FLIP ROBO

Micro Credit Loan Defaulter

Submitted by:

Swanand Barsawade

ACKNOWLEDGMENT

This includes mentioning of all the references, research papers, data sources, professionals and other resources that helped you and guided you in completion of the project.

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.

Analytical Problem Framing

- 1) 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.

- 2) 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.
- 3) Data Visualization
  On visualizing data, there were two important insights I gathered.
  a. Imbalance of data
  b. Distribution was not normal
- 4) Data Normalization
  Since the data was not normal, I normalized all the features except the target variable which was dichotomous(Values '1' and '0').
- 5) Oversampling of Minority class
  Since the data was expensive, I did not want to lose out on data by undersampling the majority class. Instead, I decided to oversample the minority class using SMOTE.

- 6) 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.

Model/s Development and Evaluation

1) Logistic Regression

a) Analysis Stats

b) Analysis Graph

2) Linear SVM

a) Analysis Stats

b) Analysis Graph

3) Decision Tree

a) Analysis Stats

b) Analysis Graph

4) Random Forest

a) Analysis Stats

b) Analysis Graph

5) Gradient Boosting Classifier

a) Analysis Stats

b) Analysis Graph

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.