Evaluating Analytical and SGD-Based Regression Methods for Parkinson's Telemonitoring and Breast Cancer Diagnosis

Shivam Aery - 260896005
Victor Ghattas - 261180673
Amine Alexandre El Khoury - 261076412

September 2025

1 Abstract

The project task consisted of performing data pre-processing and basic statistic computation for the Parkinsons telemonitoring dataset and Breast Cancer diagnostic dataset. We then implemented analytical linear regression for the Parkinsons telemonitoring dataset,logistic regression for the Breast Cancer diagnostic dataset and mini-batch stochastic gradient descent for both linear and logistic regression. For all models, we report their performance for both training sets and test sets. We reported the weights of each of the features in our trained models. We sampled growing subsets of the training data and determined how that impacted the performance of our models. We tested growing minibatch sizes and determined the corresponding performance of Linear Regression using SGD and Logistic Regression using SGD. We presented the performance of linear regression and logistic regression using three different learning rates and lastly we compared analytical linear regression solution with a mini-batch stochastic gradient descent based linear regression solution.

2 Introduction

Several studies have been completed that showcase the result of using various machine learning methods on the Parkinsons telemonitoring dataset, one of which used machine learning methods such as Dimensionality Reduction, Singular Value Decomposition and other techniques to improve the time complexity and accuracy of Parkinsons diagnosis systems [1]. The Breast Cancer diagnostic dataset has also been used for various studies, one of which presents a detailed comparison of the effectiveness of six machine learning algorithms which include: GRU-SVM, Linear Regression, Multilayer Perceptron (MLP), Nearest Neighbor (NN) Search, Softmax Regression and Support Vector Machine (SVM) [2]. In our own exploration of these datasets, we begun our experiment by implementing analytical linear regression for the Parkinsons telemonitoring dataset. We obtained the MSE (Mean Squared Error) of linear regression and the accuracy score of logistic regression. We performed this experiment multiple times with various training test splits which included: 80/20, 70/30, 60/40, 50/50, 40/60, 30/70 and 20/80. We implemented SGD for both linear and logistic regression and tested both of their performances using growing minibatch sizes which are: 8, 16, 32, 64, 128 and fully-batched. We compared the performance of both linear and logistic regression using varying learning rates of 0.001, 0.01 and 0.1. To complete our experiment, we compared the performance of analytical linear regression with mini-batch SGD linear regression.

3 Implementation

The objective of this task is to implement linear and logistic regression models from scratch, each with their fundamental logic and with mini-batch stochastic gradient descent (SGD) model. Before building the models, helper functions were used for multiple tasks such as bias handling, with a function that appends a column of ones to the features matrix and add the bias to the weight vector and simplify predictions, a function that reshapes one-dimensional inputs into column vectors, and a logistic sigmoid function for the logistic regression. Each method is built with a fit function to train the data by modifying its parameters and a predict function to predict outputs from sets of inputs

3.1 Linear Regression

This method uses the least square methods from NumPy (np.linalg.lstsq) to minimize the square error loss, which is the difference between predictions and targets, and give an exact solution for the weights.

$$w = \left(X^T X\right)^{-1} X^T y \tag{3.1}$$

Although this analytical solution is efficient for small datasets, it can become unstable for larger ones. For this reason, a gradient function was implemented.

$$\nabla J(w) = \left(\frac{1}{N}\right) X^T (X.w - y) \tag{3.2}$$

Finally, the method was extended with mini-batch SGD. Instead of computing the gradient on a full dataset, this function only chooses a single data point at each iteration. It aims to compute the gradient with a different cost for each batch and update the weight at a lower cost than the fully batched method, while introducing a stochastic optimization to it. The sampling randomness is handled with NumPy's random generator. SGD is highly dependant on hyperparameters like the batch-size and the learning rate, which will be modified during the training.

3.2 Logistic Regression

Logistic Regression differs from the first implemented method by applying the sigmoid function below to the weighted input in the predict and gradient functions.

$$\sigma(w^T x) = \frac{1}{1 + e^{-w^T x}} \tag{3.3}$$

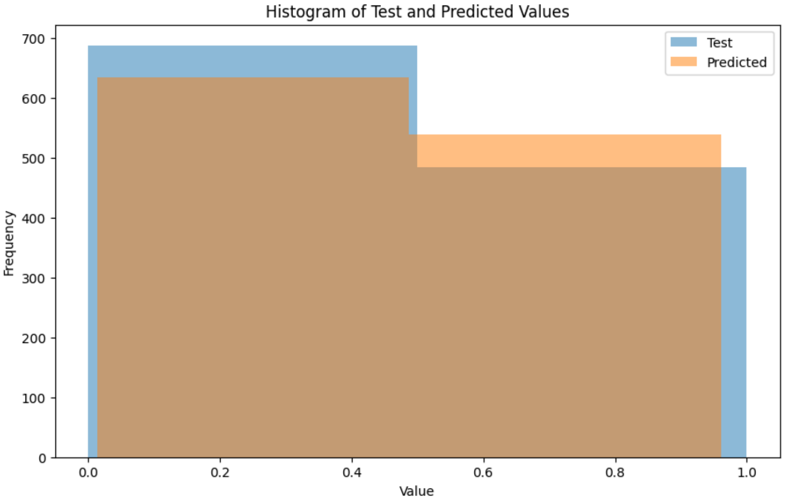
Instead of predicting values, this function's output is given as a probability between 0 and 1. The gradient of the class uses that same function instead of using the weights. The SGD implementation mirrors the one for the Linear Regression, with the key difference being the use of the logistic function instead of just the error.

4 Datasets

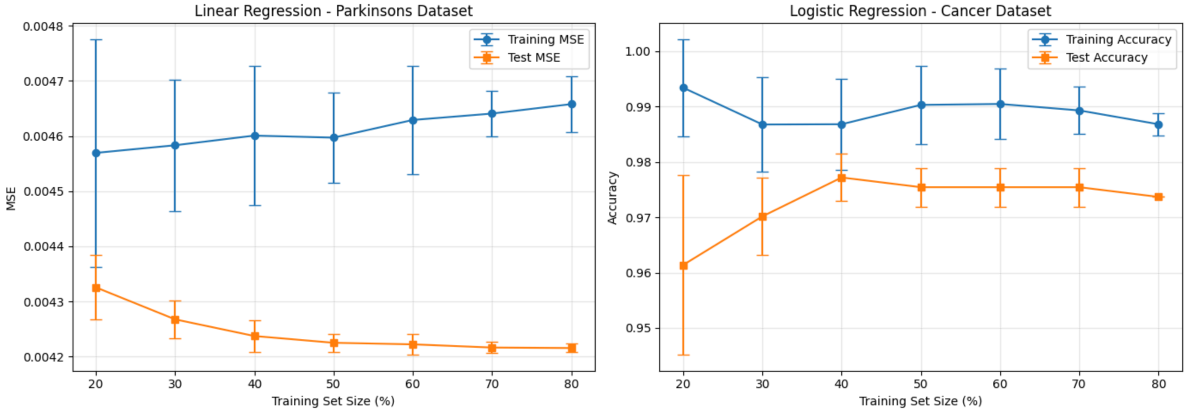
The two datasets we analyzed include the Breast Cancer diagnostic dataset that provides the characteristics of cell nuclei extracted from a fine needle aspirate of a breast mass. The other dataset is the Parkinson's Disease telemonitoring Dataset which contains biomedical voice measurements from 42 people with early-stage Parkinson's disease. The processing involved loading both datasets as dataframes, checking and removing rows that contain empty values. In both our datasets there were no rows with empty values. We attempted to remove duplicate rows but in both datasets, there were no duplicate rows. We computed the count, mean, standard deviation, minimum, first quartile, median and third quartile over all columns of both datasets. We found that the number of malignant diagnoses was 212 and benign diagnoses was 357 in the breast cancer dataset. We applied Min-Max normalization to all numerical columns in both datasets to ensure that no columns would contain disproportionate magnitudes with respect to other columns. As a result, this prevents the model from becoming unbalanced with respect to the importance of every feature of the input. We removed identifier columns in both datasets (ID in the breast cancer dataset, subject# in the parkinsons dataset). We performed binary mapping of the Diagnosis column of the breast cancer dataset such that M = 1 and B = 0. We removed all negative values in the test_time column in the parkinsons dataset since test_time should never be negative. When working with these datasets, it is vital that data is gathered from patients with the patients being aware that the data will be used in machine learning algorithms. If patients are not made aware of this, that would constitute a major breach of confidentiality and privacy.

5 Results

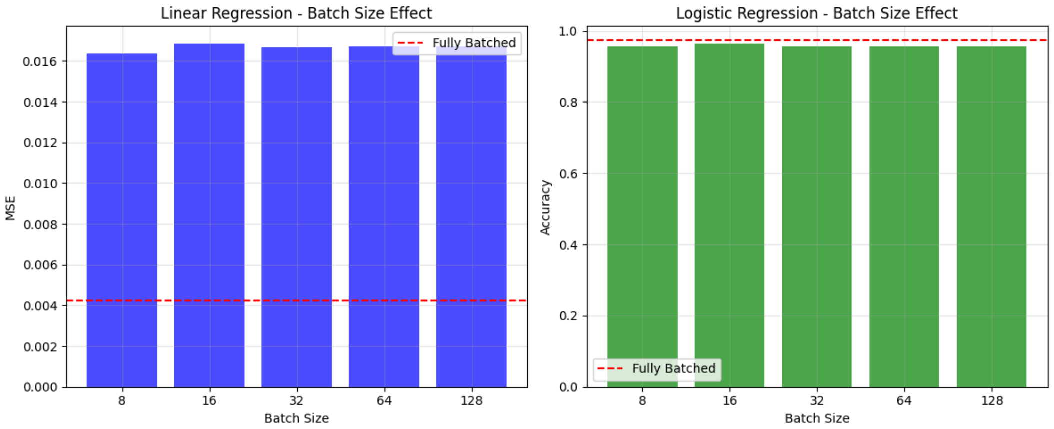
Our experiments evaluated Linear Regression on the Parkinson's dataset and Logistic Regression on the Breast Cancer dataset across multiple dimensions. For baseline performance (Task 3.1), Linear Regression achieved a test MSE of 0.0042 on the Parkinson's dataset using the analytical solution, while Logistic Regression obtained 97.37% test accuracy on the Cancer dataset. The training set performance showed minimal overfitting, with training MSE of 0.0047 and training accuracy of 98.68% respectively. Additionally, histogram analysis of the value distributions revealed that the Linear Regression model's predictions closely match the distribution of actual test values for the Parkinson's dataset, with both concentrated in the 0.0-0.6 range of normalized UPDRS scores, further confirming the model's ability to capture the underlying data patterns.



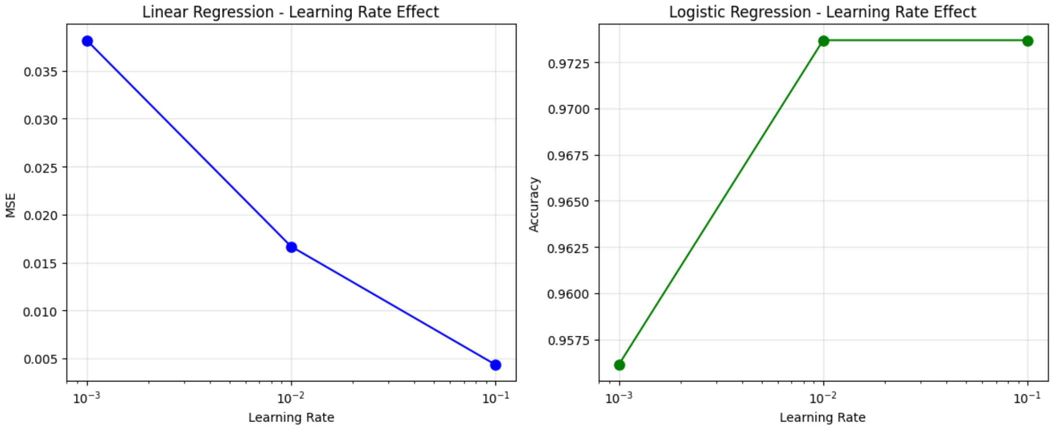
Analysis of training set size effects (Task 3.3) revealed that both models benefit from increased training data, though with diminishing returns. Linear Regression test MSE improved from 0.0043 at 20% training data to 0.0042 at 80%, while Logistic Regression accuracy increased from 96.1% to 97.5% over the same range. The relatively small performance gaps between training and test sets across all subset sizes indicate robust generalization for both models.

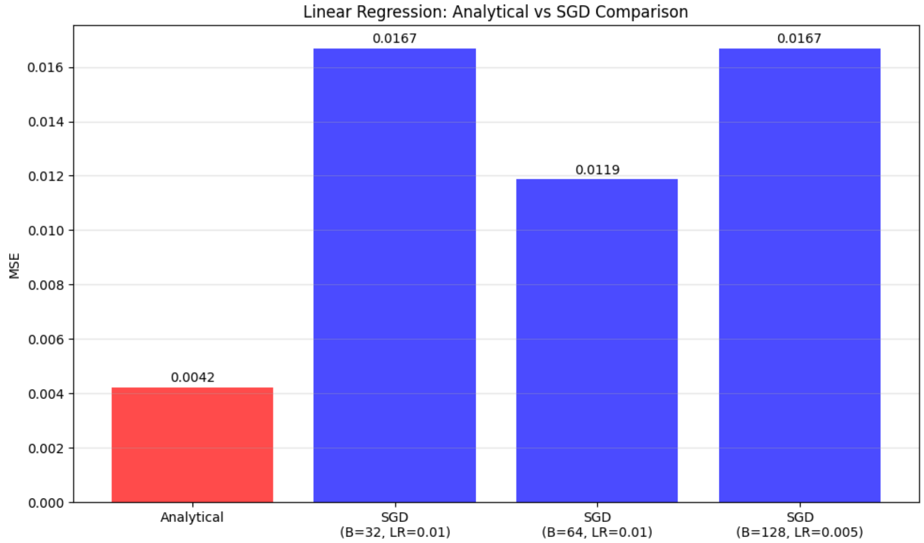


Mini-batch size analysis (Task 3.4) exposed a significant discrepancy between optimization methods for Linear Regression. While the analytical solution achieved MSE of 0.0042, all SGD variants with batch sizes from 8 to 128 plateaued around MSE of 0.016-0.017, suggesting convergence issues. Conversely, Logistic Regression showed consistent performance (97% accuracy) across all batch sizes, demonstrating robustness to this hyperparameter.

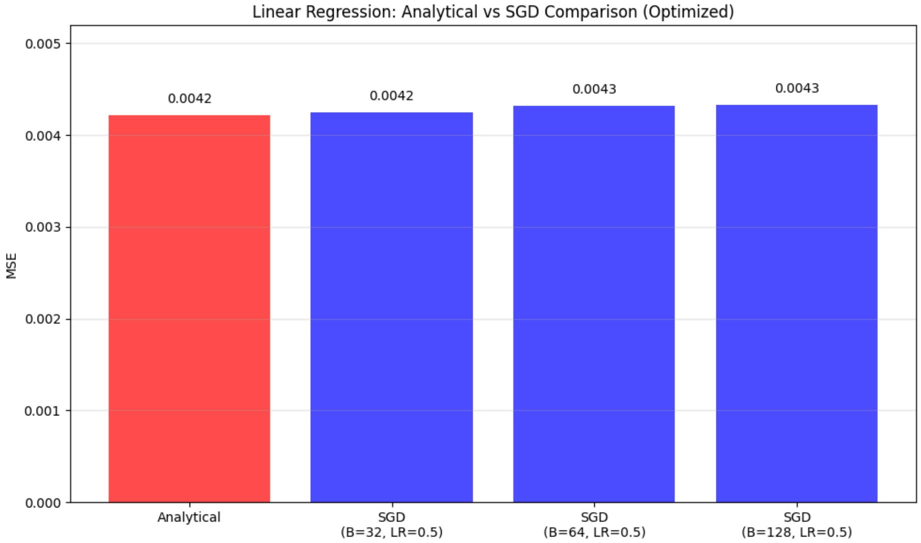


Learning rate experiments (Task 3.5) highlighted the sensitivity of gradient-based optimization. Linear Regression MSE varied from 0.038 at learning rate 0.001 to 0.0043 at 0.1, while Logistic Regression maintained stable performance (95.6-97.4% accuracy) across the same range. The comparison between analytical and SGD solutions (Task 3.6) confirmed that our SGD implementation requires further hyperparameter tuning, as even the best SGD configuration (batch=64, LR=0.01) achieved only MSE of 0.0119 compared to the analytical solution's 0.0042.





The comparison between analytical and SGD solutions in Task 3.6 prompted us to run a systematic hyperparameter search, an additional experiment which revealed that SGD can match analytical performance exactly. With optimized parameters (learning rate=0.5, iterations=5000), SGD achieved the same MSE of 0.0042 as the analytical solution, demonstrating that proper tuning enables gradient-based methods to find the global optimum.



6 Discussion and Conclusion

In conclusion, this project demonstrated from scratch implementations of linear and logistic regression models on the two datasets and how different optimization methods can impact their efficiency. Linear regression performed extremely well on the Parkinson's dataset with an analytical solution that showed minimal MSE, while logistic regression computed a stable and high accuracy on the Breast Cancer dataset. On the other hand, although the SGD underperformed at first, finer tuning of batch sizes, learning rates, and iterations achieved comparable accuracy to the analytical solution. Overall, this project's results emphasized on the importance of method selection and hyperparameters' sensitivity analysis.

7 Statement of Contributions

In this assignment, Shivam did Task 1, Task 3.1 and 3.2, Amine did Task 2, and Victor did the rest of Task 3. Each member wrote their given task on the report, Shivam wrote the abstract, the introduction and the datasets sections, Amine wrote the conclusion and did the report formatting and clean-up. Victor wrote the report section for results.

8 References

Bibliography

[1] M. Nilashi, O. Ibrahim, S. Samad, H. Ahmadi, L. Shahmoradi, and E. Akbari, “An analytical method for measuring the Parkinson’s disease progression: A case on a Parkinson’s telemonitoring dataset,” *Measurement*, vol. 136, pp. 545–557, 2019.

[2] A. F. M. Agarap, “On breast cancer detection: an application of machine learning algorithms on the wisconsin diagnostic dataset,” in *Proceedings of the 2nd international conference on machine learning and soft computing*, 2018, pp. 5–9.