Comparative Evaluation of Regression Techniques

Adityaa Mehra

May 18, 2025

1 Convergence Behavior

Method	**Time Taken (s)**
Manual GD	26.84343910217285
Vectorized GD	8.688417911529541
Scikit-learn	0.018739938735961914

Table 1: Time taken for each model to converge.

2 Model Performance

Method	**MAE**	**RMSE**	$\mathbf{R^2}$
Manual GD	0.272	0.363	0.575
Vectorized GD	0.272	0.363	0.575
Scikit-learn	0.242	0.333	0.641

Table 2: Performance metrics for each regression method.

3 Visualizations

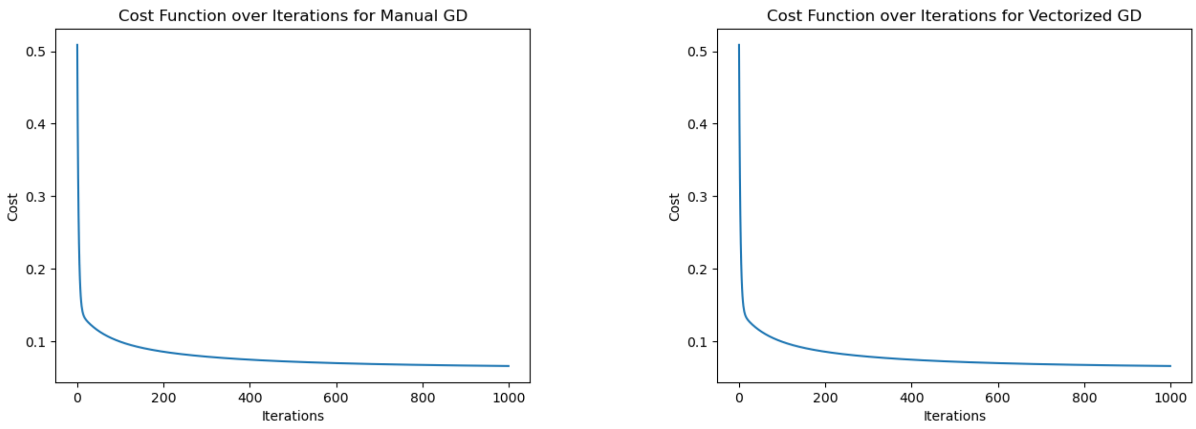

Figure 1: Cost vs. Iterations for Manual GD and Vectorized GD.

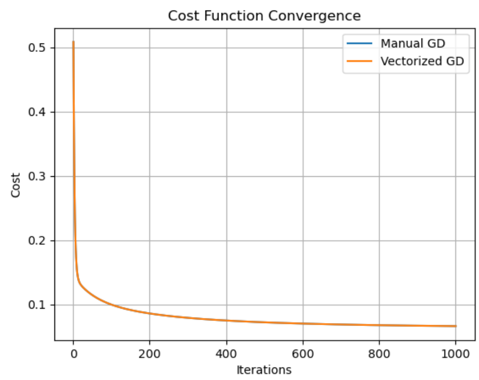

Figure 2: Cost function convergence over iterations for Manual and Vectorized Gradient Descent

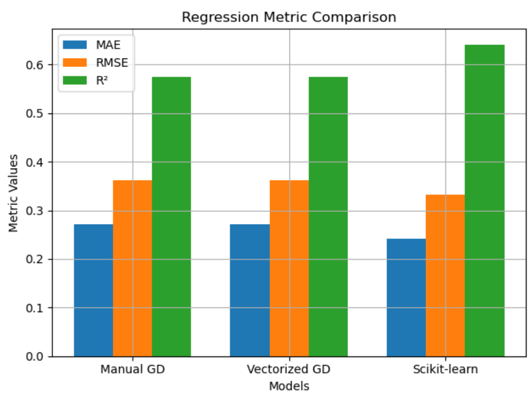

Figure 3: Comparison of MAE, RMSE, and R^2 across all models

4 Analysis and Discussion

The vectorized gradient descent not only reduced convergence time significantly but also produced comparable results to the manual implementation. The key difference lies in computational efficiency — NumPy's optimized matrix operations allow much faster execution.

Scikit-learn's implementation performed best in terms of convergence speed and final error values. This is attributed to its use of optimized solvers (like the normal equation or SVD) and internal data handling.

The learning rate and initial parameters had a noticeable effect on convergence in the manual models. A poorly chosen learning rate led to slower convergence or divergence.

Scalability is also a critical factor: while manual implementations can work for small datasets, they are inefficient for large-scale data. Scikit-learn, being built on top of efficient linear algebra backends, scales well to larger inputs.

5 Conclusion

This comparative study highlights that:

- Vectorization significantly boosts training efficiency.
- Manual gradient descent, though educational, is impractical for real-world tasks.
- Scikit-learn provides both speed and accuracy, making it ideal for production environments.

For future work, experimenting with stochastic or mini-batch gradient descent and regularization techniques (L1/L2) can provide deeper insights into optimization strategies.