Micro-Credit Defaulter Model

Submitted By

Anita Tarangi

**INTRODUCTION**

* Business Problem Framing

This is the typical business problem to find out potential defaulters to help lending institutions to reduce credit risks.

* Conceptual Background of the Domain Problem

Machine Learning helps lenders in predicting potential defaulters by their past data or history. Borrowers credit history (income, debt, repayment behavior) can be an important metrics in taking decisions.

* Review of Literature

The dataset had no NULL values but was imbalanced and treated , carried out different methods to train the model.

**Analytical Problem Framing**

* Identification of possible problem-solving approaches Logistic Regression, Decision Tree, Random Forest, Gradient Boost Classifier, K-Nearest Neighbors.
* Data Sources and their formats
* 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.
* For some features, there may be values which might not be realistic. You may have to observe them and treat them with a suitable explanation.
* You might come across outliers in some features which you need to handle as per your understanding. Keep in mind that data is expensive, and we cannot lose more than 7-8% of the data.
* Data Preprocessing Done
* **Checking the size of the dataset**
* **Checking the summary statistics of the dataset**
* **Checking the data types and null values**
* **Dropping columns which aren’t important**
* Hardware and Software Requirements and Tools Used

Python, Pandas, Seaborn, Matplotlib, sklearn, Model Selection, Logistic Regression, Decision Tree, Random Forest, Gradient Boost Classifier, K-Nearest Neighbors.

**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)

Logistic Regression, Decision Tree, Random Forest, Gradient Boost Classifier, K-Nearest Neighbors.

* Run and evaluate selected models

Logistic Regression, Decision Tree, Random Forest, Gradient Boost Classifier, K-Nearest Neighbors.

* Visualizations

Seaborn, Matplotlib

* Interpretation of the Results

Carried out Analysis as well as visualization of the Dataset, treated imbalanced dataand then, trained the model using Logistic Regression, Decision Tree, Random Forest, Gradient Boost Classifier, K-Nearest Neighbors, model predicted with different accuracy for every model, out of all the models, Random Forest has achieved with 87% accuracy rate.

**CONCLUSION**

Data exploration, cleaning, visualization is the basic steps, in which imbalanced data was identified and treated. And then, trained the model using Logistic Regression, Decision Tree, Random Forest, Gradient Boost Classifier, K-Nearest Neighbors, model predicted with different accuracy for every model, out of all the models, Random Forest has achieved with 87% accuracy rate.