Implementation of Cost-Sensitive Logistic Regression

Mayuresh Dindorkar
CS23MTECH14007

Piyush Morey
AI23MTECH14003

Sanyam Kaul
CS23MTECH14011

Shrenik Ganguli
CS23MTECH14014

1. Abstract

Traditional classifiers often overlook error costs, resulting in ineffective class imbalance handling in fraud detection. Cost-Sensitive Logistic Regression integrates cost considerations, prioritizing false negative reduction for enhanced fraud detection and minimized financial losses. This study implements this approach using genetic algorithms and demonstrates its efficacy through empirical evaluation and comparison with traditional logistic regression.

2. Problem Statement

In fraud detection and similar scenarios, traditional classifiers often fail to consider the varying costs of different errors, leading to ineffective handling of class imbalance. False negatives, such as missed fraud cases, are particularly damaging, outweighing false positives in terms of financial losses and trust erosion.

To address this issue, there's a need for ”Cost-Sensitive Logistic Regression.” This approach integrates cost considerations directly into the model, assigning specific costs to different types of errors. By prioritizing the reduction of false negatives, the goal is to improve fraud detection accuracy.

The research objective is to develop and implement a cost-sensitive logistic regression model using techniques like genetic algorithms for parameter optimization.

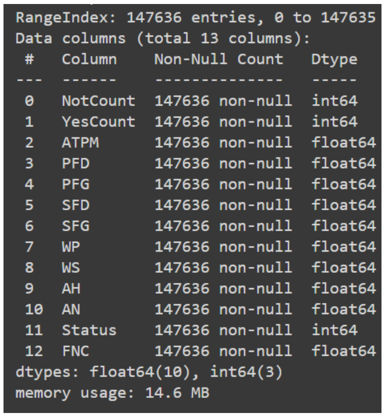
3. Description of the dataset

Within the file 'costsensitiveregression.csv', columns A to K represent independent variables, while column L serves as the dependent variable. Additionally, each entry includes a false negative cost in column M, which varies from row to row based on risk parameter details. Notably, true positive and false positive costs are constant for all entries, set at 6, while the true negative cost remains consistent at 0. Upon examination, a notable class imbalance is evident, with 44,082 instances of fraud juxtaposed with 103,554 non-fraudulent cases.

The significant class imbalance underscores the necessity for specialized techniques, such as cost-sensitive logistic regression, to effectively handle the dataset's skewed distribution and optimize model performance.

To address this, we have opted for an 80:20 split for training and testing purposes.

In summary, the dataset's class imbalance underscores the necessity for advanced modeling techniques, such as cost-sensitive logistic regression, to effectively address the skewed distribution of data and optimize the performance of fraud detection systems.



```
RangeIndex: 147636 entries, 0 to 147635
Data columns (total 13 columns):
 #   Column     Non-Null Count   Dtype
---  ------     --------------   -----
 0   NotCount   147636 non-null  int64
 1   YesCount   147636 non-null  int64
 2   ATPM       147636 non-null  float64
 3   PFD        147636 non-null  float64
 4   PFG        147636 non-null  float64
 5   SFD        147636 non-null  float64
 6   SFG        147636 non-null  float64
 7   WP         147636 non-null  float64
 8   WS         147636 non-null  float64
 9   AH         147636 non-null  float64
 10  AN         147636 non-null  float64
 11  Status     147636 non-null  int64
 12  FNC        147636 non-null  float64
dtypes: float64(10), int64(3)
memory usage: 14.6 MB
```

Figure 1. Dataset

4. Algorithm Used

4.1. Cost Sensitive Loss function

$$\begin{aligned} J^c(\theta) = \frac{1}{N}\sum_{i}^{N}(y_i(h_\theta(X_i)C_{TP_i} + (1-h_\theta(X_i))C_{FP_i}) \\ + (1-y_i)(h_\theta(x_i))C_{FP_i} + (1-h_\theta(x_i))C_{TN_i}) \end{aligned} \quad (1)$$

$$c_i = \int_0^1 a_i(y_i(-\log f(g(x_i,\beta)))) \quad (2)$$

Loss functions of Bahnsen's and Guhnmann's approach

4.2. Approaches

Here we have used **2 approaches** to implement the model:

1) Bahnsen's Approach based Cost-sensitive logistic regression with loss function in equation - Eq 1

2) Guhnmann's Approach based Cost-sensitive logistic regression with loss function in equation - Eq - 2

We have produced results for both approaches and compared them with the inbuilt Logistic Regressor of ScikitLearn in terms of savings score.

Logistic regression function

$$z = w^T * x + b$$
$$F(z) = \frac{1}{(1 + e^{-z})} \tag{3}$$

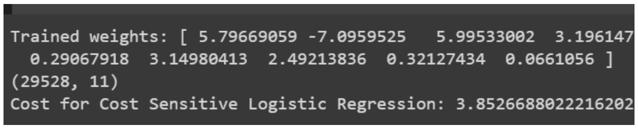
Genetic Algorithm

Algorithm 1 Genetic Algorithm

```
procedure GeneticAlgorithm($pop\_size, num\_generations,$) $num\_parents\_mating, fitness\_func$
    Initialize population with random solutions
    for $generation \leftarrow 1$ to $num\_generations$ do
        Calculate fitness for each solution in population
        Apply mutation to offspring with probability
        Replace old population with new population
        Sort population by fitness
        Print best fitness value for current generation
    end for
    return best solution from final population
end procedure
```

5. Results

5.1. Results of Bahnsen's Approach:

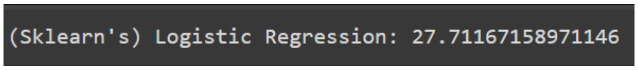


```
Trained weights: [ 5.79669059 -7.0959525   5.99533002  3.196147
  0.29067918  3.14980413  2.49213836  0.32127434  0.0661056 ]
(29528, 11)
Cost for Cost Sensitive Logistic Regression: 3.8526688022216202
```

Figure 2. Cost of Bahnsen's Cost-Sensitive Logistic Regression

The cost for Bansen's Cost-Sensitive Logistic Regression is 3.8526688022216202.

5.2. Results of Simple Sklearn Logistic Regression



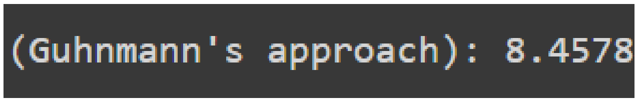
```
(Sklearn's) Logistic Regression: 27.71167158971146
```

Figure 3. Simple Sklearn Logistic Regression

The figure 3 is the simple Sklearn Logistic Regression cost. We can observe that Sk learn LR is much higher than Bahnsen's Approach.

5.3. Cost of Guhnmann's Cost-Sensitive Logistic Regression (Variant A)

We can see in figure 6 that the Cost for Cost-Sensitive Logistic Regression (Guhnmann's approach) is 8.457842755576396.



```
(Guhnmann's approach): 8.4578
```

Figure 4. Cost for Cost-Sensitive Logistic Regression (Guhnmann's approach)

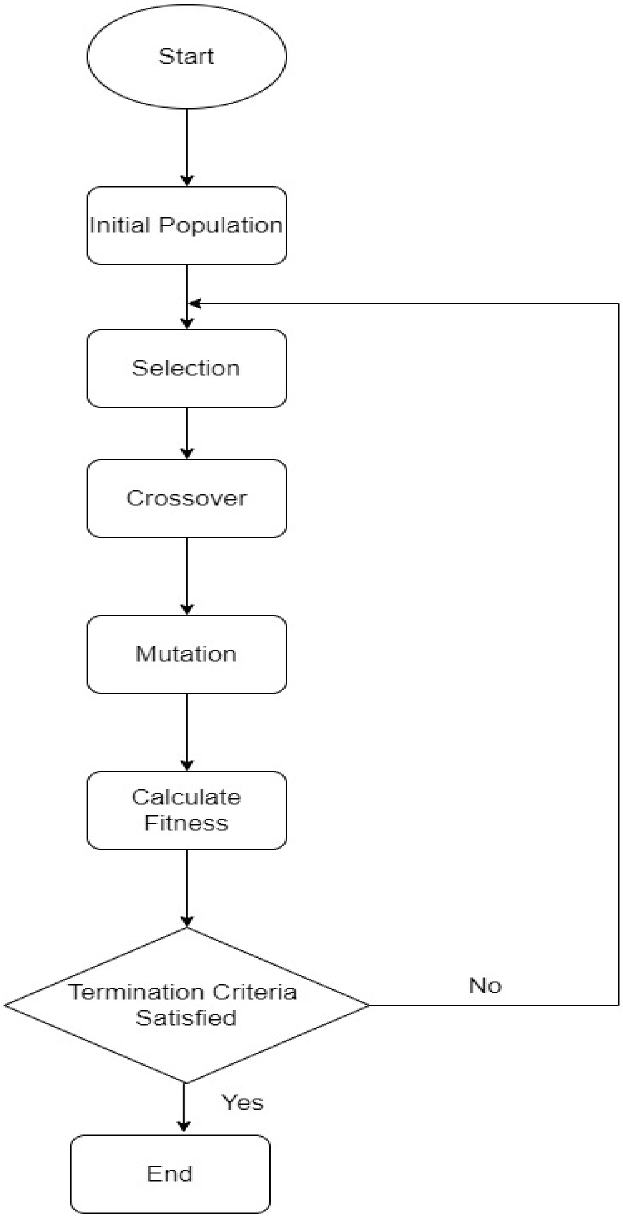


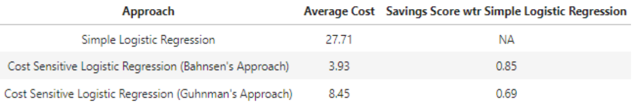
Figure 5. Genetic algorithm

5.4. Comparison based on Savings score

Savings score quantifies the cost reduction achieved by employing a cost-sensitive algorithm compared to a standard algorithm.

$$\textit{Savings score} = \frac{SLR - CSLR}{SLR}$$

Average cost and Savings Score with respect to Simple Logistic Regression are mentioned in Figure 6. We can see both Bahnsen's and Guhnmann's approaches give good results for the problem of cost-sensitive logistic regression.



Approach	Average Cost	Savings Score wtr Simple Logistic Regression
Simple Logistic Regression	27.71	NA
Cost Sensitive Logistic Regression (Bahnsen's Approach)	3.93	0.85
Cost Sensitive Logistic Regression (Guhnman's Approach)	8.45	0.69

Figure 6. Results

6. Conclusion

In conclusion, our study implemented and compared two cost-sensitive logistic regression approaches, namely Bahnsen's and Guhnmann's, against a standard logistic regression model. Both Bahnsen's and Guhnmann's approaches demonstrated superior performance in reducing the average cost of misclassifications compared to the simple logistic regression model. Additionally, the savings score metric highlighted the significant cost reduction achieved by employing cost-sensitive algorithms, with Bahnsen's approach achieving a savings score of 0.85 and Guhnmann's approach achieving a score of 0.69. These results underscore the effectiveness of cost-sensitive logistic regression in improving fraud detection accuracy while minimizing financial losses.