# Assignment 1: Exploring Linear and Logistic Regression

COMP 551: Applied Machine Learning

Audréanne Bernier (261100643)

Anjara Trachsel-Bourbeau (261119884)

Ben Coull-Neveu (261116508)

McGill University

Department of Computer Science

October 2, 2025

**Abstract**

In this project, we investigated the performance of linear regression and logistic regression machine learning models on two public datasets, with the goal of understanding how these widely-used models behave in practice, and how their predictive performance & efficiency depends on certain parameters & other conditions. We implemented gradient descent for both models, and an analytic solution for linear regression. We examined the impact of train set size, number of batches, and the impact of learning rates.

Model performance were assessed using their loss functions and accuracy, the latter only be used to evaluate the logistic regression model. Across both ML models, performance was sensitive to data partitioning, with test losses occasionally being lower than training losses. The number of batches was always inversely proportional to convergence time, though correlations to performance differed between the two; it negatively affected linear regression performance, but moderately improved that of logistic regression until a sweet-spot was achieved at $\sim 2^6$. Learning rates were found to have a profound effect on the loss stability predominantly, with large rates diverging.

Furthermore, we noted that strongly correlated features could be problematic and introducing regularization could improve generalization while retaining more features for training.

## 1 Introduction

This project was aimed at exploring the common Machine Learning models of linear regression and logistic regression by creating our own implementation of these widely used tools. These two regression models are both instances of supervised learning, meaning that we know what the *true* values will be and can use that to train our models. This is usually done by randomly splitting the data into a training and a test set, where for the former the model has access to the *true* target values to evaluate itself, and the latter is used to make predictions. Once we have made predictions, we can evaluate how well our model performed based on how accurate our fit target values were to the *true* values. Linear regression is used to predict continuous outputs, like price, or in our case, Parkinson's motor scores. Logistic regression is used to predict categorical, or binary, outputs. In our case, this represents getting diagnosed with breast cancer or not.

Two datasets were analyzed to evaluate the performance of our linear regression model and our logistic regression model. Dataset 1 corresponds to the Oxford Parkinson's Disease Telemonitoring Dataset (Tsanas and Little, 2009) and was used to predict Parkinson's motor scores, while Dataset 2 was obtained from the Diagnostic Wisconsin Breast Cancer Database (Wolberg et al., 1993) and was used to make breast cancer predictions. The Oxford Parkinson's Disease Telemonitoring was developped out of the need to have a less cost intensive way of establishing testing for Parkinson's (Tsanas et al., 2009). It proved to be very effective and enabled future large-scale clinical trials by providing an easier way to track the progression of the disease. Similarly, the model developed from the Diagnostic Wisconsin Breast Cancer Database was very significant as it provided a good first analysis tool with it's 86% accuracy rate (Street et al., 1993). While the authors of these two papers performed much more thorough analysis of their data, they did both perform some more basic tests similar to the ones we will be using in this project.

For this project, we investigated the impact of varying the size of the training data set on the performance of both the training data and the test data. We found that the test loss was consistently lower then the train loss for both linear regression model, while it higher for the logistic regression. We also experimented with using various mini-batch sizes and the impact it had on the convergence speed and performance of our models. We found that for both logistic and linear regression, the convergence speed decreased as the number of batches increased. The performance of the logistic regression improved as the number of batches increased, while the performance of the linear regression decreased. For the convergence speed and the performance of both model, the error or standard deviation between trials increased with the number of batches. Furthermore, we looked at the learning rate, $\alpha$, and how that could also impact the performance. We found that a too low learning rate would cause the gradient descent to converge at a very low speed, or not even converged in time. On the other hand, a learning rate that was too high would not converge at all. We conclude that there is a sweet spot in the middle that needs to be achieved. Lastly, we investigated different evaluation metrics by considering the $R^2$ value on top of the loss value to see if it had an impact. We found that in most cases, the loss value and the $R^2$ followed the same trends and gave us the same information about our models' performance.

## 2 Datasets

When we are talking about these datasets, it is important to consider the ethical implications of this data as patient privacy is of the most utter importance. All recognizable features/information has been removed to mitigate the risks of re-identification. It is also important to consider the demographic imbalance of the dataset in itself, in which case our model would follow these biases and could result in worse modeled outcomes for underrepresented groups. As we do not have access to the ethnic demographic of the data, we can only comment about the age and sex distributions. The average age of the participants is 64.8 years old, with most participants being above 55 years old. This bias most likely stems from who actually has Parkinson's symptoms as the chances increase with age. As for the sex distribution, one is represented twice a much as the other in the dataset. This will inevitably create bias in our models.

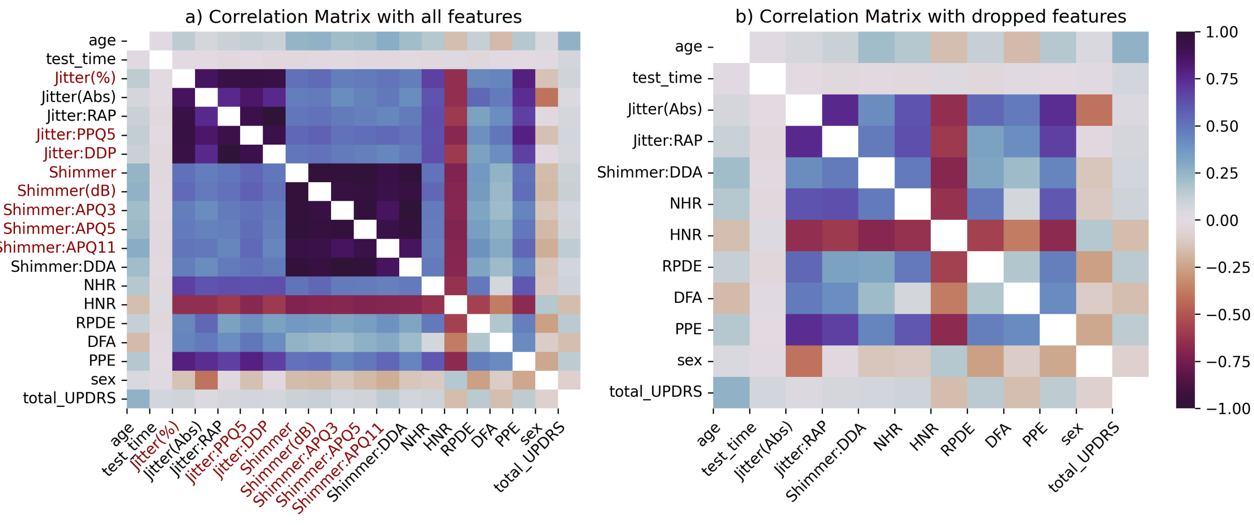


Figure 1: a) Correlation matrix with all features. Pairs of features exhibiting correlations above a threshold of |0.8| were evaluated. In said pair, the feature with higher correlation with rest of the features, in red, were dropped. b) Correlation matrix without overly correlated features. These features will be used in the analysis.

Preprocessing the datasets involved a number of steps. First, any empty rows were removed, and all features were verified to be properly formatted. Assuming the data is approximately Gaussian distributed, data with z-score $> 3$ were deemed outliers and removed for each feature. The remaining data was then normalized so that each feature has a mean of 0 and a standard deviation of 1. Finally, correlations between all features were evaluated, and features exhibiting correlations above a threshold of 0.8 were removed to avoid multicollinearity. The results of this correlation-based cleaning are shown in Figure 1. Figure 2 also compares the distributions of the raw and cleaned data for three features of Dataset 1, highlighting that features such as *HNR* or *Shimmer:DDA* show a notable reduction in counts around the mean. This occurs because outliers in other features, like *DFA*, were removed even when these features remained near the mean in certain other features. All *shimmers* and *jitters* features exhibited Poisson features as seen in Figure 2a). Additionally, *test_time* and *sex* were removed, since the former is random and has little relevance for a regression, and the latter is because we deemed it unnecessary to include. As Figure 2 illustrates, some feature distributions deviate from a true Gaussian shape. While this may introduce minor inaccuracies either by removing more than just outliers or by leaving some extreme values, the $3\sigma$ cutoff proved sufficient for the purposes of this project and the training of our models. The fourth panel shows the distribution of the targets.

A similar cleaning, normalizing and correlation analysis was applied to Dataset 2. The features that were kept were: *radius1*, *texture1*, *smoothness1*, *symmetry1*, *fractal_dimension1*, *radius2*, *texture2*, *smoothness2*, *concavity2*, *concave_points2*, *symmetry2*, *fractal_dimension2*, *smoothness3*, *compactness3*, *concave_points3*, *symmetry3*, *fractal_dimension3*. The distribution of the individual features were also approximated by a Gaussian distribution. There were Poisson and Gaussian distributed features in the final selection, but no bimodal distributions. The target *Diagnosis* is binary, with patients having either benign ($B$) or malignant ($M$) breast tumours. Since it is

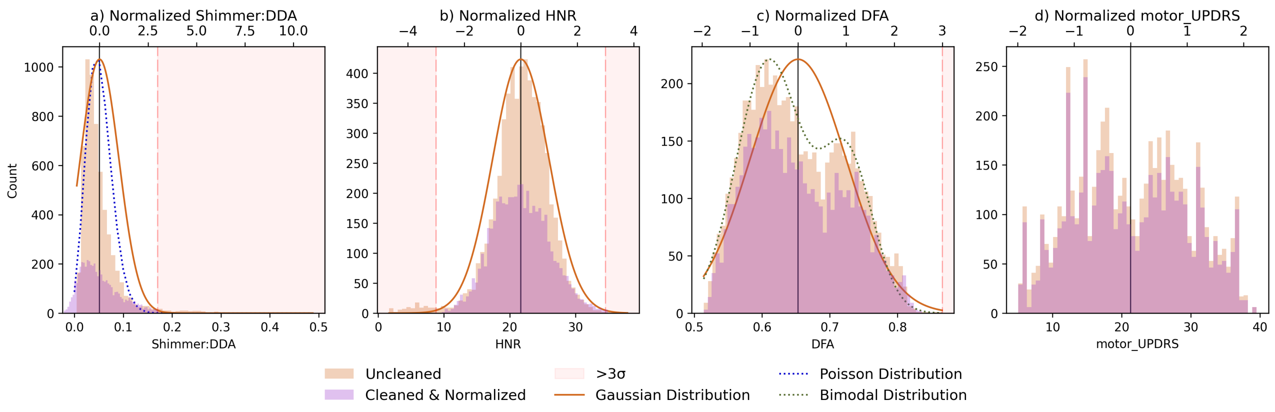


Figure 2: Examples of the distributions present in Dataset 1 and distribution of the target. Unprocessed data (N = 5875) in orange and cleaned & normalized data (N = 4954) in pink. A Gaussian approximation of the data was performed in orange. Outliers were identified through a variation $> 3\sigma$, shown in light red. a) Poisson distribution example with *Shimmer:DDA*. The underlying Poisson distribution is shown in blue. b) Gaussian distribution example with *HNR*. c) Feature *DFA* was bi-modally distributed. An approximate bimodal distribution is shown in green. d) Distribution of the target motor_UPDRS.

binary, we opted to encode $B$ to 0 and $M$ to 1 instead of implementing *OHE*, for the sake of simplicity.

## 3 Results & Discussion

The results for the 80/20 train/test splits are tabulated in Table 1. The analytic linear regression achieves a better loss value compared to its gradient descent counterpart. For all gradient descents, convergence occurs either when the gradient falls below $10^{-3}$, if the training loss stabilizes with $\Delta\text{loss} < 10^{-6}$, or if epoch 1200 is reached.

Model performance was primarily assessed using the loss function, reported as *train loss* for the training set and *test loss* for the test set. While other metrics such as $R^2$ were also used to evaluate the models, they provided little insight beyond what could be obtained using loss. For logistic regression, accuracy was also used as an additional metric, as that is a more intuitive way of measure of classification performance.

For linear regression, the analytic solution consistently outperforms the gradient descent method, even after many epochs and high learning rates, although the latter can get pretty close given enough epochs. This is to be expected, since the analytic solution is by definition the best fit of the data. The test loss is consistently lower than the training loss, indicating the model fits the test data better than it does the training data. This is unlikely, but the results suggest that this is the case given our dataset. Compared to the linear regression model, the logistic regression's loss is higher, albeit slightly, indicating that both models perform similarly on their respective data. The most heavily-weighted features for Dataset 1 and Dataset 2 are tabulated in Table 2a and 2b, respectively.

| Method | Train Loss | Test Loss | Train accuracy | Test accuracy |
|---|---|---|---|---|
| Analytic linear regression on Dataset 1 | 0.101452 | 0.09741 | — | — |
| Linear GD regression on Dataset 1 | 0.101819 | 0.096919 | — | — |
| Logistic GD regression on Dataset 2 | 0.165881 | 0.171629 | 0.967836 | 0.929412 |

Table 1: Performance of linear regression and fully batched logistic regression with an 80/20 train/test split. Reporting the loss and accuracy for both the train and the test data.

For Dataset 1, *total_UPDRS* is the most strongly weighted by a significant margin, indicating a possible overfit. For Dataset 2, the bias has the most significant impact while the weights are generally above 0.25 in magnitude, with only a few weights much closer to 0, suggesting that that there is no overfitting occurring. The features shown are those that most strongly influence the classification, creating a relatively steep separation along these dimensions compared to other features.

(a) Linear regression (analytical and GD) on Dataset 1.

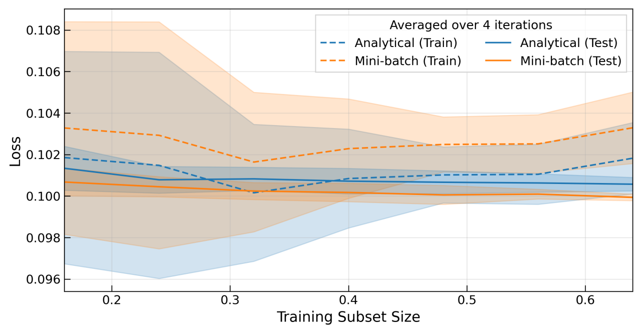
| Linear regression | total_UPDRS | Jitter(Abs) | Jitter:RAP | PPE | NHR | bias |
|---|---|---|---|---|---|---|
| Analytical feature weight | 0.948630 | -0.195526 | 0.079019 | 0.059908 | 0.038977 | -0.027283 |
| GD feature weight | 0.945387 | -0.164240 | 0.065796 | 0.052497 | 0.027091 | -0.025301 |

(b) Fully batched logistic regression on Dataset 2

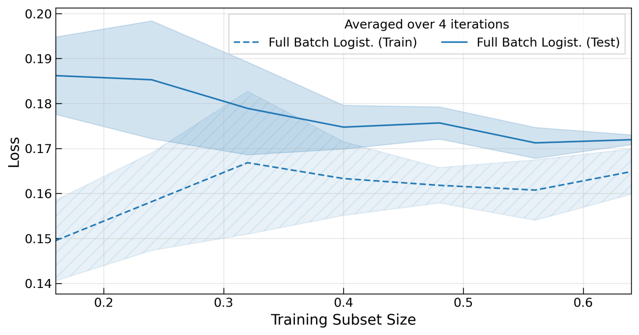
| Logistic regression | radius1 | concave_points3 | radius2 | texture1 | smoothness3 | bias |
|---|---|---|---|---|---|---|
| Feature weight | 0.732891 | 0.614979 | 0.589815 | 0.582076 | 0.410345 | -1.168772 |

Table 2: Feature weights of our models from Table 1 with an 80/20 split. Only the 5 most-weighted features are shown, as well as the bias. Linear regression with gradient descent (GD) was done using 16 mini-batches.

Figure 3 shows the performance of the models when only using subsets of the training set. For both linear and logistic models, the test loss and the variance between iterations decrease as the training set grows, indicating the model becomes more generalized with more training data. The loss for logistic regression exemplifies how the model overfits the training data, and thus underfits the test data when using a smaller training subset, but generalizes nicely with more training data.

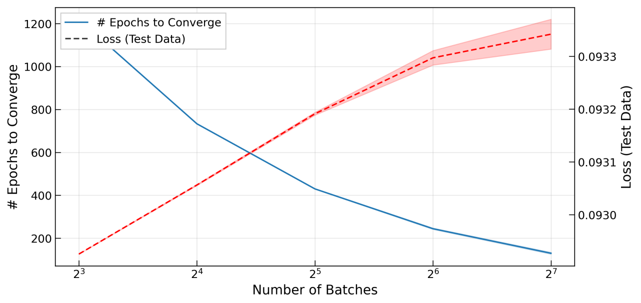


(a) Linear regression

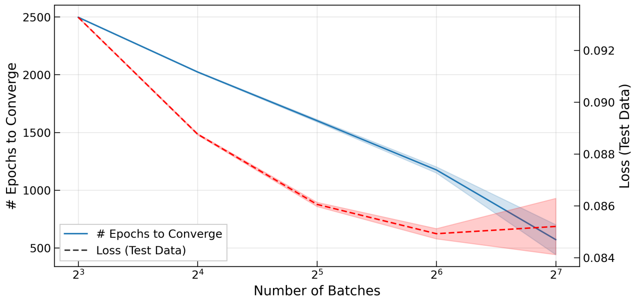


(b) Logistic regression

Figure 3: Effect of increasing the training split on the performance. The dashed lines represent the performance of the training set. The full lines represent the performance of the test set. a) Linear regression with both the analytical method and the mini-batch gradient descent. b) Logistic Regression with full batch. The shaded regions represent the standard deviation.



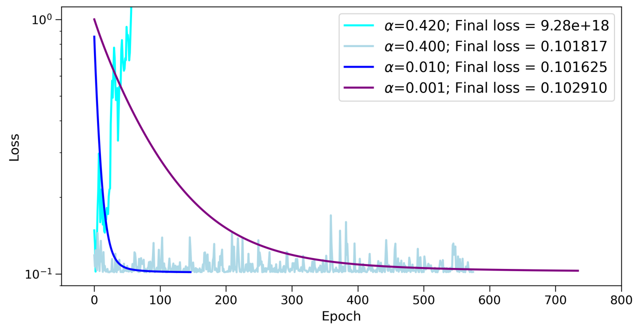
(a) Linear regression



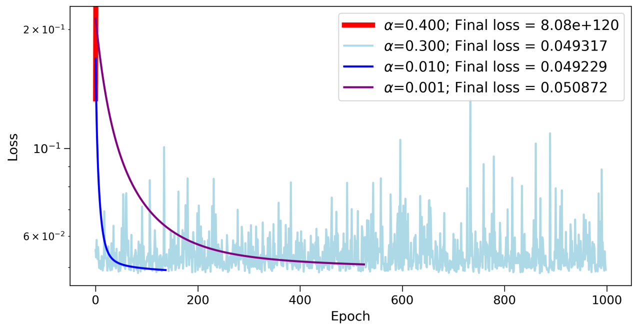
(b) Logistic regression

Figure 4: Effect of increasing the number of batches on the performance (minimizing the loss) in blue and the convergence speed in red for a) linear regression and b) logistic Regression. The shaded regions represent the standard deviation.

Figure 4 shows the time to convergence and loss as a function of number of batches. Both models converge quicker as we increase the number of mini-batches, however for the linear regression it comes at the cost of performance since we see an increase in the loss values. For the logistic model, the performance isn't negatively impacted until $\sim 2^6$, at which point the performance becomes nominally worse. This suggests that using more batches doesn't necessarily result in a better performance, and that it should be tested when training a model to optimize efficiency and performance.



(a) Linear regression



(b) Logistic regression

Figure 5: Effect of learning rate on performance and the evolution of the loss function over many epochs for a) linear regression and b) logistic regression with mini- and full-batch gradient descent, respectively. $\alpha = 0.4$ line in (b) has been highlighted since the loss diverges very quickly.

Figure 5 shows the performance and convergence time given different learning rates. Lower learning rates converge far more slowly, but have similar performance to more modest learning rates, while faster rates result in noisier loss and sometimes, if large enough, no convergence at all. Note that lower learning rates can perform worse than modest rates if the convergence constraints are the same, since the $\Delta$loss or number of iterations thresholds will be reached at higher loss (see $\alpha = 0.001$ in Figure 5). In other words, the loss with lower rates will attain a shallower slope at higher loss values, which will impact the performance of the model.

## 4 Conclusion

We implemented and analyzed two of the most used machine learning models: linear regression and logistic regression. By training and testing these models on real datasets, we observed how factors such as training set size, number of batches, and learning rate influence performance and convergence of these models. Our results confirm that the analytical solution for linear regression consistently outperforms gradient descent, but that gradient methods can achieve comparable results given enough epochs and optimal training parameters. Logistic regression demonstrated strong predictive power for binary classification, with consistently high accuracy across different configurations. We demonstrated the importance of careful preprocessing, parameter tuning, and evaluation when applying regression models to real-world datasets.

An immediate improvement to these models would consist of integrating regularization into the pipeline, as opposed to filtering out features that were strongly inter-correlated. This could allow for more generalized models given the presence of more features while mitigating the risk of multicollinearity or overfitting.

## 5 Statement of Contributions

All members contributed equally to the analysis and presentation of the data, to the implementation and training of the model, and to the writing of this report.

## References

William Nick Street, William H. Wolberg, and Olvi L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Electronic imaging*, 1993. URL `https://api.semanticscholar.org/CorpusID:14922543`.

Athanasios Tsanas and Max Little. Parkinsons Telemonitoring. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C5ZS3N.

Athanasios Tsanas, Max A. Little, Patrick E. McSharry, and Lorraine O. Ramig. Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests. *IEEE Transactions on Biomedical Engineering*, 57:884–893, 2009. URL `https://api.semanticscholar.org/CorpusID:7382779`.

William H. Wolberg, Olvi L. Mangasarian, and Nick Street. Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository, 1993. DOI: https://doi.org/10.24432/C5DW2B.