FLIP ROBO

Micro Credit Defaulter Project

Submitted by:

Rutuja Patil

ACKNOWLEDGMENT

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. 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.

INTRODUCTION

- Business Problem Framing

  This is a classic Business problem which helps Micro Financing Institutions and other Lending companies reduce Credit risks by recognizing potential Defaulters.

- Conceptual Background of the Domain Problem

  Before advancement of Data Science, loan lending companies used to risk a high rate of defaulting. Many a times a perfect candidate would display erratic financial and repayment behaviour after being approved for loan. Machine Learning can help lenders predict potential defaulters before approving their candidature using their past data. The candidates' income, past debt and repayment behaviour can be important metrics for the same.

- Review of Literature

  "Microfinance" is often seen as financial services for poor and low-income clients. In practice, the term is often used more narrowly to refer to loans and other services from providers that identify themselves as "microfinance institutions". 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

Analytical Problem Framing

- Data Preprocessing Done

  1.Data Exploration

  2.Data cleaning

  3.Data Visualization

  4.Feature Selection

  5.Build model

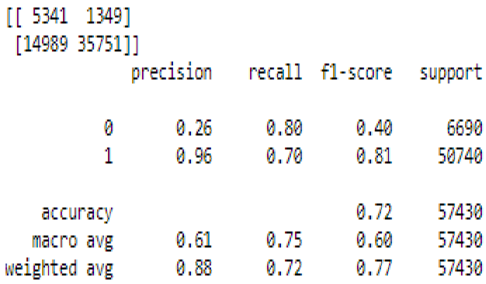
- Hardware and Software Requirements and Tools Used
  - a. Jupyter Notebook
  - b. Libraries —

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier,
ExtraTreesClassifier, GradientBoostingClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis,
QuadraticDiscriminantAnalysis
from sklearn.metrics import accuracy_score, log_loss
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler, RobustScaler
from sklearn.naive_bayes import GaussianNB
import statsmodels.api as sm
from scipy.stats import probplot
from sklearn import metrics
from sklearn import preprocessing
from sklearn import utils
from sklearn.metrics import mean_absolute_error
from statistics import mean, stdev
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score, roc_curve, auc, confusion_matrix,
plot_confusion_matrix
from sklearn.model_selection import KFold, cross_val_score, cross_validate,
StratifiedKFold
from sklearn.svm import SVC
from sklearn.feature_selection import chi2
from sklearn.feature_selection import SelectKBest
```

Models Development and Evaluation

- Data Exploration and Cleaning On data exploration, I found that the dataset was imbalanced for the target feature(87.5% for Non-defaulters and 12.5% for Defaulters). Also, I found that the data had some very unrealitic values such as 999860 days which is not possible. Also, there were negative values for variables which must not have one (example:frequency,amount of recharge etc). All these unrealistic values were dropped which caused a data loss of 8% only.
- Feature Selection Since there were 36 features, many of which I suspected were redundant because of the data duplication. It was imperative to select only most significant of them to make ML models more efficient and cost effective. The method used was 'Univariate Selection' using chi-square test. I selected top 20 features which were highly significant.
- Data Visualization On visualizing data, there were two important insights I gathered. a. Imbalance of data b. Distribution was not normal
- Data Normalization Since the data was not normal, I normalized all the features except the target variable which was dichotomous(Values '1' and '0').
- Oversampling of Minority class Since the data was expensive, I did not want to lose out on data by under sampling the majority class. Instead, I decided to oversample the minority class using SMOTE.
- Build Models Since it was a supervised classification problem, I built 5 models to evaluate performance of each of them: a. Logistic Regression b. Linear SVM c. Decision Tree d. Random forest e. Gradient Boost Classifier Since the data was imbalanced, accuracy was not the correct performance metric. Instead I focused on other metrics like precision, recall and ROC-AUC curve.

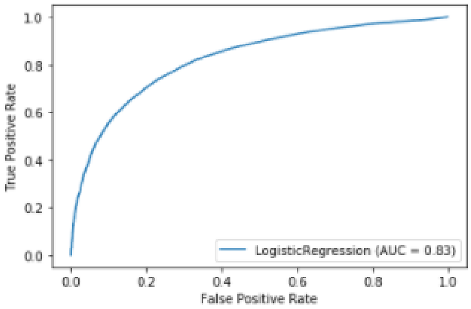
1. Logistic Regression



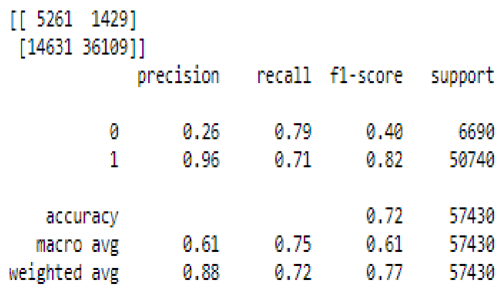
```
[[ 5341  1349]
 [14989 35751]]
              precision    recall  f1-score   support

           0       0.26      0.80      0.40      6690
           1       0.96      0.70      0.81     50740

    accuracy                           0.72     57430
   macro avg       0.61      0.75      0.60     57430
weighted avg       0.88      0.72      0.77     57430
```



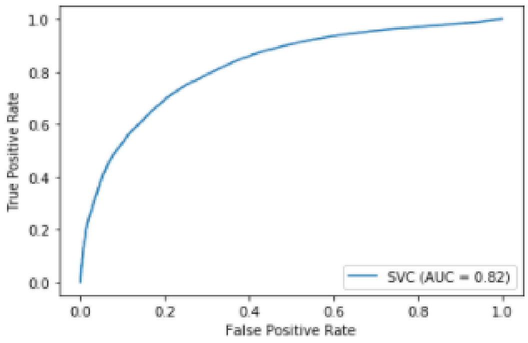
2. Linear SVM



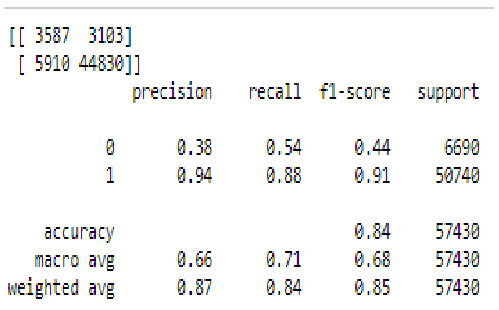
```
[[ 5261  1429]
 [14631 36109]]
              precision    recall  f1-score   support

           0       0.26      0.79      0.40      6690
           1       0.96      0.71      0.82     50740

    accuracy                           0.72     57430
   macro avg       0.61      0.75      0.61     57430
weighted avg       0.88      0.72      0.77     57430
```



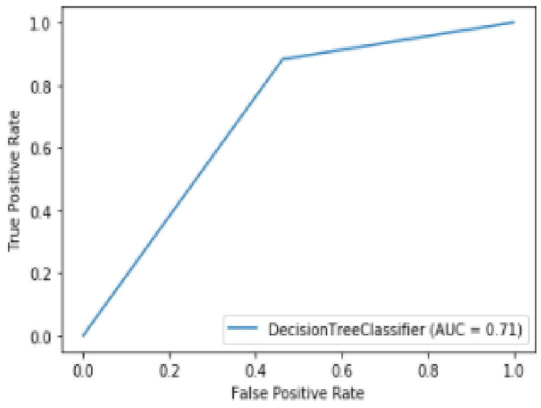
3. Decision Tree



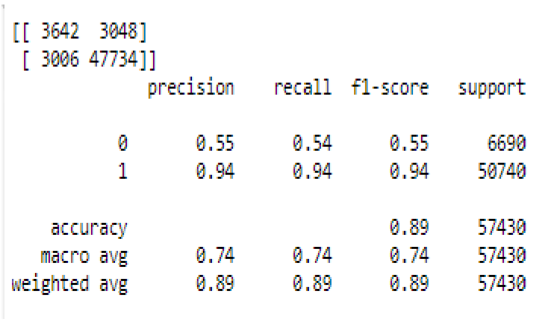
```
[[ 3587  3103]
 [ 5910 44830]]
              precision    recall  f1-score   support

           0       0.38      0.54      0.44      6690
           1       0.94      0.88      0.91     50740

    accuracy                           0.84     57430
   macro avg       0.66      0.71      0.68     57430
weighted avg       0.87      0.84      0.85     57430
```



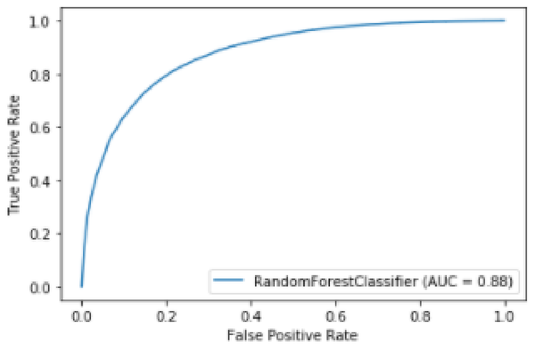
4. Random Forest



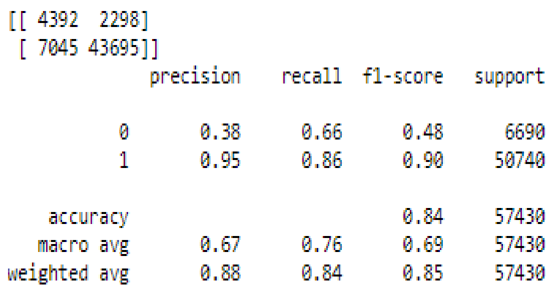
```
[[ 3642  3048]
 [ 3006 47734]]
              precision    recall  f1-score   support

           0       0.55      0.54      0.55      6690
           1       0.94      0.94      0.94     50740

    accuracy                           0.89     57430
   macro avg       0.74      0.74      0.74     57430
weighted avg       0.89      0.89      0.89     57430
```



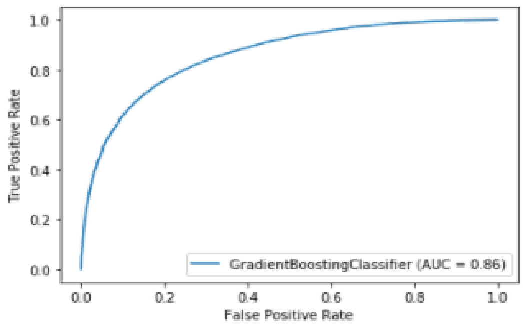
5. Gradient Boosting Classifier



```
[[ 4392  2298]
 [ 7045 43695]]
              precision    recall  f1-score   support

           0       0.38      0.66      0.48      6690
           1       0.95      0.86      0.90     50740

    accuracy                           0.84     57430
   macro avg       0.67      0.76      0.69     57430
weighted avg       0.88      0.84      0.85     57430
```



CONCLUSION

According to the performance metrics, Random Forrest scores highest in accuracy. Also, the curve is tending towards the ideal shape. Hence, Random Forrest looks like the best fit for this data.