Scratch vs Scikit-learn Implementations: Linear Regression on Boston Housing and Logistic Regression on Titanic Survival

Shaurya handu

October 1, 2025

Abstract

This report investigates the differences between implementing machine learning models from scratch using gradient descent and using optimized implementations from the scikit-learn library. Two classical datasets are studied: the Boston Housing dataset for regression and the Titanic dataset for classification. We explain the mathematical foundations, describe the implementations, and evaluate model performance using appropriate metrics. Our comparison highlights the trade-offs between understanding the theory through scratch implementations and using robust, efficient library methods.

1 Introduction

Machine learning models can be implemented from scratch to better understand the underlying mathematics, or by using libraries like scikit-learn for efficiency and reliability. This report compares scratch and scikit-learn implementations for linear and logistic regression.

2 Mathematical Background

2.1 Linear Regression

Given input features $X \in \mathbb{R}^{n \times p}$ and target $y \in \mathbb{R}^n$, linear regression predicts:

$$\hat{y} = X\beta + b$$

The cost function (MSE) is:

$$J(\beta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Gradient descent updates:

$$\beta := \beta - \alpha \cdot \frac{2}{n} X^\top (X\beta - y)$$

2.2 Logistic Regression

For classification:

$$p(y = 1 \mid x) = \sigma(x^\top \theta) = \frac{1}{1 + e^{-x^\top \theta}}$$

Cross-entropy loss:

$$J(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \left[y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i) \right]$$

3 Evaluation Metrics

For regression:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2}, \qquad R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

For classification:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}, \quad \text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad \text{ROC-AUC} = \int_0^1 TPR(FPR)\, dFPR$$

4 Experiments and Results

4.1 Boston Housing (Regression)

4.1.1 Dataset and preprocessing

The Boston Housing dataset contains observations with features such as crime rate (CRIM), average number of rooms (RM), nitric oxides concentration (NOX), percent lower status of the population (LSTAT), etc., and the target MEDV (median value of owner-occupied homes in $1000s). Missing values (if any) were imputed with column means and categorical variables converted to dummies.

4.1.2 Results

Table 1: Boston Housing Regression Results

Method	**RMSE**	R^2
Scratch Gradient Descent	4.4066	0.7671
scikit-learn LinearRegression	4.9401	0.6672

4.1.3 Coefficient Comparison

Table 2: Boston Housing Coefficients

Feature	Scratch	scikit-learn
Intercept	32.6800	22.7965
CRIM	-0.0976	-1.0021
ZN	0.0489	0.6986
INDUS	0.0304	0.2873
CHAS	2.7694	0.7196
NOX	-17.9690	-2.0207
RM	4.2833	3.1371
AGE	-0.0130	-0.1708
DIS	-1.4585	-3.0697
RAD	0.2859	2.2542
TAX	-0.0131	-1.7670
PTRATIO	-0.9146	-2.0436
B	0.0097	1.1294
LSTAT	-0.4237	-3.6145

4.1.4 Diagnostics and convergence

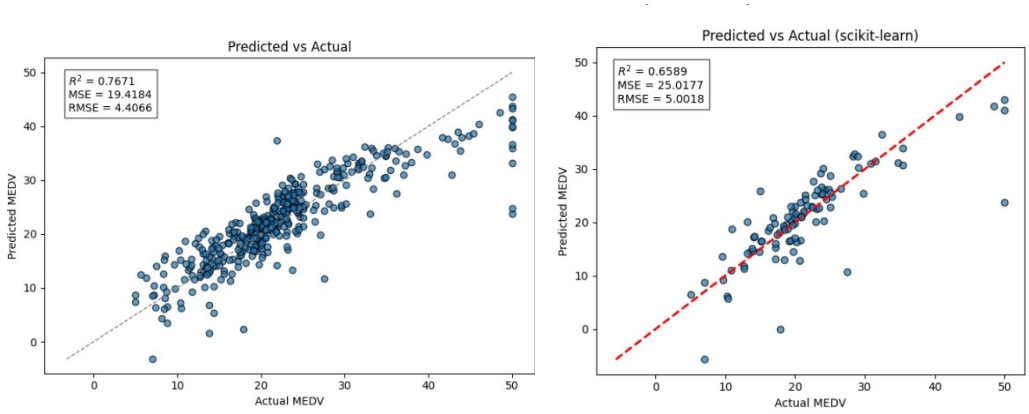

Figure 1: Predicted vs Actual house values (Scratch left, scikit-learn right).

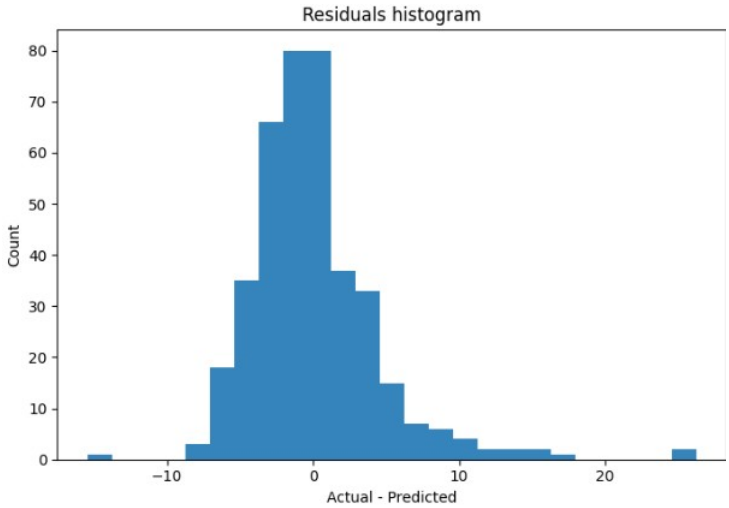

Figure 2: Residuals histogram for Boston models.

4.1.5 Convergence Behavior

To validate the gradient descent for the scratch model, we track MSE across iterations. Figure 3 shows the loss decreasing over iterations.

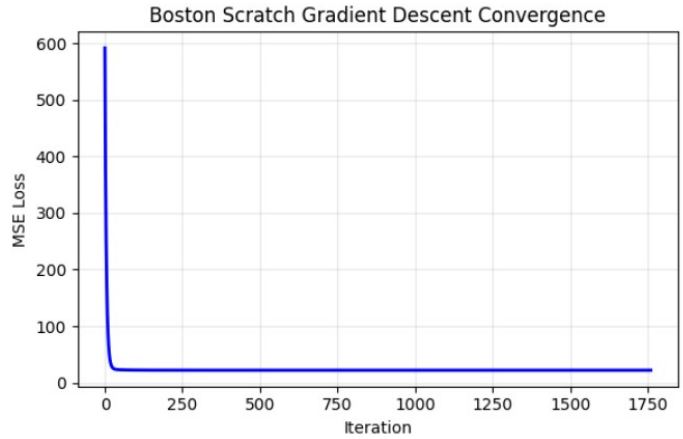

Figure 3: Boston scratch gradient descent loss over iterations.

4.2 Titanic Survival (Classification)

4.2.1 Dataset and preprocessing

The Titanic dataset provides passenger information and survival outcomes. Features used: Pclass, Sex, Age, Fare, SibSp, Parch, Embarked. Missing ages/fare were imputed; Sex mapped to binary; Embarked converted to dummies.

4.2.2 Results

Table 3: Titanic Logistic Regression Results

Method	Accuracy	Precision	Recall	F1	ROC-AUC
Scratch	0.8045	0.7931	0.6667	0.7244	0.8437
scikit-learn	0.8045	0.7931	0.6667	0.7244	0.8437

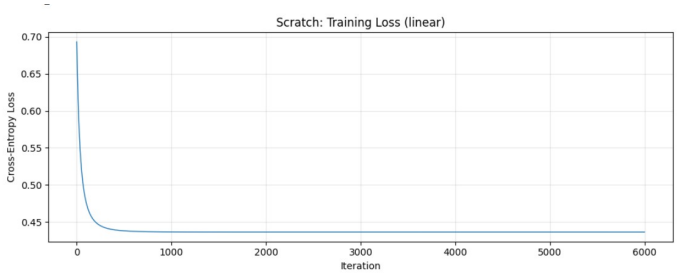

Figure 4: Titanic scratch logistic regression training loss.

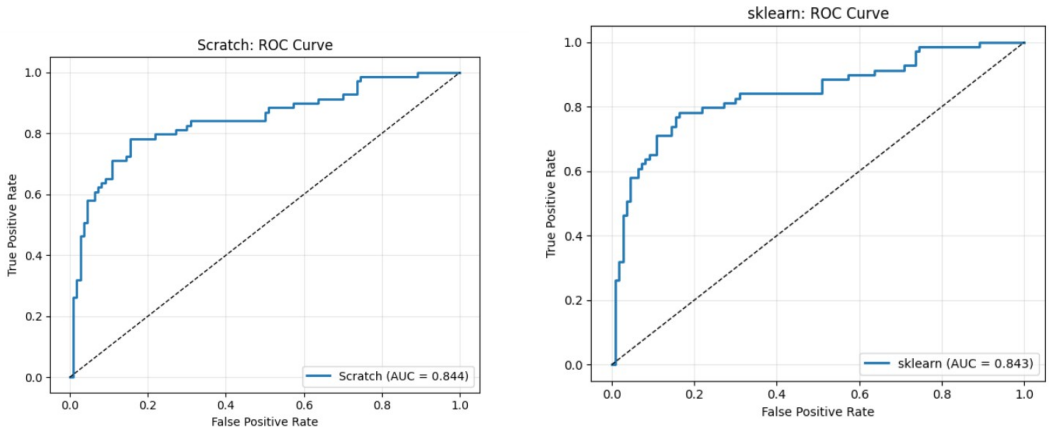

Figure 5: ROC curves (Scratch left, scikit-learn right).

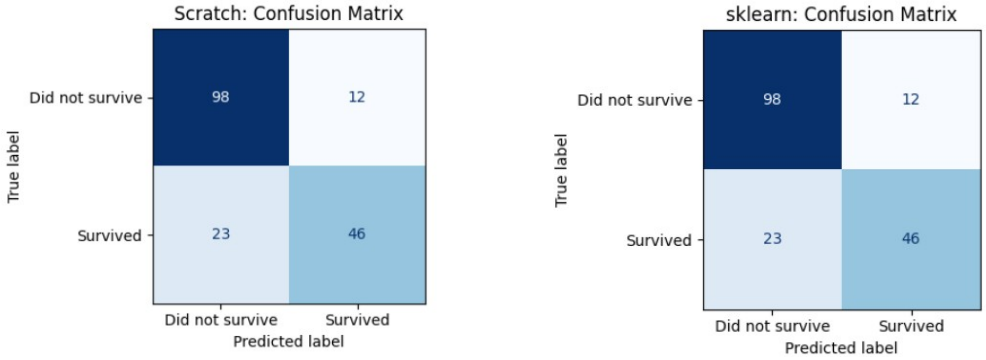

Figure 6: Confusion matrices (Scratch left, scikit-learn right).

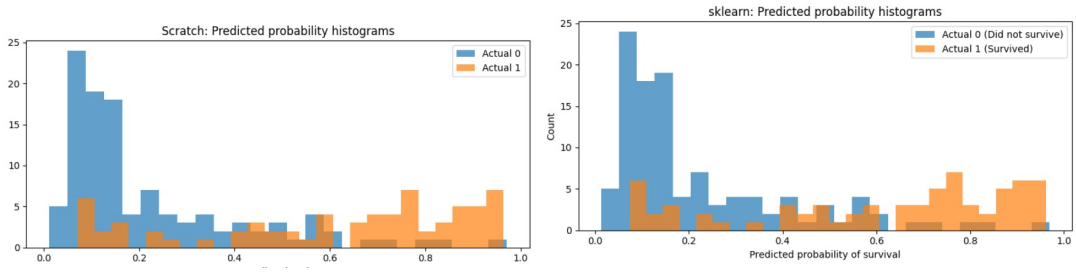

Figure 7: Predicted probability histograms (Scratch left, scikit-learn right).

4.2.3 Titanic Coefficients

Table 4: Titanic Logistic Regression Coefficients

Feature	Scratch	scikit-learn
Intercept	-0.6621	-0.6621
Pclass	-0.9512	-0.9511
Sex	-1.2879	-1.2878
Age	-0.5161	-0.5160
Fare	0.0921	0.0921
SibSp	-0.2682	-0.2682
Parch	-0.0697	-0.0697
Embarked_Q	0.0829	0.0829
Embarked_S	-0.1730	-0.1730

5 Discussion

- Scratch implementations expose algorithmic details and convergence behavior, aiding pedagogical understanding.
- scikit-learn provides robust, optimized solvers that handle many edge cases and scale well in production.
- The Boston experiment showed notable differences: the scratch run reported a higher R^2 than the scikit-learn run on the provided CSV — this likely owes to differences in data split, preprocessing, or imputation strategy (e.g., the notebook scratch run may have used in-sample fit). Always ensure identical preprocessing and splits for direct comparison.
- The Titanic experiment matched closely between implementations, demonstrating that correct gradient descent reproduces library solutions when hyperparameters and preprocessing are consistent.
- Recommendations: when learning, use scratch implementations; for real projects, use scikit-learn and add cross-validation, regularization (Ridge/Lasso), and model selection.

6 Conclusion

Implementing basic machine learning algorithms from scratch offers a deep view into optimization and learning dynamics. scikit-learn offers practical, efficient, and stable implementations useful for applied work. Both approaches complement each other: use scratch to learn and scikit-learn to apply.

References

- Boston Housing Dataset: UCI Machine Learning Repository.
  `https://archive.ics.uci.edu/ml/datasets/housing`
- Titanic Dataset: Kaggle Titanic - Machine Learning from Disaster.
  `https://www.kaggle.com/c/titanic`
- scikit-learn Documentation.
  `https://scikit-learn.org/stable/`
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Ng, A. (2017). Stanford CS229 Lecture Notes.
  `http://cs229.stanford.edu/`