CSOC Project Report

Pragya Rathal

May 2025

1 Introduction

This report explores three different implementations of linear regression: two developed from scratch and one using the `scikit-learn` library. The models are compared in terms of accuracy, convergence time, and implementation efficiency.

2 Performance Metrics

All performance metrics, including R^2 score and accuracy, are presented in the accompanying code.

- Model 1 (loop-based): 10 minutes to train
- Model 2 (vectorized): 0.64 seconds
- Model 3 (scikit-learn): 0.18 seconds
- Model 1 (loop-based): mae = 49914
- Model 2 (vectorized): mae = 49914
- Model 3 (scikit-learn): mae = 49911

Despite differences in implementation, all models achieved similar R^2 scores up to two decimal places. However, the scikit-learn model had the best overall accuracy.

3 Analysis

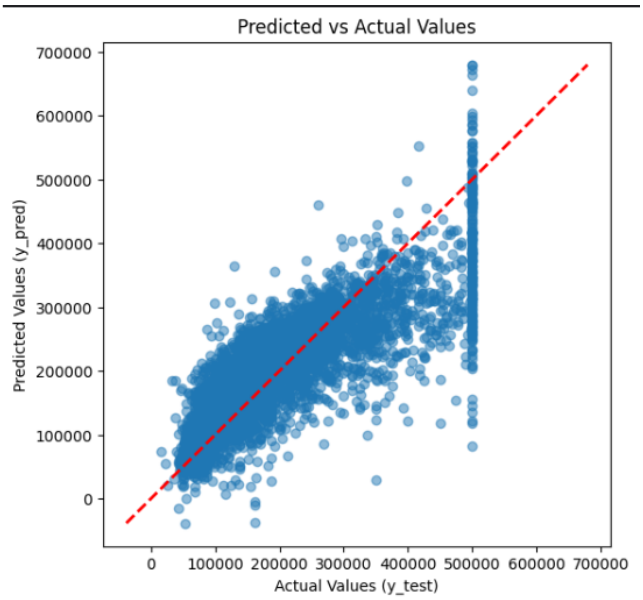
The scikit-learn model achieved the highest R^2 score and had the shortest training time due to its use of Ordinary Least Squares, which computes optimal coefficients analytically using matrix algebra.

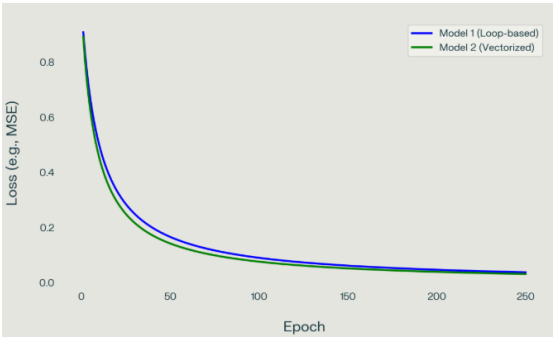
Models 1 and 2 used gradient descent, which iteratively minimizes the loss. While effective, gradient descent is slower and dependent on hyperparameters like learning rate and number of epochs.

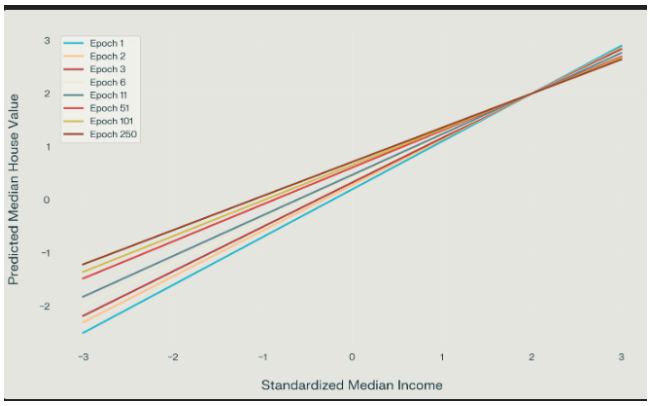
For large datasets with many features, OLS can become computationally expensive due to the inversion of large matrices. In such cases, gradient descent is often preferred because it scales better and avoids matrix operations.

The behavior of gradient descent is highly influenced by the choice of initial parameters and learning rate. If the learning rate is too low, convergence is slow; if it is too high, the algorithm may diverge or overshoot the minimum. Additionally, in non-convex loss surfaces (common in deep learning), gradient descent can get stuck in local minima, though this is less of a concern in convex problems like linear regression.

4 Visualization







5 References

- CampusX – Used as a primary learning resource for model implementation and ML concepts.
- ChatGPT – Used for proofreading this document and improving clarity and structure in technical descriptions.
- ChatGPT – Assisted with resolving syntax issues in Model 1, particularly in the line: `errors = [y_train.iloc[o] - y_hat[o] for o in range(len(y_train))]`, and in similar instances throughout the code.
- Greg Hogg – one-hot encoding and switching positions of columns in a dataset
- geeksforgeeks and chatgpt – for fit function in model-1
- graphs code – perplexity AI