COMP551-A1-Report

Jawdat Al-Jabi, Tal Smith, Steven Thao

September 2025

Abstract

This project explores the implementation and analysis of two machine learning models: Linear Regression and Logistic Regression, and their applications to two distinct datasets. The first, Parkinson's Telemonitoring, involves predicting motor skill scores using linear regression. The second, Breast Cancer Diagnostic, involves a binary classification using logistic regression with gradient descent. Both models were implemented from scratch in Python without reliance on ML libraries. Experiments include comparisons between fully-batched and mini-batch gradient descent, the effect of training data size, different learning rates, and feature contributions. Using our implementation, we were able to achieve a test MSE loss value as low as about 61 for regression while we also obtained accuracy values closing in on 96.5% for the classification task. The results demonstrate the practical application of these classical models, the importance of careful pre-processing, and the influence of hyperparameters on model performance, providing a foundation for understanding more advanced machine learning techniques.

1 Introduction

The purpose of this project is to examine two datasets using different machine learning models:

1. Linear regression is used to predict motor scores ($motor\_UPDRS$) of patients with Parkinson's disease.
2. Logistic regression is used to predict the presence of breast cancer in a patient given their test results.

Within the project, we use various methods, with the aim to compare them:

- We report the performance of our models when using an 80/20 train/test split.
- We compare the results of using differently sized subsets of the training data.
- We verify the accuracy of stochastic gradient descent given different batch sizes.
- We compare the performance of gradient descent given different learning rates.
- We compare the the analytical linear regression solution and that using gradient descent.

Numerous notable results were discovered through our experimentation, including, but not limited to:

- Larger train splits lead to higher test accuracy although too much training can counterintuitively lead to worse testing accuracy.
- Smaller batch sizes generally mean faster convergence, with some exceptions.
- Higher learning rates mean higher accuracy.

2 Datasets

The two datasets that were used were:

1. Parkinsons Telemonitoring [1]: Biomedical voice measurements of 42 people with early-stage Parkinson's. The trial lasted 6 months and consisted of 16 different voice measures. The target value is the motor_UPDRS, which is a value quantifying the loss of motor skills due ot the disease. We have observed that the data was often bunched up around a defined range of values for many features, which would reduce the robustness of our model, but the occasional spikes in UPDRS value does help provide a better model.

2. Breast Cancer Wisconsin (Diagnostic) [2]: Descriptions of cell nuclei characteristics are used to classify possible breast cancer tumours as Malignant (M) or Benign (B). Some features, such as radius1, show clear value ranges (e.g., smaller radii usually indicate benign tumours), making diagnosis easier. Others, like fractal_dimension3, contribute little and are expected to have small weights. In general, when group means are distinct, higher weights are expected; when they overlap, weights approach zero.

   Note: The features age, sex, and test_time were dropped from dataset 1, as the dataset's stated aim is to predict motor_UPDRS and total_UPDRS from the 16 voice measures.

The datasets were pre-processed by transforming the features into a design matrix and the targets were put in vectors. Thereafter, we have standardized the data to have a zero mean, unit standard deviation to facilitate convergence and prevent exploding gradients / weights while training.

3 Results

3.1 80/20 analytic linear regression & full-batch logistic regression

For our linear regression model, we obtained a MSE of 58.6705 on our training data, and 61.2742 on our testing data. For logistic regression, we obtained an accuracy of 0.9868 on our training data, and 0.9561 on our testing data.

3.2 Weights of models trained in 3.1

Dataset 1: Top 10 features by weight in different orders

Feature	**Weight**
Jitter(Abs)	-54022.809240
Jitter:RAP	-31539.713646
Jitter:DDP	10605.220591
Shimmer:APQ3	-3878.273253
Shimmer:DDA	1251.161776
Shimmer:APQ5	-243.273964
Jitter(%)	151.198116
Shimmer:APQ11	148.697495
Shimmer	126.048493
Jitter:PPQ5	100.297881

Table 1: By absolute weight value

Feature	**Weight**
Jitter:DDP	10605.220591
Shimmer:DDA	1251.161776
Jitter(%)	151.198116
Shimmer:APQ11	148.697495
Shimmer	126.048493
Jitter:PPQ5	100.297881
PPE	19.250856
RPDE	0.443245
HNR	-0.424450
Shimmer(dB)	-2.817814

Table 2: By positive weight value

Feature	**Weight**
Jitter(Abs)	-54022.809240
Jitter:RAP	-31539.713646
Shimmer:APQ3	-3878.273253
Shimmer:APQ5	-243.273964
NHR	-29.175656
DFA	-28.829803
Shimmer(dB)	-2.817814
HNR	-0.424450
RPDE	0.443245
PPE	19.250856

Table 3: By negative weight value

Features such as Jitter(Abs), Jitter:RAP, Shimmer:APQ3 contribute largely to a low score. Features like Shimmer:DDA, Jitter:DDP contribute largely to a high score. Features with decreasing absolute weight are less impactful on the score.

Dataset 2: Top 10 features by weight in different orders

Feature	Weight
texture3	0.638094
radius3	0.626967
area3	0.609820
perimeter3	0.599590
radius2	0.583438
concave_points3	0.547129
concave_points1	0.542509
texture1	0.516008
smoothness3	0.503896
area1	0.501992

Table 4: By absolute weight value

Feature	Weight
texture3	0.638094
radius3	0.626967
area3	0.609820
perimeter3	0.599590
radius2	0.583438
concave_points3	0.547129
concave_points1	0.542509
texture1	0.516008
smoothness3	0.503896
area1	0.501992

Table 5: By positive weight value

Feature	Weight
fractal_dimension1	-0.303404
fractal_dimension2	-0.263464
compactness2	-0.228651
symmetry2	-0.186450
concavity2	-0.083579
symmetry1	0.008997
texture2	0.025409
compactness1	0.036833
concave_points2	0.074981
smoothness2	0.090984

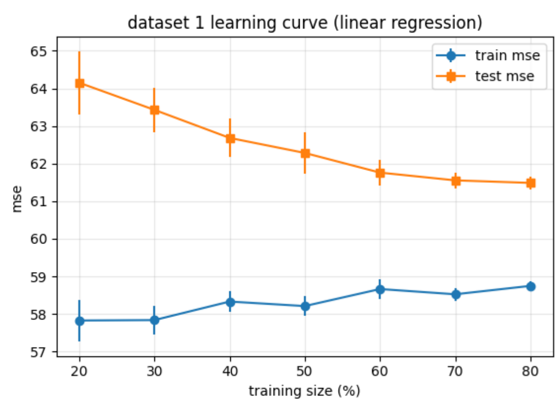
Table 6: By negative weight value

Features with the highest absolute weights have the highest impact on probability. Features with the most positive weights have the most impact on a high probability, while features with a negative weight have impact on a low probability of malignancy.

3.3 Linear regression and Logistic regression trained with GD with various dataset splits

For dataset 1, the MSE on the used training data increases slightly as training data accumulates. However, the accuracy of the test data becomes higher (lower MSE) as more data is used.

For data set 2, the accuracy of the training data slowly decreases. As training size increases, accuracy on the test data slowly increases.



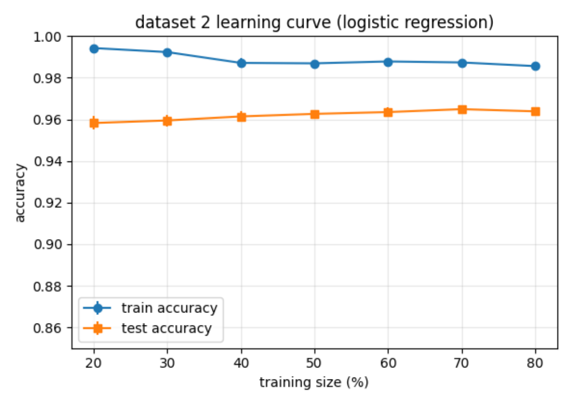


Figure 1: Learning curves for Dataset 1 (left) and Dataset 2 (right) vs. various dataset splits

3.4 Testing different minibatch sizes

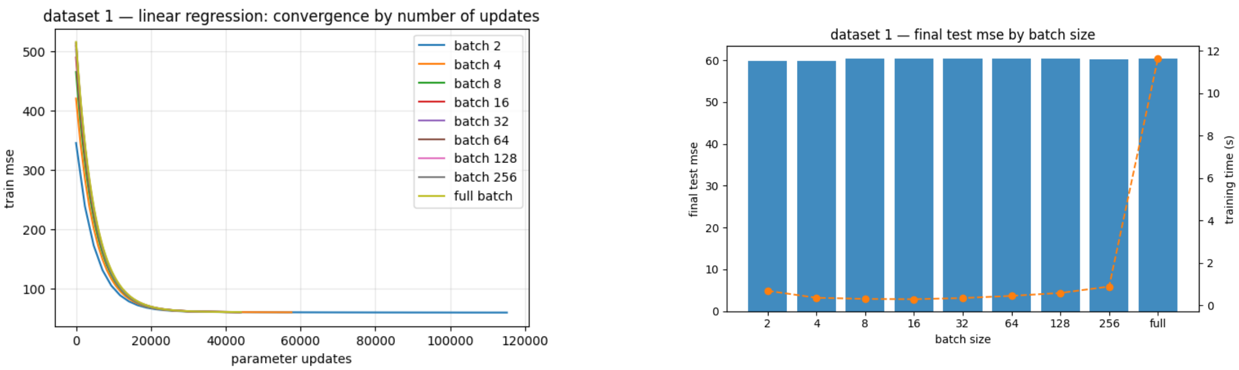


Figure 2: Learning curves for Dataset 1 vs. various minibatch sizes

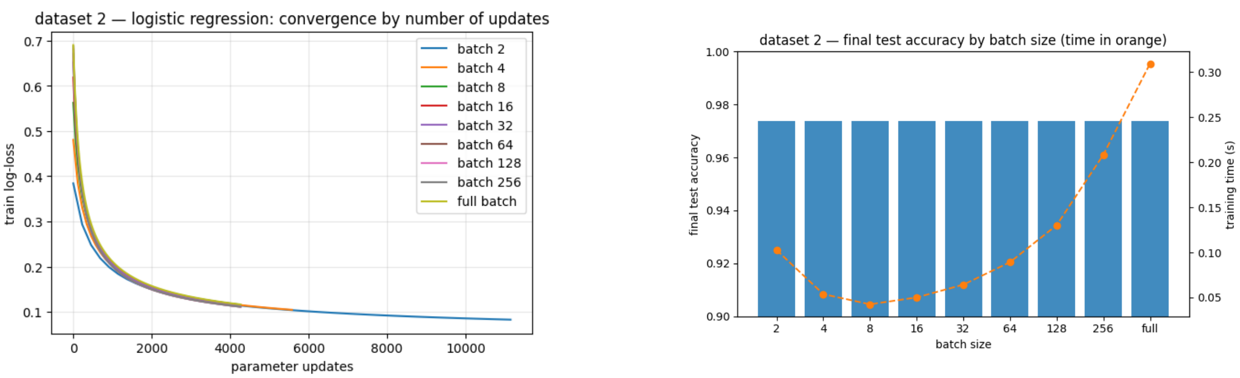


Figure 3: Learning curves for Dataset 2 vs. various minibatch sizes

According to the graphic above, for linear regression and logistic regression, a smaller batch leads more quickly to a lower convergence. If we judge by generalization (test error/accuracy) + compute time, the sweet spot is a mini-batch, not full batch. We would recommend a minibatch size of about 8-16.

3.5 Testing different learning rates

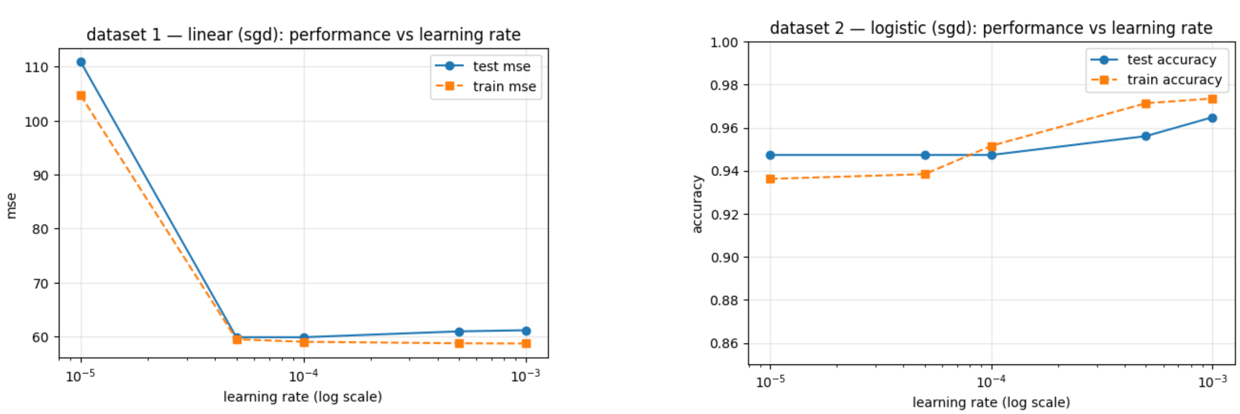


Figure 4: Learning curves for Dataset 1 (left) and Dataset 2 (right) vs. various learning rates

As shown above, higher learning rate contributes to accurate learning. In dataset 1, the MSE decreases for training and testing data when moving from a small learning rate to an average one. However, further increasing the learning rate does no good for the model. In dataset 2, a higher learning rate consistently improves the accuracy for both the training and testing data.

3.6 Comparing analytic vs. SGD solutions for linear regression

Method	**Train MSE**	**Test MSE**
Analytical (closed-form)	58.670	61.274
SGD (mini-batch, std)	58.948	60.042

Table 7: Linear regression comparison for Dataset 1.

As shown above, the analytical solution performs slightly better on the training data, but worse on the test data when compared to SGD. This is an example of a slight overfitting, where the model performs better on the data it has been trained on than all data.

4 Discussion and Conclusion

Below are some key takeaways from this project:

- Weights with the largest absolute values have the strongest impact: positive weights raise predictions, negative weights lower them.
- Accuracy on test data improves with more training data, while training accuracy decreases slightly—indicating broader **learning** rather than narrow **memorization**.
- Smaller batch sizes speed up convergence, though very small ones (e.g., 2 or 4) make training unstable due to noisy gradients.
- Higher learning rates ($> 10^{-4}$) generally improve accuracy for both linear and logistic regression.
- The analytical (closed-form) solution may overfit training data, reducing test performance.
- Regularization (L1/Lasso, L2/Ridge) can improve generalization, especially with many correlated features.
- Non-linear transformations or polynomial expansions may capture relationships beyond linear/logistic regression.

5 Statement of Contributions

Part	**Contributor**
Part 1	Steven Thao
Part 2	Tal Smith
Part 3	Jawdat Al-Jabi
Code Review	All
Project Write-Up	All

Table 8: Project Contributions

References

[1] A. Tsanas and M. A. Little, "Parkinsons telemonitoring dataset," https://archive.ics.uci.edu/dataset/189/parkinsons+telemonitoring, 2009, uCI Machine Learning Repository.

[2] W. H. Wolberg, O. L. Mangasarian, N. Street, and W. Street, "Breast cancer wisconsin (diagnostic) [dataset]," UCI Machine Learning Repository, 1993. [Online]. Available: https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic