Heart Disease Classification using Logistic Regression

Parth Agrawal
Department of Information Technology, IIIT Allahabad

Abstract

This report presents the design and implementation of a Logistic Regression-based binary classifier for heart disease prediction using the Cleveland Heart Disease dataset. The model was implemented from scratch using only NumPy and trained with both L1 (Lasso) and L2 (Ridge) regularization to improve generalization and prevent overfitting. Evaluation metrics such as accuracy, precision, recall, F1-score, confusion matrix, and ROC curves are presented to compare the performance of both regularized models.

1 Introduction

Cardiovascular diseases are among the leading causes of death worldwide. Machine learning models, particularly Logistic Regression, are effective in predicting the likelihood of heart disease based on patient health indicators. Logistic Regression is a linear model that estimates the probability of a binary outcome (disease or no disease) using the sigmoid function.

Objective

The goal of this experiment is to:

1. Implement Logistic Regression from scratch using NumPy.
2. Apply both L1 and L2 regularization techniques.
3. Evaluate model performance using confusion matrix, accuracy, precision, recall, F1-score, and ROC curves.

2 Theory

2.1 Logistic Regression

Logistic Regression is a probabilistic linear model used for classification. Given a set of features $\mathbf{x} = [x_1, x_2, ..., x_n]$, the model computes a linear combination:

$$z = \mathbf{w}^T\mathbf{x} + b$$

The probability that the sample belongs to class 1 is modeled using the sigmoid function:

$$P(y = 1|\mathbf{x}) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

The decision boundary is defined by the threshold $P(y = 1|\mathbf{x}) = 0.5$.

2.2 Loss Function (Binary Cross-Entropy)

To optimize model parameters, the following loss function is minimized:

$$J(\mathbf{w}, b) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

2.3 Gradient Descent Optimization

Weights are updated iteratively to minimize the loss function:

$$w_j := w_j - \alpha \frac{\partial J}{\partial w_j}$$

$$b := b - \alpha \frac{\partial J}{\partial b}$$

where α is the learning rate.

2.4 Regularization

Regularization helps reduce overfitting by penalizing large weights.

- **L1 Regularization (Lasso)** adds an absolute penalty:

$$J_{L1} = J + \lambda \sum_{j=1}^{n} |w_j|$$

  L1 tends to drive some weights to zero, leading to feature selection.

- **L2 Regularization (Ridge)** adds a squared penalty:

$$J_{L2} = J + \lambda \sum_{j=1}^{n} w_j^2$$

  L2 discourages large weights, improving model stability and reducing variance.

2.5 Relation to Gaussian Discriminant Analysis (GDA)

Logistic regression and Gaussian Discriminant Analysis (GDA) are related probabilistic models. GDA assumes that class-conditional densities follow Gaussian distributions, and the posterior probability is derived via Bayes' theorem. Logistic regression can be viewed as a discriminative counterpart to GDA—it directly models $P(y|\mathbf{x})$ without assuming a distribution for $\mathbf{x}$.

3 Dataset Description

The dataset used is the **Cleveland Heart Disease dataset**, containing 303 records and 14 features such as age, sex, chest pain type, resting blood pressure, cholesterol, fasting blood sugar, maximum heart rate, and others. The final column indicates the diagnosis of heart disease:

$$0 = \text{No Disease}, \quad 1 = \text{Heart Disease}$$

4 Implementation Details

- Programming Language: Python (NumPy only)
- Data Split: 70% training, 30% testing
- Learning Rate: 0.01
- Epochs: 2500
- Regularization Strength (λ): 0.1

5 Training Logs

The following loss values were observed during training:

L1-Regularized Model

```
Epoch 0 | Loss (l1): 0.693152
Epoch 500 | Loss (l1): 0.366205
Epoch 1000 | Loss (l1): 0.347277
Epoch 1500 | Loss (l1): 0.341695
Epoch 2000 | Loss (l1): 0.339195
Epoch 2500 | Loss (l1): 0.337850
```

L2-Regularized Model

```
Epoch 0 | Loss (l2): 0.693147
Epoch 500 | Loss (l2): 0.365487
Epoch 1000 | Loss (l2): 0.346543
Epoch 1500 | Loss (l2): 0.340932
Epoch 2000 | Loss (l2): 0.338427
Epoch 2500 | Loss (l2): 0.337089
```

6 Results and Evaluation

Confusion Matrices

	Predicted 0	Predicted 1
Actual 0	33	4
Actual 1	11	43

Performance Metrics

Model	Accuracy	Precision	Recall	F1-score
L1 Regularization	0.8352	0.8919	0.7500	0.8148
L2 Regularization	0.8352	0.8919	0.7500	0.8148

ROC Curves

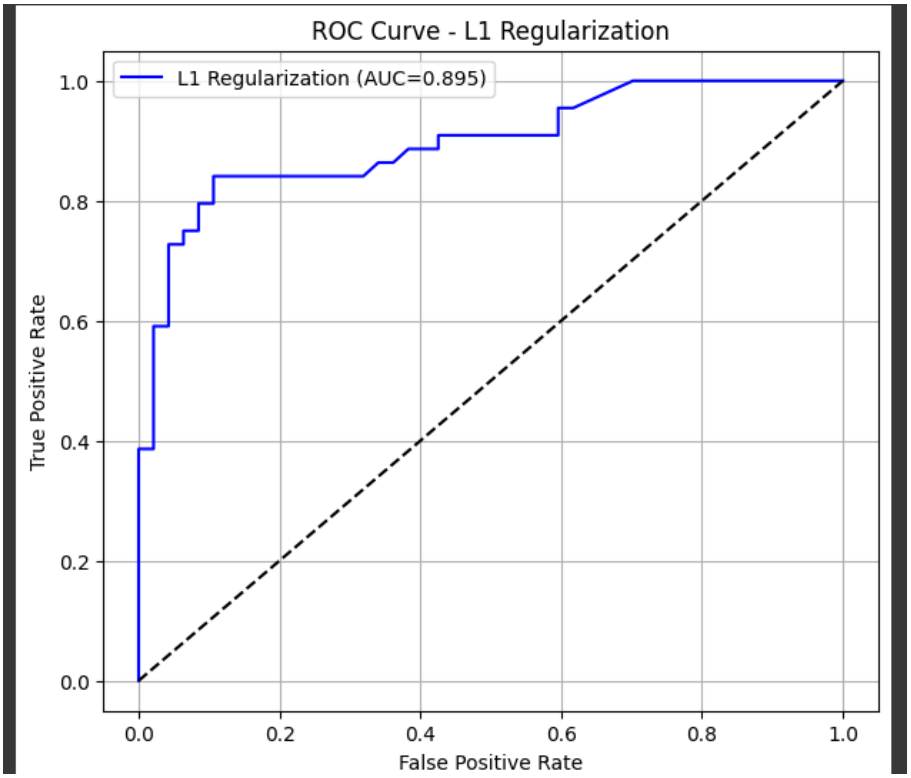

Figure 1: ROC Curve for Logistic Regression with L1 Regularization

7 Discussion

Both regularization techniques produced similar performance on this dataset. The accuracy of 83.5% indicates that the model successfully learned to separate patients with and without heart disease. L1 regularization introduced sparsity, meaning that some less informative features had zero or near-zero weights. L2 regularization provided smoother weight distributions and more stable convergence. The ROC curves for both models

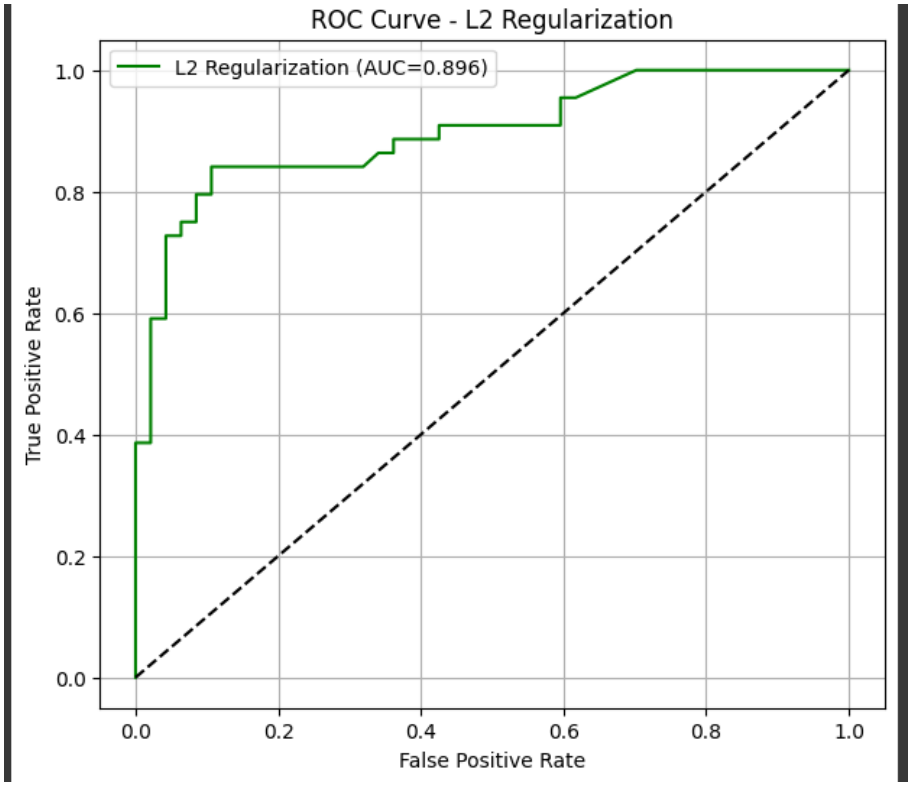

Figure 2: ROC Curve for Logistic Regression with L2 Regularization

showed good separability between the two classes, confirming reliable predictive capability.

8 Conclusion

The experiment demonstrated that logistic regression with L1 and L2 regularization is effective for heart disease classification. Both models achieved comparable accuracy, precision, recall, and F1 scores. Regularization proved essential for improving generalization and avoiding overfitting. Future work could include hyperparameter tuning, feature scaling comparisons, or comparison with nonlinear classifiers such as Support Vector Machines or Neural Networks.