i.e. RandomForestClassifier with the accuracy score of 94.47%.

**Understand what does precision recall and f1 score and accuracy do**

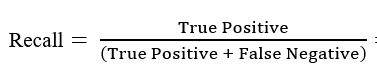
* **F1 score**: this is the harmonic mean of precision and recall and gives a better measure of the incorrectly classified cases than the accuracy matrix.



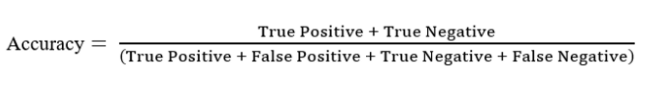
* **Precision:**It is implied as the measure of the correctly identified positive cases from all the predicted positive cases. Thus, it is useful when the costs of False Positives are high.



* **Recall:**It is the measure of the correctly identified positive cases from all the actual positive cases. It is important when the cost of False Negatives is high.



* **Accuracy:**One of the more obvious metrics, it is the measure of all the correctly identified cases. It is most used when all the classes are equally important.



# ****Cross Validation****

Cross-validation is **a resampling procedure used to evaluate machine learning models on a limited data sample**. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. We need provide the value of K to divide the group in same number.

In cross Validation we are also getting Random Forest classifier accuracy score is high which is 93.78%

# ****Hyper parameter tuning****

Hyper parameter optimization in machine learning intends to find the hyper parameters of a given machine learning algorithm that deliver the best performance as measured on a validation set. Hyper parameters, in contrast to model parameters, are set by the machine learning engineer before training. The number of trees in a random forest is a hyper parameter while the weights in a neural network are model parameters learned during training. I like to think of hyper parameters as the model settings to be tuned so that the model can optimally solve the machine learning problem.

We will use GridSearchCV for the hyper parameter tuning.

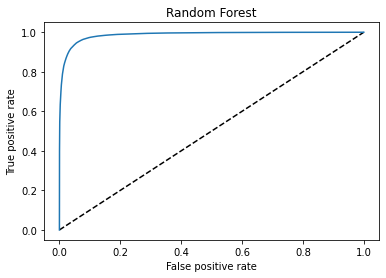
**GridSearchCV**

In the GridSearchCV approach, the machine learning model is evaluated for a range of hyper parameter values. This approach is called GridSearchCV, because it searches for best set of hyper parameters from a grid of hyper parameters values.

We got Random Forest Classifier with the accuracy score of 93.47% after Hyper parameter Tuning

**ROC curve:**Itis a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0s as 0s and 1s as 1s. By analogy, the Higher the AUC, the better the model is at distinguishing between patients with the disease and no disease.

The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.



# ****Remarks****

This project has built a model that can detect auto insurance fraud. In doing so, the model can reduce losses for insurance companies. The challenge behind fraud detection in machine learning is that frauds are far less common as compared to legit insurance claims.

3 different classifiers were used in this project: logistic regression, Random forest, Decision tree. Five different ways of handling imbalance classes were tested out with these 3 classifiers: model with class weighting, oversampling with SMOTE, Cross validation, hyper parameter tuning, and plotting roc curve of the models.

The best and final fitted model was a weighted **Random Forest** that yelled a F1 score of 0.94 and a ROC AUC of 0.95. In conclusion, the model was able to correctly distinguish between fraud claims and legit claims with high accuracy.