Linear Regression from Scratch on Boston Housing Dataset

Vaibhav Chourasia

October 2, 2025

Abstract

This report presents a linear regression model implemented from scratch using gradient descent to predict housing prices on the Boston Housing dataset. The model is evaluated using Mean Squared Error (MSE) and the coefficient of determination (R^2). Visualizations include Actual vs Predicted plots and Residual plots to assess model performance.

1 Introduction

Linear regression is a fundamental supervised machine learning algorithm. It models the relationship between a dependent variable (y) and one or more independent variables (X) by fitting a linear equation:

$$\hat{y} = w^T X + b \tag{1}$$

where w is the vector of feature weights, b is the bias, and $\hat{y}$ is the predicted value. The Boston Housing dataset contains 506 instances with 13 features related to housing conditions and prices.

2 Dataset Description

The dataset consists of 13 features:

- CRIM - per capita crime rate by town
- ZN - proportion of residential land zoned for lots over 25,000 sq.ft.
- INDUS - proportion of non-retail business acres per town
- CHAS - Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- NOX - nitric oxides concentration (parts per 10 million)
- RM - average number of rooms per dwelling
- AGE - proportion of owner-occupied units built prior to 1940
- DIS - weighted distances to five Boston employment centres

- RAD - index of accessibility to radial highways
- TAX - full-value property-tax rate per $10,000
- PTRATIO - pupil-teacher ratio by town
- B - proportion of blacks by town
- LSTAT - % lower status of the population
- MEDV - Median value of owner-occupied homes (target)

3 Methodology

3.1 Feature Scaling

Gradient descent converges faster if features are normalized. Each feature x_j is scaled as:

$$x'_j = \frac{x_j - \mu_j}{\sigma_j} \tag{2}$$

where μ_j and σ_j are the mean and standard deviation of feature j.

3.2 Gradient Descent Derivation

The model minimizes the Mean Squared Error (MSE) loss function:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

where $\hat{y}_i = \sum_{j=1}^{m} w_j x_{ij} + b$.

To minimize MSE, we compute partial derivatives with respect to each weight w_j and the bias b:

$$\frac{\partial \text{MSE}}{\partial w_j} = \frac{\partial}{\partial w_j}\left[\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right] \tag{4}$$

$$= \frac{1}{n}\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\frac{\partial \hat{y}_i}{\partial w_j} \tag{5}$$

$$= \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)x_{ij} \quad \text{(factor 2 omitted as in code)} \tag{6}$$

For the bias term:

$$\frac{\partial \text{MSE}}{\partial b} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i) \tag{7}$$

Using gradient descent, the weights and bias are updated iteratively as:

$$w_j := w_j - \alpha \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i) x_{ij} \tag{8}$$

$$b := b - \alpha \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i) \tag{9}$$

where α is the learning rate and n is the number of samples.

3.3 Residuals

Residuals are defined as:

$$r_i = y_i - \hat{y}_i \tag{10}$$

These are used to create the residual plot to visualize error distribution.

3.4 Mean Squared Error and R^2

The model performance is evaluated using:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{11}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{12}$$

4 Results

The model was trained with a learning rate $\alpha = 0.01$ for 10,000 epochs.

4.1 Final Weights and Bias

```
Final Weights: [-0.92787566  1.08109247  0.13942259  0.6819511  -2.05646143  2.674498
  0.01923921 -3.10415187  2.65849003 -2.07254369 -2.06046895  0.84924216
 -3.74348336]
Final Bias: 22.532806324110496
```

4.2 Regression Equation

The regression equation obtained from the scratch implementation (using normalized features) is:

$$\begin{aligned} \text{Price} = &(-0.9279 \cdot \text{CRIM}) + (1.0811 \cdot \text{ZN}) + (0.1394 \cdot \text{INDUS}) + (0.6820 \cdot \text{CHAS}) \\ &+ (-2.0565 \cdot \text{NOX}) + (2.6745 \cdot \text{RM}) + (0.0192 \cdot \text{AGE}) + (-3.1042 \cdot \text{DIS}) \\ &+ (2.6585 \cdot \text{RAD}) + (-2.0725 \cdot \text{TAX}) + (-2.0605 \cdot \text{PTRATIO}) + (0.8492 \cdot \text{B}) \\ &+ (-3.7435 \cdot \text{LSTAT}) + 22.5328 \end{aligned} \tag{13}$$

4.3 Evaluation Metrics

- Mean Squared Error (MSE): 21.8948
- R^2 Score: 0.7406

4.4 Plots

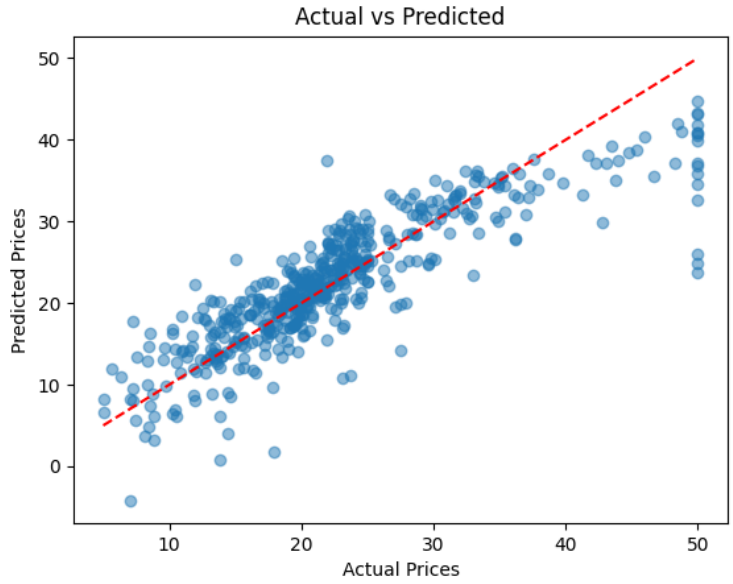

Figure 1: Actual vs Predicted Prices

Figure 2: Residual Plot

5 Discussion

The Actual vs Predicted plot shows that the model captures the general trend of housing prices, although some points deviate from the ideal fit. The Residual plot indicates that residuals are roughly randomly distributed around zero, confirming the linear model is appropriate.

6 Conclusion

A linear regression model was implemented from scratch using gradient descent. Despite its simplicity, it provides reasonable predictions for the Boston Housing dataset. Feature scaling and sufficient epochs ensured convergence.