
# MICRO CREDIT DEFAULTER PROJECT

Submitted by:

SANKALP GUPTA

# ACKNOWLEDGMENT

I would like to express special thanks to Sir. Nishant Kadian. For providing detailed case study and data set. Which includes detailed explanation of feature properties and other supporting documents.

References :

i. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/
ii. https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SVMSMOTE.html

# INTRODUCTION

- ## Business Problem Framing

  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. 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.

- ## 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.

- ## Review of Literature

"Microfinance" is often seen as financial services for poor and low-income clients(Ayayi, 2012; Mensah, 2013; Tang, 2002). In practice, the term is often used more narrowly to refer to loans and other services from providers that identify themselves as "microfinance institutions" (MFIs) [Consultative Group to Assist the Poor (CGAP) 2010]. Microfinance can also be described as a setup of a number of different operators focusing on the financially under-served people with the aim of satisfying their need for poverty alleviation, social promotion, emancipation, and inclusion. Microfinance institutions reach and serve their target market in very innovative ways (Milana 2012).

Microfinance operations differ in principle, from the standard disciplines of general and entrepreneurial finance. This difference can be attributed to the fact that the size of the loans granted with microcredit is typically too small to finance growth-oriented business projects. The CGAP (2010) identifies some unique features of microfinance as follows;

- ## Motivation for the Problem Undertaken

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 (Acquah & Addo, 2011).

# Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

Output/target variable is an categorical variable, it's an classification problem.

We have several classification Machine learning algorithms developed by researchers i.e. Logistic regression, Decision Tree, KNN etc.

i. Logistic Regression:

Logistic regression analysis was first used to predict default probability by Ohlson (1980) and since then several other researchers have also used Logistic regression to predict default (Altman & Sabato, 2007; Jain, Gupta, & Mittal, 2011; Laitinen, 2010; Lugovskaya, 2010; Westgaard & Wijst, 2001).

Logistic Regression Analysis (LRA) does not make assumptions of multivariate normality also Variance-Covariance matrix of the independent variables can be different for defaulting and non-defaulting customers. The logistic regression model offer the following advantages (1) fits the problem of default prediction, that is, Logistic regression has a score between 0 and 1 conveniently giving us the probability of default. (2) LRA models allow us to model dichotomous dependent variable in our case default and non-default. Lastly (3) estimated coefficients of the LRA model can show the relative importance of particular independent variable.

The equation for Logistic Regression can be generalized as

$$ln\left(\frac{p}{1-p}\right) = b_0 + b_1 x$$

As mentioned before, logistic regression can handle any number of numerical and/or categorical variables.

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p)}}$$

ii. KNN (K-Nearest Neighbor):

K acts as hyperparameter in KNN algorithm. Get the k nearest neighbors to query/output point.

Now, how to determine class 1 or 0 for query point? Go for majority vote amongst neighbors.

Ideal choice for K will be and odd number in case of classification problem. Distance metric can be any amongst i.e. Euclidean distance, Manhattan, Minkowski, Hamming and cosine distance.
Cosine distance is the most preferred distance as it computes normal distance between two points.

iii. Decision Tree:

Decision trees are one of the most commonly used predictive modelling algorithms in practice. The reasons for this are many. Some of the distinct advantages of using decision trees in many classification and prediction applications are explained below along with some common pitfalls.

Decision trees typically consist of three different elements:

Root Node:
This top-level node represents the ultimate objective, or big decision you're trying to make.

Branches:
Branches, which stem from the root, represent different options—or courses of action—that are available when making a particular decision. They are most commonly indicated with an arrow line and often include associated costs, as well as the likelihood to occur.

Leaf Node:
The leaf nodes—which are attached at the end of the branches—represent possible outcomes for each action. There are typically two types of leaf nodes: square leaf nodes, which indicate another decision to be made, and circle leaf nodes, which indicate a chance event or unknown outcome.

Entropy:

Entropy is amount of information is needed to accurately describe the some sample. So if sample is homogeneous, means all the element are similar than Entropy is 0, else if sample is equally divided than entropy is maximum 1.

$$Entropy = -\sum_{i=1}^{n} p_i * \log(p_i)$$

Gini index / Gini impurity:

Gini index is measure of inequality in sample. It has value between 0 and 1. Gini index of value 0 means sample are perfectly homogeneous and all element are similar, whereas, Gini index of value 1 means maximal inequality among elements. It is sum of the square of the probabilities of each class. It is illustrated as,

$$Gini\,index = 1 - \sum_{i=1}^{n} p_i^2$$

i is number of classes

- Data Sources and their formats

  Our data was obtained from one of the leading microfinance companies in Indonesia, data is on Telecom Industry which provides which is providing micro credit on mobile balances.

  The data explains customers past credit history, mostly related to last 30days and 90 days. The data consists credit history details of applicants applied between June 2016 and August 2016. Within this period the microfinance company received 2,09,593 loan applications from individuals across the country. Our data contains information on only individuals that were granted the loan and does not contain those who were denied.

  Customers were marked as defaulters, if the customer failed in 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.

  Screenshot of data is attached below.

| | label | msisdn | aon | daily_decr30 | daily_decr90 | rental30 | rental90 | last_rech_date_ma | last_rech_date_da | last_rech_amt_ma |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 21408I70789 | 272 | 3055.05 | 3065.15 | 220.13 | 260.13 | 2 | 0 | 1539 |
| 2 | 1 | 76462I70374 | 712 | 12122 | 12124.75 | 3691.26 | 3691.26 | 20 | 0 | 5787 |
| 3 | 1 | 17943I70372 | 535 | 1398 | 1398 | 900.13 | 900.13 | 3 | 0 | 1539 |

- Data Pre-processing Done:
    i. Missing Value Analysis:

    Missing values in data is a common real world problem, we face while analysing and fitting the model. It's possible occurrences of missing values due to human error or was not available at the first place.

    In our case, no missing values were present.

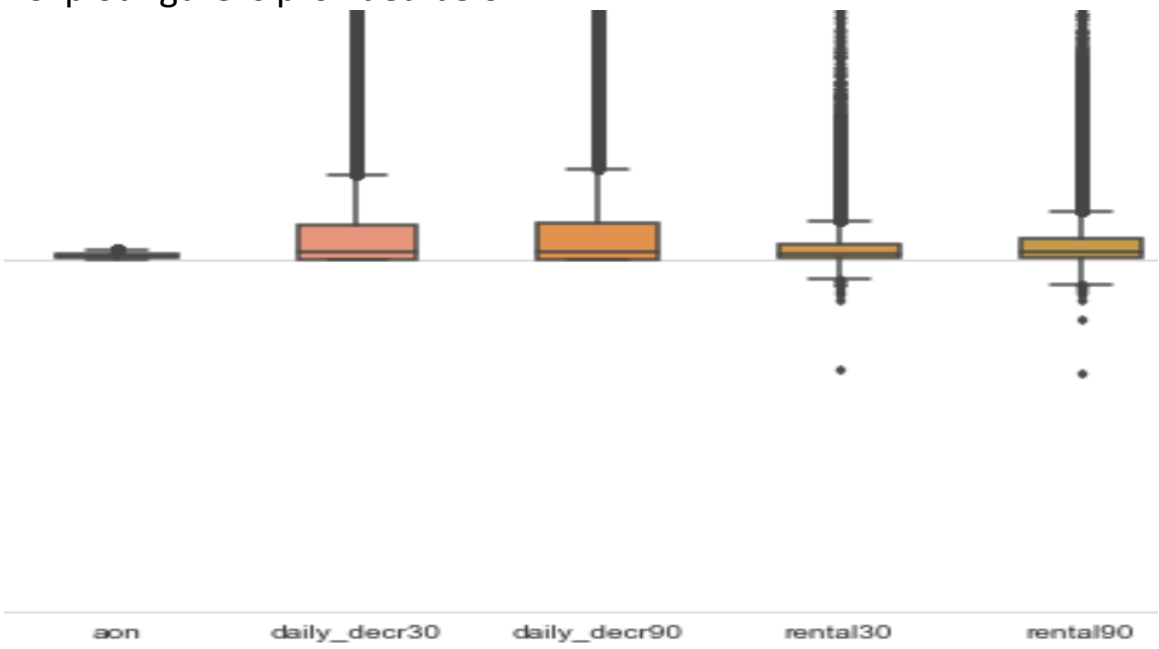
    ii. Outlier Analysis:

    This is one of the most important data pre-processing parts which needs good understanding of the data and careful analysis. Outliers are the points which are far away from the other observations/or from their distribution.

    We are going to observe outliers with the help of Box-Plot.

    Most of the features contains outliers in this data set except features pdate, pcircle, Unnamed:0, label and msisdn.

    Boxplot figure is provided below.



## iii. Feature Selection:

When the number of features are very large. We can't visualize or create correlation heat map to observe which features are important and which are not. In our case we have known that have only 37 features out of which 35 are continuous features and 2 categorical among which one is output variable 'label'(Two possibilities 0 and 1).

Observations:

a. It's clearly visible through the heat map that most of the features are highly correlated to each other..
b. Features mentioned below are highly correlated to each other and we will reduce the dimensions
    1. daily_decr30 and daily_decr90
    2. rental30 and rental90
    3. cnt_ma_rech30 and cnt_ma_rech90
    4. sumamnt_ma_rech30 and sumamnt_ma_rech90
    5. medianamnt_ma_rech30 and medianamnt_ma_rech90
    6. maxamnt_loans30 and maxamnt_loans90
    7. medianamnt_loans30 and medianamnt_loans90
    8. medianamnt_ma_rech30 and last_rech_amt_ma

## iv. Feature scaling:

Feature scaling is one of the most important pre-processing techniques. It majorly involves two techniques named as Normalization and standardization.

It's important to rescale features else it may lead to wrong predictions, especially in the case of algorithms having distance metrics within.

Rescaling data between 0 and 1 is known as feature scaling. In our case we will normalize all the features in giving data.

Mathematical explanation of normalization is below:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Where:
x = active point to be normalized
x(min)=minimum in the target column
x(max)=maximum point in the target column

- ## Data Inputs- Logic- Output Relationships

Data file contains approx. 209593 points and 37 features. We have features giving information of each customers transaction history, includes loan taken and repaid.

Output feature 'label' is categorical variable having 1 and 0, where 1 represents loan has been repaid within 5 days(Non-defaulter), whereas 0 represents, who failed to repay loan within 5 days(Defaulter).

Where, features like payback30/90(Average payback time in 30/90 days), cnt_loans30/90 are few which are highly correlated to output feature as they explains how well they have been repaying loans taken.

- ## Hardware and Software Requirements and Tools Used

Analysed and developed model for data on Anaconda on Mac OS.

Following libraries and packages were used to get meaningful information.

Pandas to read csv file as DataFrame.
Seaborn and matplotlib to visualize the data.
Sklearn library for metrics and ML Algorithms.
Imblearn for oversampling the data.

# Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  Our task is to predict the new customer is how likely to be a defaulter. We have two possible outcomes either it's defaulter i.e. 1 or not an defaulter.

  Clearly, we have only two possible outputs. So, it's an classification problem.

- Testing of Identified Approaches (Algorithms)

  Algorithms used for training and testing purpose.

  i. AdaBoost.
  ii. Gradient Boosting
  iii. KNN
  iv. Decision Tree
  v. Logistic Regression
     Detailed mathematical explanation given in section "Mathematical/Analytical modelling of the problem" section.

- Run and Evaluate selected models

  Used accuracy score in order to select best models and later part consists of cross validation and getting best parameters for model giving best accuracy score at the first stage.

  Below are the code snippets and accuracy score respectively:

  Decision Tree Classifier:

```
max_accScore=0
active_as=0
iBest_rs=0
# cm: classification model
cm=DecisionTreeClassifier()
for iActive_rs in range(42,101):
    x_train,x_test,y_train,y_test=train_test_split(X,Y,random_state=iActive_rs,test_size=0.2)
    cm.fit(x_train,y_train)
    pred=cm.predict(x_test)
    active_as=accuracy_score(pred,y_test)
    if active_as>max_accScore:
        max_accScore=active_as
        iBest_rs=iActive_rs
print("Best(max) accuracy score is {} for random state {}.".format(max_accScore,iBest_rs))
```

## KNN:

```
max_accScore=0
active_as=0
iBest_rs=0
# cm: classification model
cm=KNeighborsClassifier()
for iActive_rs in range(42,101):
    x_train,x_test,y_train,y_test=train_test_split(X,Y,random_state=iActive_rs,test_size=0.2)
    cm.fit(x_train,y_train)
    pred=cm.predict(x_test)
    active_as=accuracy_score(pred,y_test)
    if active_as>max_accScore:
        max_accScore=active_as
        iBest_rs=iActive_rs
print("Best(max) accuracy score is {} for random state {}.".format(max_accScore,iBest_rs))
```

## Logistic Regression:

```
max_accScore=0
active_as=0
iBest_rs=0
# cm: classification model
cm=LogisticRegression()
for iActive_rs in range(42,101):
    x_train,x_test,y_train,y_test=train_test_split(X,Y,random_state=iActive_rs,test_size=0.2)
    cm.fit(x_train,y_train)
    pred=cm.predict(x_test)
    active_as=accuracy_score(pred,y_test)
    if active_as>max_accScore:
        max_accScore=active_as
        iBest_rs=iActive_rs
print("Best(max) accuracy score is {} for random state {}.".format(max_accScore,iBest_rs))
```

## Ada Boost Classifier:

```
max_accScore=0
active_as=0
iBest_rs=0
# cm: classification model
cm=AdaBoostClassifier()
for iActive_rs in range(42,101):
    x_train,x_test,y_train,y_test=train_test_split(X,Y,random_state=iActive_rs,test_size=0.2)
    cm.fit(x_train,y_train)
    pred=cm.predict(x_test)
    active_as=accuracy_score(pred,y_test)
    if active_as>max_accScore:
        max_accScore=active_as
        iBest_rs=iActive_rs
print("Best(max) accuracy score is {} for random state {}.".format(max_accScore,iBest_rs))
```

Gradient Boosting Classifier:

```
max_accScore=0
active_as=0
iBest_rs=0
# cm: classification model
cm=GradientBoostingClassifier()
for iActive_rs in range(42,101):
    x_train,x_test,y_train,y_test=train_test_split(X,Y,random_state=iActive_rs,test_size=0.2)
    cm.fit(x_train,y_train)
    pred=cm.predict(x_test)
    active_as=accuracy_score(pred,y_test)
    if active_as>max_accScore:
        max_accScore=active_as
        iBest_rs=iActive_rs
print("Best(max) accuracy score is {} for random state {}.".format(max_accScore,iBest_rs))
```

Respective accuracy scores and random states:

| Algorithm | Random State | Accuracy Score |
|---|---|---|
| Decision Tree | 62 | 86 |
| KNN | 81 | 88 |
| Logistic Regression | 47 | 88 |
| AdaBoost | 67 | 90 |
| Gradient Boost | 67 | 91 |

- Key Metrics for success in solving problem under consideration

As we are developing a classification model to identify defaulter and non-defaulter.

| | | Actual | |
|---|---|---|---|
| | | Positive | Negative |
| Predicted | Positive | True Positive | False Positive |
| | Negative | False Negative | True Negative |

Key metrics used to identify generalized model are:

i. Accuracy score= n/N

Where, n = Points correctly identified

N = Total points(Total number of rows)

ii. Precision:

Precision = TP/(TP+FP)

Where, TP = True positive

FP = False positive

iii. Recall:

Recall = TP/(TP+FN)

Where, TP = True positive

FN = False Negative

iv. F1 Score: It's a number between 0 and 1, it's an harmonic mean of precision and recall.

It makes sure that we have good precision and recall. If accuracy score is good and F1 score is poor, then good accuracy score is worthless for our case.

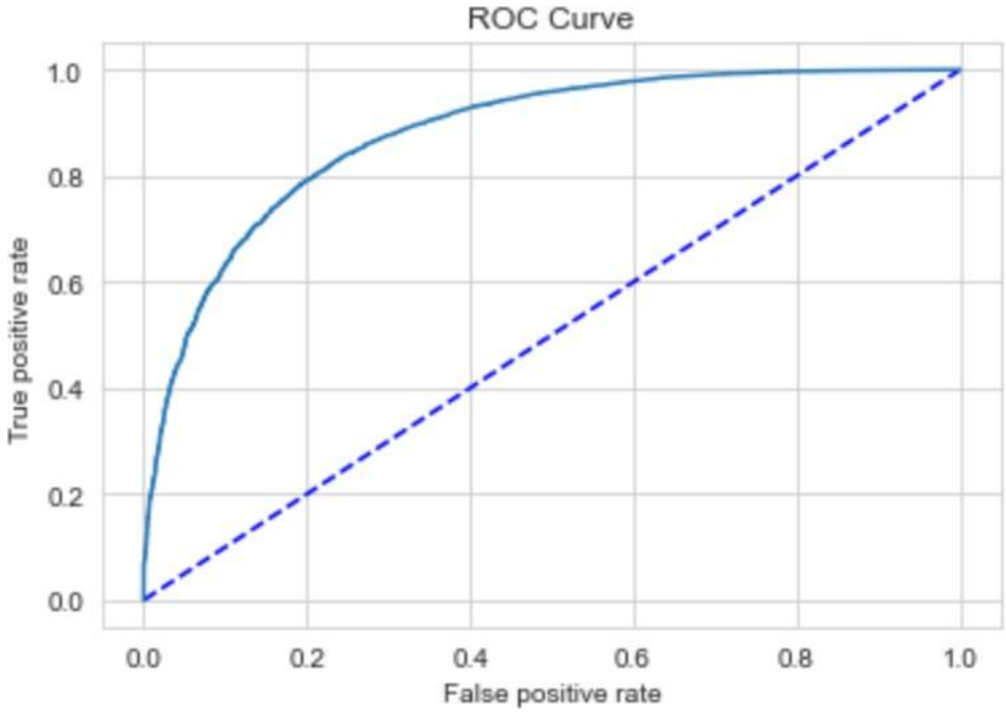
$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

v. AUC ROC:

AUC ROC indicates how well the probabilities from the positive classes are separated from the negative classes.

We have got the probabilities from our classifier. We can use various threshold values to plot our sensitivity(TPR) and (1-specificity)(FPR) on the cure and we will have a ROC curve.

Sensitivity = TPR(True Positive Rate) = Recall = TP/(TP+FN)
1-specificity = FPR(False Positive Rate) = FP/(TN+FP)




Note: We are getting AUC score as .6742, which not considered as the best or good, but an acceptable score.

- Interpretation of the Results

Outliers were present in almost all of the features, in features like daily_decr30, rental30 negative values were present. Which is not possible in real life.

Converted negative values to positive and observed the distribution, result distribution were not similar. So, from all the observed feature having negative values, removed the outliers.

Gradient Boosting algorithm is giving 91% accuracy with auc score 67 which is less.

Whereas, Logistic Regression is giving 87% accuracy with .76 auc score, which is a good score. But having False positive report is high.

I am ending up with Gradient Boosting Algorithm to release the trained generalized model.

Note: Both models were trained after oversampling the train data as the dataset is an imbalanced dataset. I have used SVMSMOTE algorithm to oversample the minority class of defaulters customers.

# CONCLUSION

Microfinance is being a globally accepted to help rural and productive poor with micro credits through banking services. The only drawback to this is if customers are not returning the loan in time it reduces the confidence of investors and help can't be reached to potential people in hard times.

To overcome this we need to develop model to identify such defaulters i.e. investors face minimal for this good work.

In current study, Gradient Boosting algorithm is best fit model and has been deployed to identify factors associated in differentiating defaulters and non-defaulters.

Analysis showed that features 'aon', 'daily_decr30', 'rental30', 'last_rech _date_ma', 'cnt_ma_rech30', 'fr_ma_rech30', 'sumamnt_ma_rech30', 'm edianamnt_ma_rech30', 'medianmarechprebal30', 'cnt_loans30', 'amnt_l oans30', 'maxamnt_loans30', 'medianamnt_loans30', 'cnt_loans90', 'amnt _loans90', 'payback30' are important determinants of default.