Mathematical Foundation for AI

Project

Impact of Linear Regression, SVD, and Gradient Descent on Disease Prediction

Group Members

Aaish Faisal Hameedi (25K-7608)

Osama Bin Javed (25K-7604)

Mehdi Abbas (25K-7601)

Syed Muhammad Sheeraz Nadeem (25K-7624)

Submission Date

23rd November 2025

Abstract

This project implements and compares three core mathematical approaches to linear regression: Ordinary Least Squares (OLS), Singular Value Decomposition (SVD), and Gradient Descent. Using the Scikit-Learn **Diabetes dataset**, we modeled disease progression based on ten physiological features. The data was preprocessed via standardization to ensure numerical stability. We analyzed the performance of each method in terms of Mean Squared Error (MSE) and computational behavior. Furthermore, we applied Principal Component Analysis (PCA) via SVD to demonstrate dimensionality reduction.

Our results indicate that OLS and SVD provide identical analytical solutions with a Test MSE of approximately **2900.19**. However, SVD offers superior stability for ill-conditioned matrices. Batch Gradient Descent achieved a comparable Test MSE of **2895.30** with a learning rate of 0.1. Notably, PCA analysis revealed that using only the top **4** principal components improved generalization, lowering the Test MSE to **2874.18** by reducing overfitting.

1. Introduction

We utilized the **Diabetes Dataset** provided by Scikit-Learn. The dataset consists of **442 samples** and **10 numerical features** (age, sex, bmi, bp, s1, s2, s3, s4, s5, s6). The objective is to predict a quantitative measure of disease progression one year after baseline.

The regression problem is formulated as

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

where we aim to find the coefficient vector

$$\beta$$

that minimizes the sum of squared residuals:

$$L(\beta) = || y - X\beta ||^2$$

2. Methodology

2.1 Ordinary Least Squares (OLS)

We implemented the normal equation

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

While this provides an exact solution, it requires computing the inverse of

$$X^T X$$

If features are highly correlated, this matrix can become singular or ill-conditioned, leading to numerical instability.

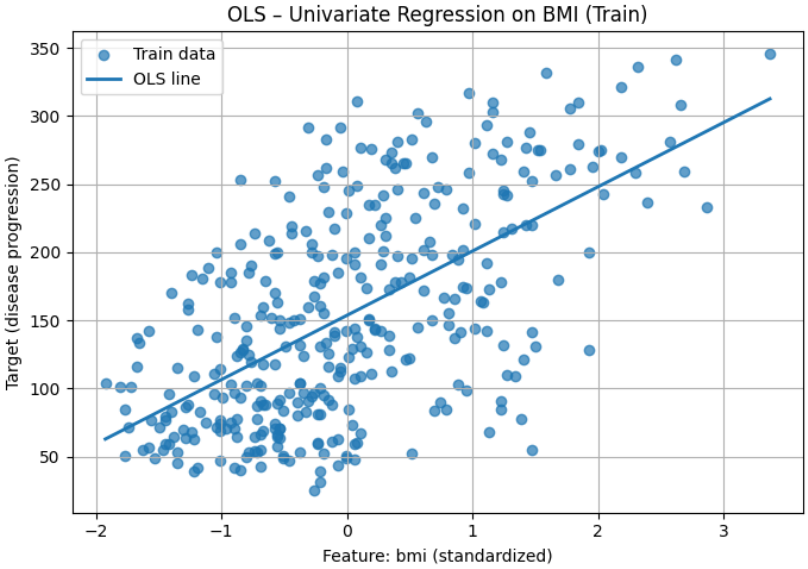

Fig 1: Univariate regression showing the relationship between BMI (standardized) and disease progression.

2.2 SVD-Based Solution

To address potential singularity, we utilized Singular Value Decomposition where

$$X = U\Sigma V^T$$

. The weights were calculated using the Moore-Penrose pseudoinverse:

$$\hat{\beta} = V\Sigma^{+}U^{T}y$$

. This method relies on SVD to handle near-zero singular values robustly.

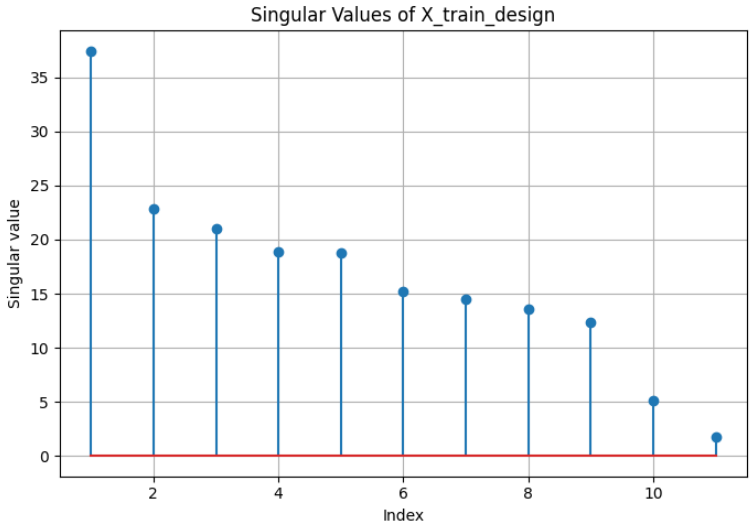

Fig 2: Singular Values of the design matrix. The rapid decay suggests potential redundancy in features.

2.3 Gradient Descent

We implemented Batch Gradient Descent, an iterative optimization method using the update rule:

$$\beta_{k+1} = \beta_k - \eta \nabla L(\beta_k)$$

. We experimented with learning rates (η) of 0.0001, 0.001, 0.01, and 0.1 to find the optimal convergence speed.

2.4 PCA and Dimensionality Reduction

We utilized SVD to compute principal components. The data was projected onto the top-k components, and the regression was re-run to analyze the trade-off between variance explained and prediction error.

3. Results

3.1 Model Performance Comparison

The analytical methods (OLS and SVD) produced identical results. The Gradient Descent method converged closely to the analytical solution. Interestingly, the PCA-reduced model (with k=4) slightly outperformed the full OLS model on the test set, likely due to a regularization effect.

Table 1: Summary of Mean Squared Errors (MSE)

Method	Parameters	Training MSE	Test MSE
OLS (Normal Eq)	All features	2868.55	2900.19
SVD (Pseudoinverse)	All features	2868.55	2900.19
Gradient Descent	LR = 0.1	2869.33	2895.30
PCA + OLS	**Top k=4**	3005.82	**2874.18**

3.2 Gradient Descent Convergence

We tested multiple learning rates. A learning rate of **0.1** proved most effective, minimizing the loss efficiently without diverging.

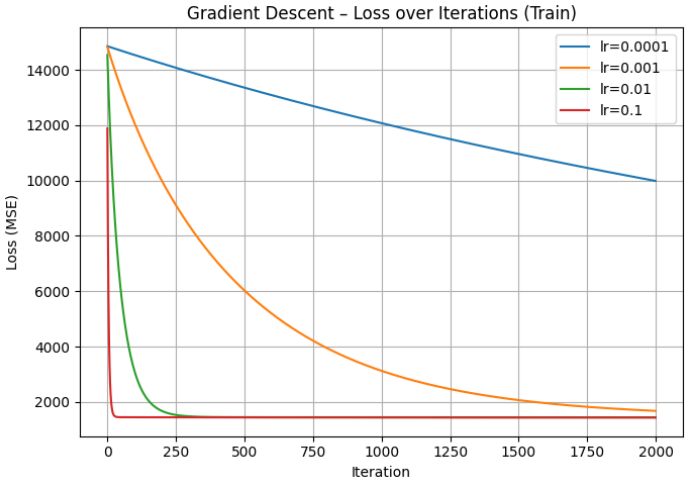

Fig 3: Loss function vs. Iterations. Larger learning rates (0.1) converged significantly faster than smaller ones.

3.3 PCA Analysis

We analyzed the Test MSE across different numbers of principal components (k).

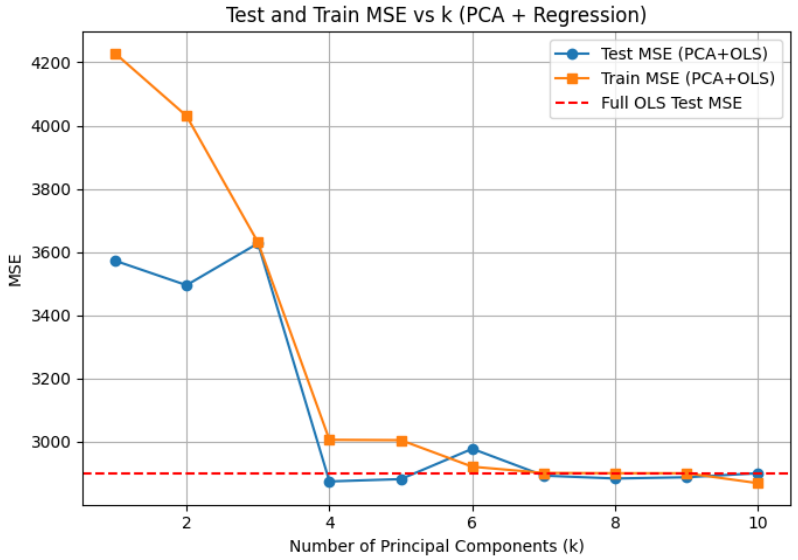

Fig 4: Test MSE vs. Number of Principal Components (k). The minimum error is achieved at k=4.

4. Discussion

- **Numerical Stability:** Singular values decline without hitting zero, showing a full-rank but ill-conditioned matrix; SVD stays stable even when OLS fails.
- **Residual Analysis:**

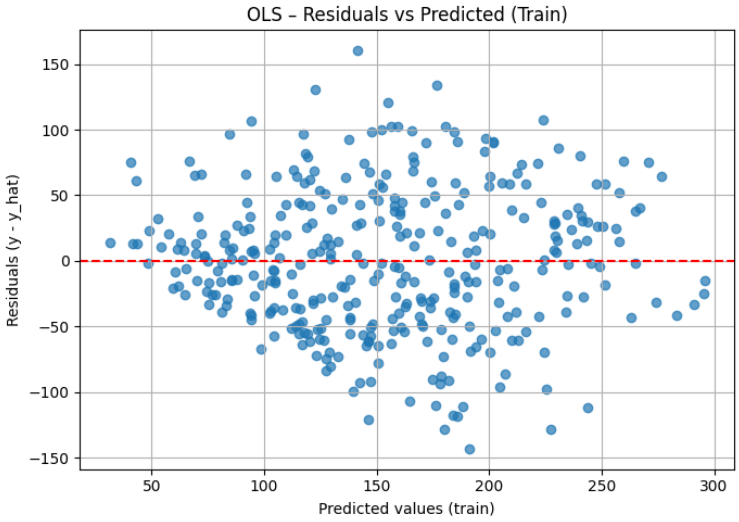

Fig 5: Residual plot. The errors are randomly distributed around zero, validating the linear assumption.

- **Dimensionality:** The PCA results were significant: using the top 4 components (77% variance) gave a lower Test MSE (2874.18) than all 10 features (2900.19), indicating PCA reduces noise and overfitting.

5. Conclusion

In this project, we successfully implemented linear regression from scratch using NumPy. We demonstrated that:

1. **SVD** is a robust alternative to the normal equation for matrix inversion.
2. **Gradient Descent** is a viable iterative alternative but requires careful tuning of the learning rate (0.1 was optimal here).
3. **PCA** is a powerful tool for this dataset; reducing the feature space to 4 dimensions actually improved the model's predictive performance on unseen data.

References

[1] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html

[2] https://numpy.org/doc/

GitHub Project: https://github.com/Aaish-Developer26/Mathematical-Foundations-For-Artificial-Intelligence