
# Indian Institute of Technology Hyderabad

## Fraud Analytics (CS6890)

Assignment : 4 | Cost Sensitive Logistic Regression


| Name | Roll Number |
|---|---|
| Manan Darji | CS22MTECH14004 |
| Dhwani Jakhaniya | CS22MTECH14011 |
| Ankit Sharma | CS22MTECH12003 |
| Vishesh Kothari | CS22MTECH12004 |
| Jayanti Mudliar | CS22MTECH14001 |

# Contents

# 1 | Problem statement

Machine learning often encounters the issue of class imbalance, where the number of instances in one class is notably lower than the other. This results in classification models which will be biased and will perform inadequately for the minority classes. In order to address this problem, we utilize cost-sensitive learning as an approach that considers the cost of misclassification.

Cost-sensitive logistic regression aims to optimize the classification model by incorporating a cost matrix that consists of false negative, false positive, true positive, and true negative costs. The primary objective is to minimize the overall cost of misclassification, which is determined by the total of false negative and false positive costs.

The objective of cost-sensitive logistic regression is hence to create an algorithm that can make precise predictions while considering the cost of misclassification, particularly the false negative and false positive costs. The algorithm's effectiveness is compared to that of baseline algorithms like logistic regression, without any cost consideration. The ultimate aim is to prove that cost-sensitive logistic regression can enhance the performance of classification models on imbalanced datasets.

# 2 | Description of the data set

The dataset provided for cost-sensitive logistic regression, named 'costsensitiveregression.csv', comprises multiple variables, including both independent and dependent variables, along with a false negative cost. These variables are structured in a tabular format, where each row pertains to a specific example, and each column represents a feature. The dataset contains a total of 147,636 rows and 13 columns a part of which is shown in below table

| | NotCount | YesCount | ATPM | PFD | PFG | SFD | SFG | WP | WS | AH | AN | Status | FNC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2 | 21 | 0.0 | 0.000 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.0 | 0.0 | 0.0 | 0 | 0.0 |
| **1** | 23 | 0 | 0.0 | 0.044 | 0.0 | 0.0 | 0.0 | 0.306179 | 0.0 | 0.0 | 0.0 | 1 | 0.0 |
| **2** | 1 | 22 | 0.0 | 0.000 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.0 | 0.0 | 0.0 | 0 | 0.0 |
| **3** | 5 | 18 | 0.0 | 0.000 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.0 | 0.0 | 0.0 | 1 | 0.0 |
| **4** | 1 | 22 | 0.0 | 0.000 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.0 | 0.0 | 0.0 | 0 | 0.0 |

**Figure 2.1:** Data set

The dataset utilized in this study comprises independent variables denoted by columns A through L, which are employed to forecast the dependent variable. However, the exact interpretation of these variables is not mentioned in the problem statement. The dependent variable in the dataset is denoted by column L, which is a binary variable that determines the target class. It can take one of two possible values, 0 or 1, indicating the two classes. Also the dataset have class imbalance where the number of instances in one class is much lower than the other which is shown in Figure 2.2.

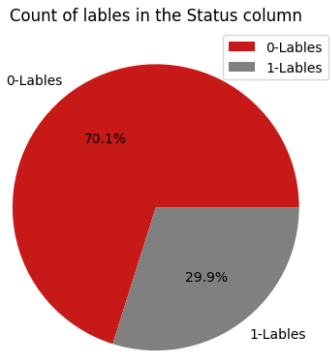


**Figure 2.2:** Class Imbalance

Column M represents the false negative cost, which varies from row to row based on the business details making the dataset suitable for cost-sensitive learning. Cost matrix which provides cost of misclassification is

- True Positive as 4
- False Positive as 4
- True Negative as 0
- False Negative as **variable**.

| | Actual Positive $y_i = 1$ | Actual Negative $y_i = 0$ |
|---|---|---|
| Predicted Positive $c_i = 0$ | $C_{TP_i} = 4$ | $C_{FP_i} = 4$ |
| Predicted Positive $c_i = 0$ | $C_{FN_i} = x$ | $C_{TN_i} = 0$ |

**Figure 2.3:** Cost Matrix

# 3 | Algorithm Used

As we know, Logistic regression is a popular supervised learning algorithm which we use to predict binary outcomes. While in cost-sensitive logistic regression, the model is trained to minimize a cost function that considers the costs associated with making different types of prediction errors.

To calculate the cost function in this implementation, the costs of false positives and true positives are both set to 4, while the cost of false negatives varies for each row in the dataset. The cost of true negatives is set to 0.

According to the material shared by sir the new cost-sensitive logistic regression cost function, by including the different costs into the logistic function as,

$$\begin{aligned} \mathrm{J}_c(\theta) = \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \Big( & y_i(h_\theta(x_i)C_{TP_i} + (1 - h_\theta(x_i))C_{FN_i}) + \\ & (1 - y_i)(h_\theta(x_i)C_{FP_i} + (1 - h_\theta(x_i))C_{TN_i}) \Big) \end{aligned} \tag{3.1}$$

And since this cost function is not convex, we will estimate its parameters using a genetic algorithm.

Genetic algorithms are optimization algorithms which are inspired by the process of natural selection. They use a population-based approach, where a set of candidate solutions, that is chromosomes, is evolved over time through selection, crossover, and mutation operations. In this implementation, the GeneticAlgorithmOptimizer class from the PyEA library is used to optimize the cost function. The optimizer is configured to run for 50 iterations with 50 chromosomes and a mutation rate of 0.25. The result of it can be used to determine the cost-sensitive function of a logistic regression.

Once the training is complete, the logistic regression model can be utilized to make predictions.To evaluate the model's performance, we compare its cost to that of the traditional logistic regression model as per the below equation,

$$savings = \frac{Cost_{LR} - Cost}{Cost_{LR}} \tag{3.2}$$

Where the The $Cost_{LR}$ is obtained from Traditional logistic regression model and $Cost$ is obtained from cost sensitive logistic regression which we have implemented and difference between these two costs is used to calculate a savings score, which is obtained by dividing the cost difference by the cost from the traditional logistic regression model.

# 4 | Results

The given code employs a genetic algorithm to implement cost-sensitive logistic regression, utilizing PyEA for the same. In order to assess the model's efficiency, we compare its cost to that of a traditional model. The difference in costs between the two models is divided by the cost of the traditional logistic regression model, resulting in a saving score. From 4.1 we can see the cost value of the traditional logistic model is 927321.21, whereas 118112.0 for cost-sensitive model, which results in a saving score of 0.87.

| Algorithm | Cost value |
|---|---|
| Traditional logistic regression | 927321.21 |
| Cost-sensitive logistic regression | 118112.0 |
| **Saving cost** | **0.87** |

**Table 4.1:** Cost value

Figure 4.1 represents the illustrative depiction of cost value of traditional versus cost-sensitive model. From this, we can conclude that cost-sensitive logistic regression significantly enhances the performance of logistic regression models, yielding an improvement of ten-fold.

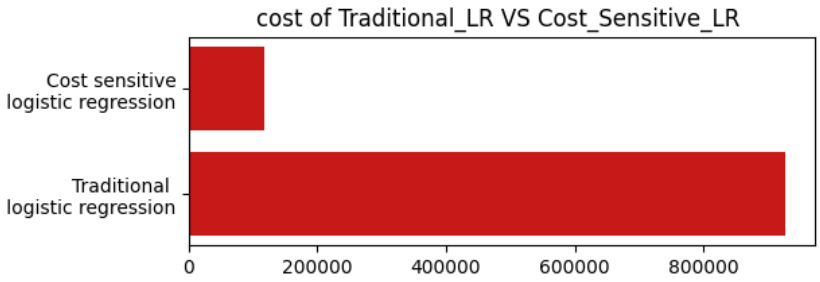


**Figure 4.1:** Traditional Vs Cost Sensitive