# Applied Machine Learning: Mini-Project 1

JAN TIEGGES, RISHABH THANEY, JONATHAN COLAÇO CARR

ABSTRACT: This report provides a thorough analysis of gradient-based optimization methods for linear and logistic regression. Using the Boston Housing and Wine Datasets, we investigate how the performance of these models change with respect to several variables including the training set size, batch size, learning rate and optimization method. Our best linear regression model achieves a Mean Squared Error of 5.396 on the Boston Housing dataset and our best logistic regression model achieves perfect accuracy on the Wine dataset. Lastly, we experiment with augmenting the Boston Housing Dataset through Gaussian and Sigmoid transformations. Our findings are consistent with the theories presented in class lectures.

## 1. Introduction

Linear and logistic regression are two of the most common models in machine learning [PSb]. The aim of this project was to gain hands-on experience in implementing both of these models as well as to understand how they are trained with first order optimization methods. Our linear regression experiments were carried out on the Boston Housing [bos] while the logistic regression experiments were performed on the Wine dataset [AF91]. Both datasets have been widely used to study regression and classification models [Aga, Goy, Ece, Tan].

For both linear and logistic regression, we performed a suite of experiments to study the effect of various hyperparameter choices including the effects of test size, cross validation, mini-batch size, momentum, learning rate, and feature augmentation on the overall performance. We found that training on small data sets like poses challenges in the variance of results, emphasizing the importance of pre-processing and cross-validation for evaluation. Moreover, we observed the interdependence of parameter choices, underlining the importance of performing extensive tests on a range of choices. We also found that more sophisticated optimization methods like the Adam optimizer improved the model's performance and convergence rate. These results are consistent with the theory presented in class lectures [PSb, PSc, PSa].

## 2. Datasets

This section highlights our key findings and preprocessing decisions for both datasets. A more extensive data analysis is available in our code.

**Boston Housing Dataset:** The Boston Housing dataset [bos] consists of samples collected from U.S. Census Data that contains price housing information for 506 houses in the Boston area. The data was originally collected by Harrison and Rubinfeld [HR78]. The dataset contains 14 variables such as the per capita crime rate by town (CRIM) and the average number of rooms per dwelling (RM). We used the Median Value of Owner-occupied Homes in $1 000s (MEDV) as the target variable for each sample. As shown in Figure 1, the distribution of the target variable is approximately Normal with a mean of 22.5, but spikes at the value 50, indicating that the price was perhaps capped here. Many of the other features in the dataset had heavily imbalanced distributions (see Figure A.1). For example, the feature CHAS had 471 samples with 0 and only 35 with 1. The imbalance and occurrence of outliers were considered in our preprocessing (discussed below).

In Figure A.3 we identified both linear relationships (e.g. RM, and NOX) and non-linear relationships (e.g. LSTAT, DIS, and ZN) between the features and target variable. Our correlation analysis (provided in Figure A.4) shows that only 4 features have positive correlation with the target, with RM (0.7) and ZN (0.36) having the highest positive correlation, and LSTAT (-0.73) and PTRATIO (-0.5) the highest negative correlation. Finally, we looked at the correlations between the different features and found some strong correlations. As shown in Figure A.4, TAX and RAD had the highest positive correlation (0.91) and DIS and NOX the highest negative correlation (-0.77). Although we did not perform any dimensionality reduction in this report, these correlations may be useful to identify redundant information.

As a final note, the Boston Housing dataset raises some ethical concerns. We excluded one feature, the portion of Black residents by town as a feature, due to historical bias an ethical concerns. The small size of the dataset also lends itself to potential bias, since it may not be representative of the Boston housing market. Lastly, the risk of de-anonymizing the data plays a role when working with Census data.

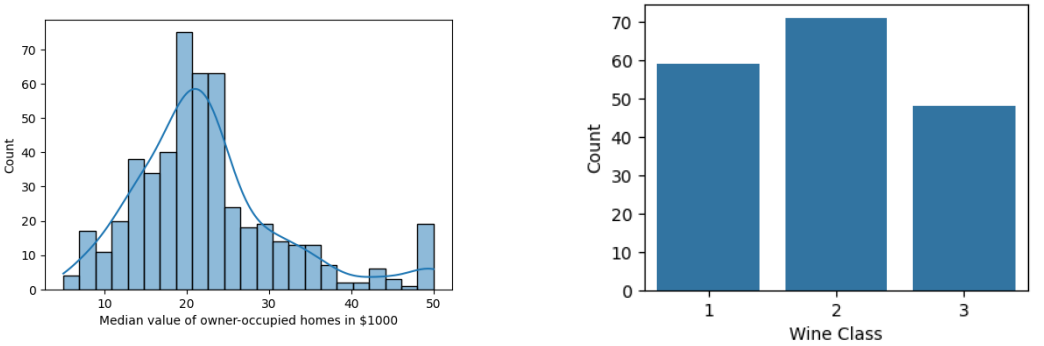


Figure 1. Distribution for the target values of the Boston Housing dataset (left) and Wine dataset (right).

**Wine Dataset:** The Wine dataset [AF91] provides a chemical analysis of wines made with grapes grown from three different Italian cultivators. It consists of 178 wine samples and 13 features, such as the alcohol content and flavanoids. Figure 1 shows the proportion of samples from each of the three cultivars, who we refer to "wine classes". The wine classes were distributed relatively equally, with the second class having the highest representation.

Our analysis of the Wine dataset revealed Normal distribution patterns in several features as well as correlations between features and the wine classes. As shown in Figure A.5, many of the features including Alcohol, Ash, Alcalinity of Ash, Total Phenols, Nonflavanoid Phenols, Proanthocyanins, and OD280/OD315 of Diluted Wines exhibit characteristics of a Normal distribution. However, some features such as Malic acid display sharp spikes at specific values. The bivariate analysis provided in Figure A.7 reveals interesting relationships between features and the target. The means, medians and standard deviations of the individual features differ substantially between the different classes. Sometimes clear patterns can be identified; for instance, there is a clear linear relationship between the Total Phenols and wine class index. It can be seen in Figure A.6 that class 3 is less sensitive to outliers, while class 2 has the most outliers across all features. Overall, this preliminary analysis shows that the features are a good indication of the respective wine classes. Lastly, we studied the correlation between the individual features. As shown in Figure A.7, the highest positive correlation is observed between 'Flavanoids' and 'Total phenols' (0.86), whereas the lowest negative correlation exists between 'Hue' and 'Malic acid' (-0.56). Combined, these results suggest that the dataset is well suited for the logistic regression classification problem.

**Data Preprocessing:** In preparation for our experiment, we applied some pre-processing steps to the data. Both datasets do not contain null values, but we did remove some samples as part of the outlier removal. For this we applied the interquartile range (IQR) method, which holds the data within a certain range determined by the first (Q1) and third quartile (Q3) of the data [VPS18]. The scale factor here is usually 1.5, but we also experimented with 1.0 (more strict) and 2.0 (less strict). We also applied min max scaling [PS15] to the Boston Housing dataset so that the range of values for each feature was in between 0 and 1.

## 3. Results

In this section we present the respective results of Linear Regression applied to the Boston Housing dataset and Logistic Regression applied to the Wine dataset. For linear regression we compared the analytical solution and mini-batch Stochastic Gradient Descent (SGD) optimization methods. For Logistic Regression

| Split \| Method | MSE | |
|---|---|---|
| | Train | Test |
| 80/20 \| Analytic | 6.775 | 6.179 |
| 80/20 \| SGD | 17.691 | 14.688 |
| 5-CV \| GD | $6.518 \pm 0.292$ | $7.062 \pm 1.140$ |
| 5-CV \| SGD | $16.588 \pm 0.320$ | $16.885 \pm 1.284$ |

TABLE 1 *Mean Squared Error (MSE) of Baseline Linear Regression Model*

| Split \| Method | Accuracy | | Precision | | Recall | | F1 Score | |
|---|---|---|---|---|---|---|---|---|
| | Train | Test | Train | Test | Train | Test | Train | Test |
| 80/20 \| GD | 0.938 | 1.000 | 0.940 | 1.000 | 0.938 | 1.000 | 0.937 | 1.000 |
| 80/20 \| SGD | 0.922 | 1.000 | 0.926 | 1.000 | 0.922 | 1.000 | 0.921 | 1.000 |
| 5-CV \| GD | $0.935 \pm 0.017$ | $0.901 \pm 0.023$ | $0.940 \pm 0.014$ | $0.915 \pm 0.014$ | $0.935 \pm 0.017$ | $0.901 \pm 0.023$ | $0.933 \pm 0.018$ | $0.899 \pm 0.024$ |
| 5-CV \| SGD | $0.929 \pm 0.037$ | $0.913 \pm 0.037$ | $0.933 \pm 0.031$ | $0.917 \pm 0.033$ | $0.929 \pm 0.037$ | $0.913 \pm 0.037$ | $0.927 \pm 0.040$ | $0.912 \pm 0.037$ |

TABLE 2 *Performance Metrics of Baseline Logistic Regression Model. For cross validation, the mean and standard deviation are reported across each of the five folds.*

we compared full batch gradient descent (GD) with mini-batch SGD. We investigated a variety of different hyperparameters and as well as other factors including the strength of the outlier removal.

**Experiments 1 and 2:** In our first two experiments, we studied various performance metrics for our baseline models. To compare different metrics, we chose the following default hyperparameter settings; a batch size of 32, a learning rate of 0.1, a IQR factor of 1.5 and a limit of 100,000 gradient update steps. We analyzed our linear regression model in terms of mean squared error (MSE) and our logistic regression model in terms of accuracy, precision, recall, and f1 score. The results, shown in Tables 1 and 2, are on par with prior work [Aga, Goy, Tan, Ece].

For the linear regression model there is a clear discrepancy between the analytical and SGD solutions. For our default hyperparameters, the analytical solution shows a significantly stronger performance. This is not only seen in the 80/20 training split but also in the results of 5-fold cross validation. The latter indicates that the result is stable over the entire data set and not due to an imbalance in the test set. Both the analytic and SGD solutions show goods signs of generalization as the train error and test error lie very close to each other. For logistic regression, GD and SGD perform equally well with very subtle differences (see Table 2). The performance on the test set is even better than on the training set and reaches 1.0 for all scores. This already flags challenge in modelling with the Wine data set; its very small size causes reported scores to be polarized quickly. In this case, cross-validation is particularly useful, but here too these results indicate a very good performance. Here SGD seems to perform slightly better than GD. Both methods exhibit small standard deviations across all scores, indicating consistent performance across different folds. Overall, the performance of our baseline models are promising but there is still room for improvement, especially for linear regression with SGD.

**Experiment 3:** Our next task was to investigate the effect of the training set size on the performance of each model. As shown in Figure 2, when the training size is increased for each model the testing performance tends to increase and the training performance tends to decrease. However, this trend was slight, and each model was still able to perform well with small amounts of training data. This is consistent with our data analysis which showed that the data was well distributed and that there were clear relationships between feature and target variables. The slight trend in decreased training performance suggests that models tend to overfit on the training set when it does not have enough samples.

**Experiment 4++:** In our next suite of experiments, we investigated the effect of batch size, momentum and optimization method on the final performance and convergence rate for both models. For both linear

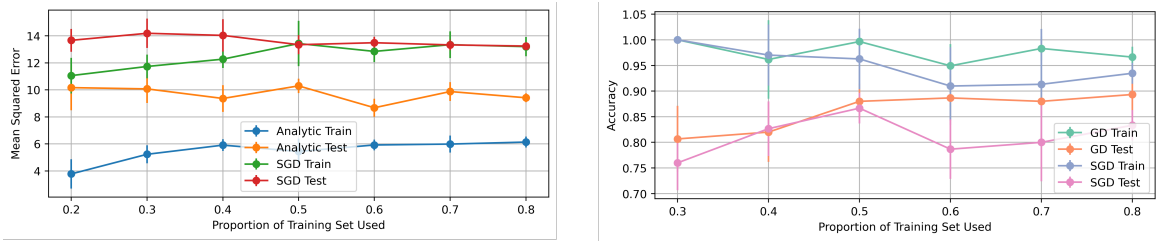


Figure 2. Effect of training set size on model performance for linear regression (left) and logistic regression (right). The performance is reported as the average over five trials, with error bars of one standard deviation.

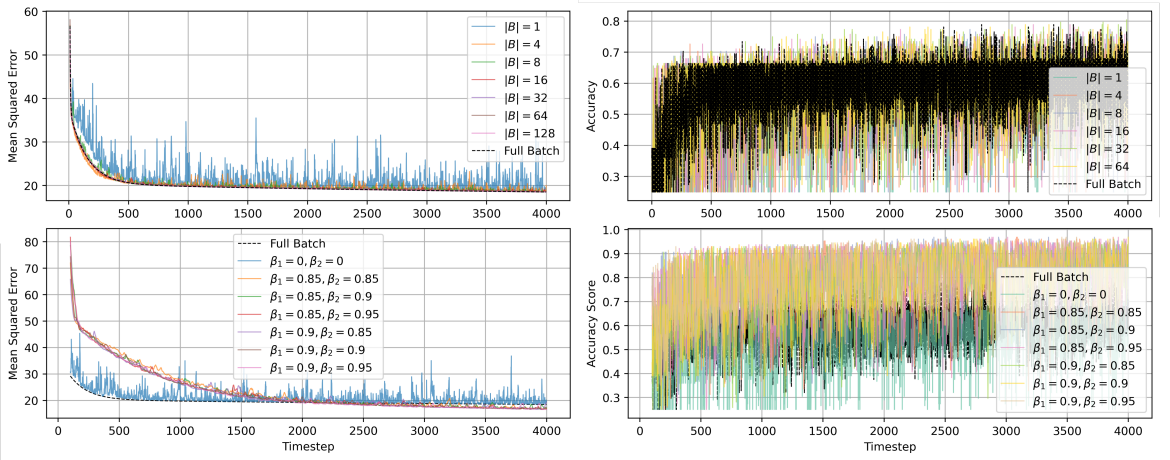


Figure 3. The effect of Batch size (top) and Adam optimizer (bottom) on the convergence rate of Linear Regression (left) and Logistic Regression (right) models, plotted for the first 4000 timesteps. The Adam optimizer experiments are compared with the standard Gradient Descent (Full Batch) and Stochastic Gradient Descent ($\beta_1 = 0, \beta_2 = 0$) optimization methods.

regression and logistic regression, the performance and overall runtime decreased with batch size (see Tables B.3 and B.4 the appendix). However, as shown in Figure 3, even small batch sizes (e.g. $|B| = 16$) were a good approximation of both the overall performance and per-timestep gradient update of the Full Batch baseline. Figure 3 also shows that the performance of the model during training was more volatile for smaller batch sizes, which reflects the fact that stochastic gradient descent only performs better at each timestep in expectation. While the additional momentum parameter made the model performance smoother with each timestep, it did not improve the rate of convergence (see Figure B.8 in the appendix). It was only when using both the momentum and adaptive learning rate (i.e. the Adam optimizer) that we were able to improve not only the stability during convergence, but also the final performance. In particular, the time-dependent weight updates used in the Adam optimizer allow it to outperform the fixed learning rate Gradient Descent method. This is consistent with the theoretical motivation for momentum and Adam optimizers discussed in class [PSa].

**Experiment 5:** Both the linear and logistic regression models performed very well at high learning rates, which was not expected. As shown in Tables B.5 and B.6, our linear regression models reached their maximum performance at learning rates of 1.0 and 0.5, respectively, whereas the logistic regression was quite consistent accords most learning rates. This was surprising given that learning rates are typically set around 0.1 [PSa].

**Experiment 6+:** We performed parameter search over a 540 parameter combinations for both models (Table B.9) and identified several notable relationships between hyperparameter choices. We performed 3-fold cross-validation for each parameter configuration and identified the best models according the mean validation

MSE and accuracy for linear regression and logistic regression, respectively. We chose these metrics because they are the most common and widely used metrics for evaluating these models, and used 3-fold cross-validation to reduce the variance of our results. The best 10 models and parameter configurations are shown in Tables B.7 and B.8. There are several notable findings from our hyperparameter search. For instance, an IQR factor of 1.0 is favored for linear regression, while logistic regression performs better with factors of 1.5 or 2.0. Batch sizes between 1 and 64 lead to strong results for linear regression. Batch size has a particular effect on the advantage of momentum. For logistic regression the best batch size is usually 16 or 32. Both models achieved better results when given more iterations for the convergence (100,000) indicating that they need a long time to fit to the data. As already indicated in Experiment 5, linear regression achieves the best results with a relatively high learning rate. However, as shown in Figure B.10, the results are much more unstable across different parameter choices at a high learning rate (0.7). For logistic regression, the learning rate is less important with consistent results over different learning rates, with the lowest (0.05) performing marginally better.

**Experiment 7+:** In addition to hyperparameter experiments, we investigated data transformations with Gaussian and Sigmoid basis functions for the Linear Regression model. We applied five Gaussian basis transformations to the ZN feature and five Sigmoid basis transformations to the DIS feature. These two features were selected because they exhibited some of the strongest non-linear relationships with the target variables in our data analysis. Both of these transformations improved the overall performance of the analytic linear regression model, as shown in Table B.10. However, the model that was trained with both Sigmoid and Gaussian basis features showed signs of overfitting, as it had the largest difference between training and testing error.

**Experiment 8:** The final experiment considered was a comparison between the speed and performance of analytical and mini-batch SGD methods for linear regression. For the Boston Housing dataset, the analytical method finds a better (indeed, optimal) solution in a shorter amount of time. While the analytical solution will always provide the optimal solution, the compute time for analytical linear regression may be much larger when there are many more features in the training data. Furthermore, the analytical fit cannot be obtained when the input feature matrix is not invertible. Thus, while the analytic solution is practical for the Boston Housing dataset, we expect mini-batch gradient based algorithms to be much more feasible for higher dimensional and potentially degenerate data.

## 4. Discussion and Conclusion

We have performed an extensive analysis of linear and logistic regression design choices for gradient based optimization methods. While our models were tested on small datasets, our results are largely consistent with theory that is known to scale to higher dimensional data. For instance, we found that relatively small batch sizes (i.e. of size 16) were good approximations to the full gradient update and produced competitive performance for both models. The Adam optimizer led to both a faster rate of convergence and a better performance than other gradient based methods, which we attribute in part to its time-dependent weight updates. Further analysis into other optimization methods such as Newton's method would be another interesting avenue of exploration. Our baseline models performed well with high learning rates [PSb], which we would like to study in greater detail in future experiments. We would also like to consider the effect of a time-dependent learning rate for SGD. We highlighted the importance of parameter optimization for mini-batch gradient descent and showed that gradient-based methods are able to approach the analytical optimal solution in the case of linear regression. Lastly, we investigated adding features to our dataset with Gaussian and Sigmoid basis functions. While this technique increased the performance of our model, it also was prone to overfitting to the training data. We conclude that the linear and logistic regression models are able to successfully make predictions on the Boston Housing and Wine datasets, which is consistent with prior work.

**Statement of Contribution:** All members contributed to experiments and report writing. RT performed the data analysis for the wine dataset, JCC implemented the models and JT performed data analysis for the boston housing dataset as well as data preprocessing.

## References


AF91. Stefan Aeberhard and M. Forina. Wine. UCI Machine Learning Repository, 1991. DOI: https://doi.org/10.24432/C5PC7J.

Aga. Animesh Agarwal.

bos. boston. http://lib.stat.cmu.edu/datasets/boston. Accessed: 2023-09-19.

Ece. EceDolen.

Goy. Mohit Goyal.

HR78. David Harrison and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978.

PSa. Isabeau Prémont-Schwarz. Gradient descent.

PSb. Isabeau Prémont-Schwarz. Linear regression.

PSc. Isabeau Prémont-Schwarz. Logistic regression.

PS15. SGOPAL Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

Tan. Aaron Tanjaya.

VPS18. HP Vinutha, B Poornima, and BM Sagar. Detection of outliers using interquartile range technique from intrusion dataset. In *Information and Decision Sciences: Proceedings of the 6th International Conference on FICTA*, pages 511–518. Springer, 2018.


# Appendix

## A. Additional Data Analysis

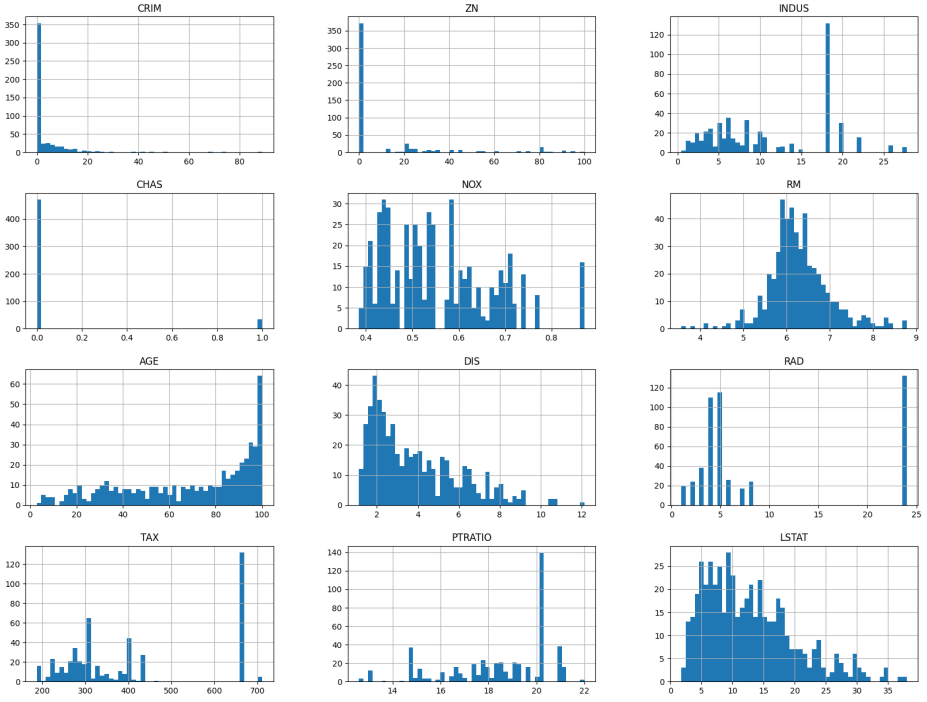


Figure A.1. Distribution of the features in the Boston Housing Dataset.


Figure A.2. Correlation of features to target in Boston Housing Dataset.

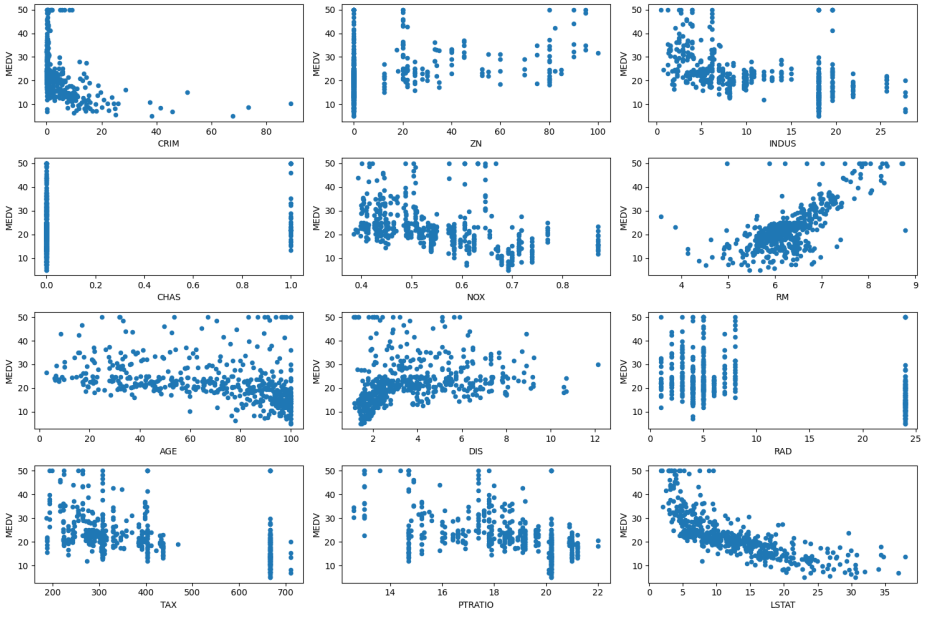


Figure A.3. Relationship of features to target in the Boston Housing Dataset.

## B. Additional Experimental Details

B.1. *Experiment 4*

| Batch Size | Train MSE | Test MSE | Convergence Time (s) |
|---|---|---|---|
| 1 | 15.68 | 16.4079 | 4.95 |
| 4 | 13.94 | 13.0283 | 4.82 |
| 8 | 13.46 | 12.9738 | 4.91 |
| 16 | 13.389 | 12.9855 | 5.06 |
| 32 | 13.365 | 13.1454 | 4.93 |
| 64 | 13.389 | 13.2915 | 5.09 |
| 266 | 13.363 | 13.2223 | 6.22 |

TABLE B.3 *Performance and Convergence Speed of Mini-batch Stochastic Gradient Descent for Linear Regression.*

| Batch Size | Train Accuracy | Test Accuracy | Convergence Time (s) |
|---|---|---|---|
| 8 | 0.85938 | 0.969697 | 7.41 |
| 16 | 0.92969 | 0.939394 | 7.51 |
| 32 | 0.95312 | 0.939394 | 7.98 |
| 64 | 0.88281 | 0.878788 | 8.74 |
| 128 | 0.95312 | 0.939394 | 10.4 |

TABLE B.4 *Performance and Convergence Speed of Mini-batch Stochastic Gradient Descent for Logistic Regression.*

| | $\alpha = 0.005$ | $\alpha = 0.01$ | $\alpha = 0.05$ | $\alpha = 0.1$ | $\alpha = 0.2$ | $\alpha = 0.5$ | $\alpha = 0.7$ | $\alpha = 1.0$ |
|---|---|---|---|---|---|---|---|---|
| Train MSE | 18.242 | 17.425 | 15.603 | 14.284 | 12.727 | 11.67 | 11.25 | 10.724 |
| Test MSE | 14.94 | 13.899 | 11.31 | 9.6628 | 8.0369 | 7.3306 | 6.4193 | 6.173 |

TABLE B.5 *Performance vs. Learning Rate for Linear Regression*

| | $\alpha = 0.005$ | $\alpha = 0.01$ | $\alpha = 0.05$ | $\alpha = 0.1$ | $\alpha = 0.2$ | $\alpha = 0.5$ | $\alpha = 0.7$ | $\alpha = 1.0$ |
|---|---|---|---|---|---|---|---|---|
| Train Accuracy | 0.82812 | 0.89844 | 0.96875 | 0.94531 | 0.95312 | 0.92969 | 0.71875 | 0.95312 |
| Test Accuracy | 0.87879 | 0.9697 | 0.93939 | 0.9697 | 0.93939 | 0.9697 | 0.66667 | 0.87879 |

TABLE B.6 *Performance vs. Learning Rate for Logistic Regression*

| IQR Factor | Batch Size | Learning Rate | Max Iters | Momentum | Mean Validation MSE | Test MSE |
|---|---|---|---|---|---|---|
| 1.0 | 32 | 0.7 | 100,000 | 0.9 | 8.515 | 5.396 |
| 1.0 | 64 | 0.7 | 100,000 | 0.9 | 8.521 | 5.559 |
| 1.0 | 32 | 0.7 | 100,000 | 0.5 | 8.532 | 5.375 |
| 1.0 | 32 | 0.7 | 100,000 | 0.0 | 8.620 | 5.372 |
| 1.0 | 64 | 0.5 | 100,000 | 0.5 | 8.662 | 5.821 |
| 1.0 | 8 | 0.5 | 100,000 | 0.0 | 8.702 | 5.891 |
| 1.0 | 64 | 0.5 | 100,000 | 0.0 | 8.752 | 5.887 |
| 1.0 | 1 | 0.5 | 100,000 | 0.9 | 8.769 | 9.384 |
| 1.0 | 64 | 0.7 | 100,000 | 0.5 | 8.789 | 5.417 |
| 1.0 | 1 | 0.5 | 100,000 | 0.5 | 8.797 | 8.455 |

TABLE B.7 *The 10 Linear Regression models with the lowest mean validation MSE across different paramater configurations (sorted from from top to bottom)*

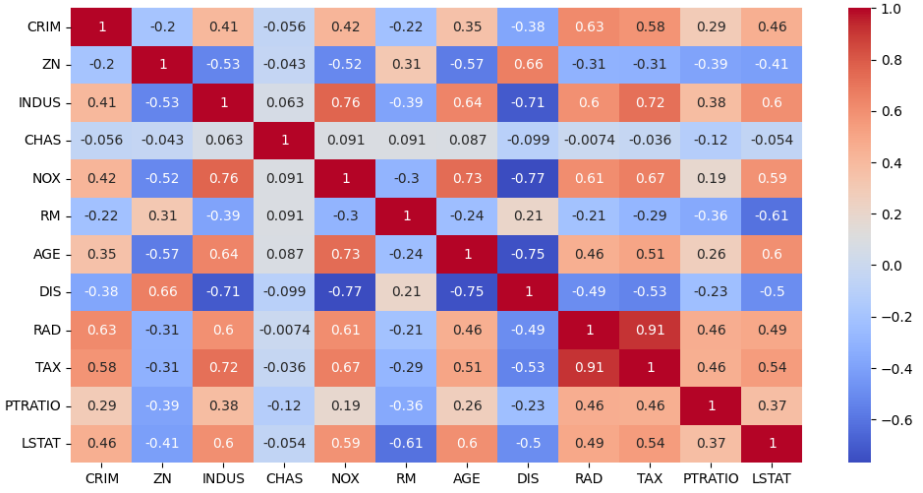


Figure A.4. Feature correlations in Boston Housing Dataset

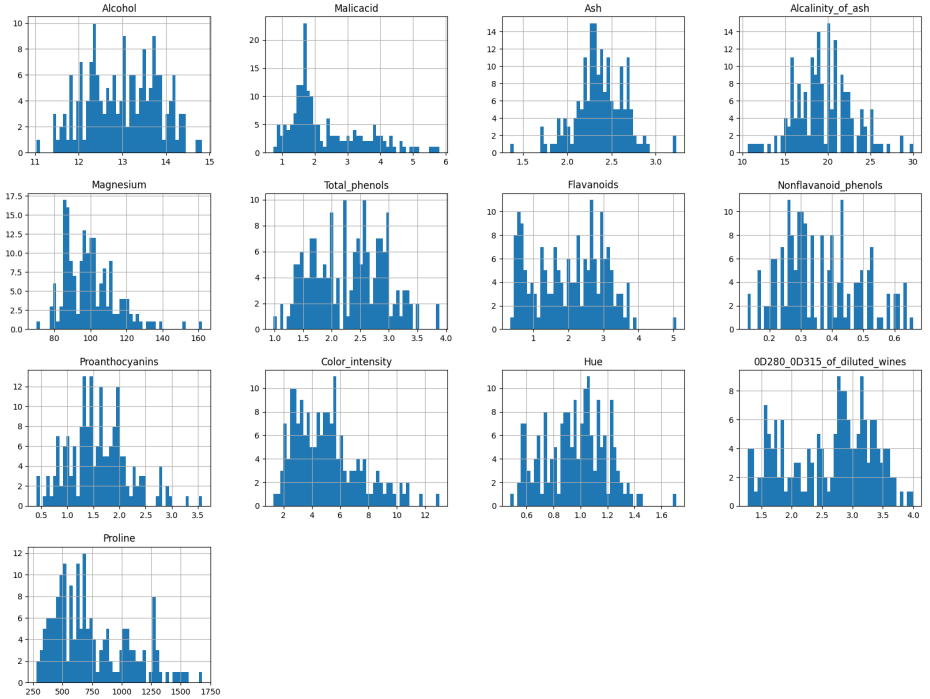


Figure A.5. Distribution of the features in the Wine Dataset.

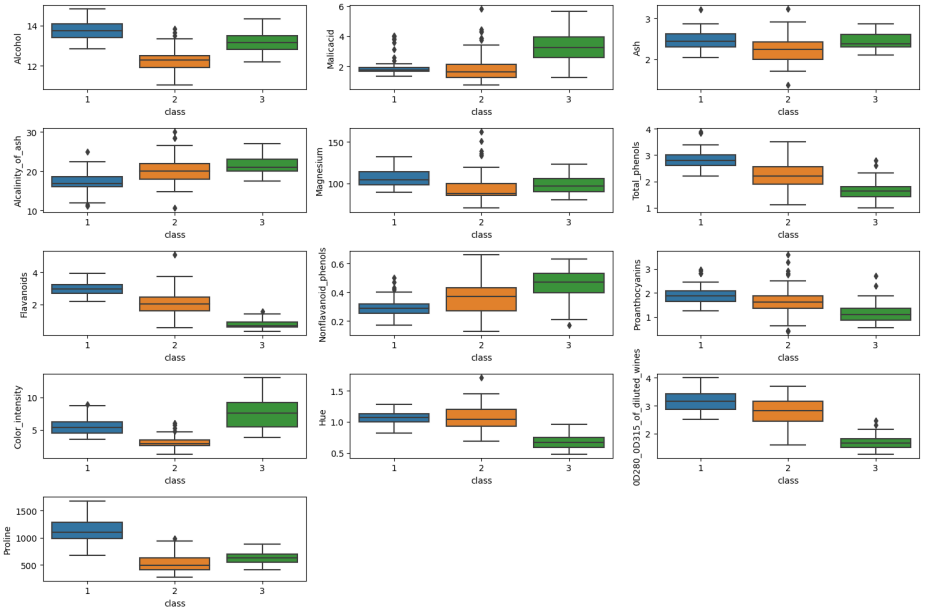


Figure A.6. Boxplots showing the distributions of each feature across target classes in Wine Dataset.

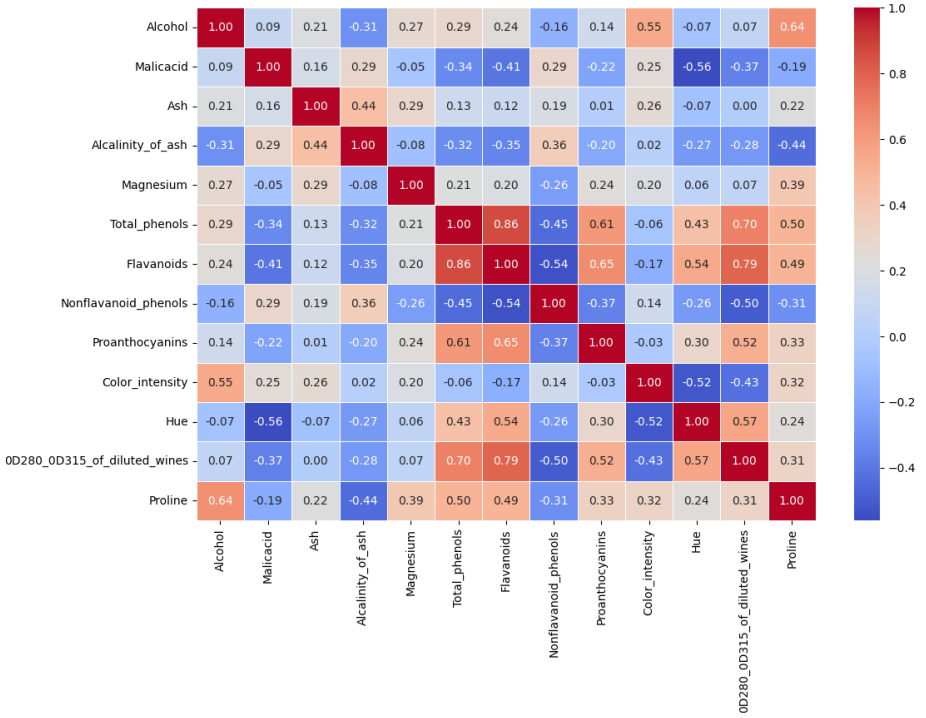


Figure A.7. Feature correlations in Wine Dataset

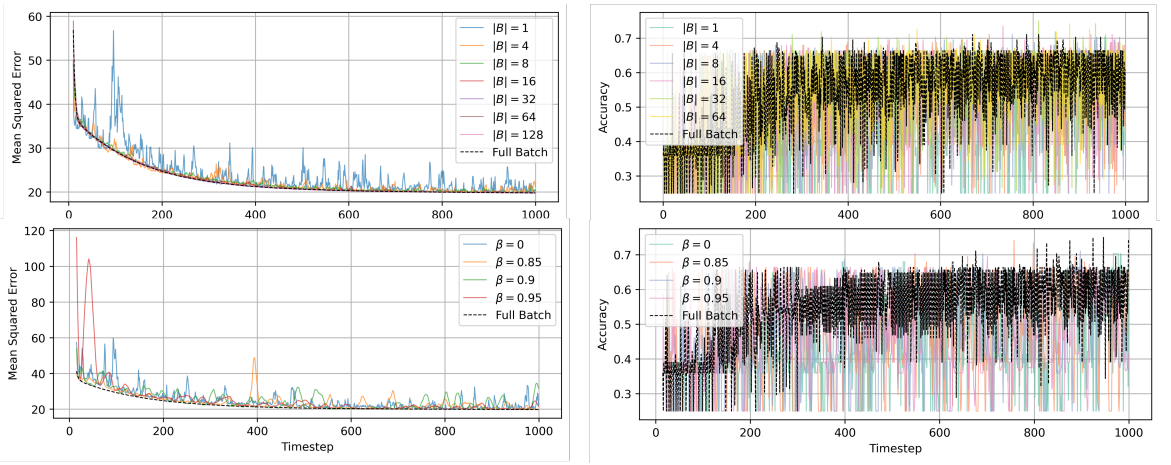


Figure B.8. Effect of batch size (top) and momentum (bottom) on linear regression (left) and logistic regression (right) models. In the top row, For linear regression, small batch sizes ($|B| = 16$) begin to approximate the full gradient very well. The momentum was tested using a mini-batch of size 1. Although momentum smooths the performance of the model during training, it does not lead to a faster rate of convergence.

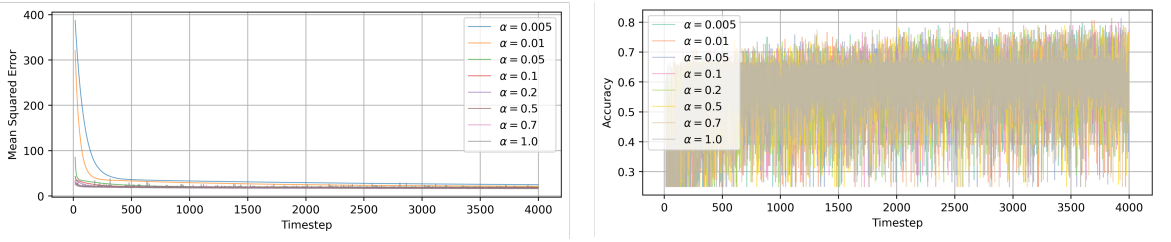


Figure B.9. Speed of convergence for different learning rates.

| IQR Factor | Batch Size | Learning Rate | Max Iters | Momentum | Mean Validation Acc | Test Acc |
|---|---|---|---|---|---|---|
| 1.0 | 8 | 0.5 | 100,000 | 0.5 | 0.948 | 0.917 |
| 2.0 | 32 | 0.5 | 100,000 | 0.9 | 0.942 | 0.914 |
| 2.0 | 32 | 0.1 | 100,000 | 0.9 | 0.942 | 0.857 |
| 2.0 | 16 | 0.1 | 40,000 | 0.9 | 0.935 | 0.771 |
| 2.0 | 16 | 0.7 | 100,000 | 0.9 | 0.935 | 0.914 |
| 2.0 | 32 | 0.5 | 100,000 | 0.5 | 0.935 | 0.857 |
| 2.0 | 32 | 0.5 | 100,000 | 0.0 | 0.935 | 0.914 |
| 2.0 | 32 | 0.7 | 100,000 | 0.9 | 0.935 | 0.886 |
| 2.0 | 32 | 0.7 | 100,000 | 0.5 | 0.930 | 0.909 |
| 2.0 | 32 | 0.5 | 100,000 | 0.9 | 0.929 | 0.970 |

TABLE B.8 *The 10 Logistic Regression models with the highest mean validation accuracy across different paramater configurations (sorted from from top to bottom)*

| Parameter | Values |
|---|---|
| IQR Factor | 1.0, 1.5, 2.0 |
| Batch Size | 1, 8, 16, 32, 64 |
| Learning Rate | 0.05, 0.1, 0.5, 0.7 |
| Max Iterations | $10^4$, $4 \times 10^4$, $10^5$ |
| Momentum | 0.0, 0.5, 0.9 |

TABLE B.9 *Parameter Grid*

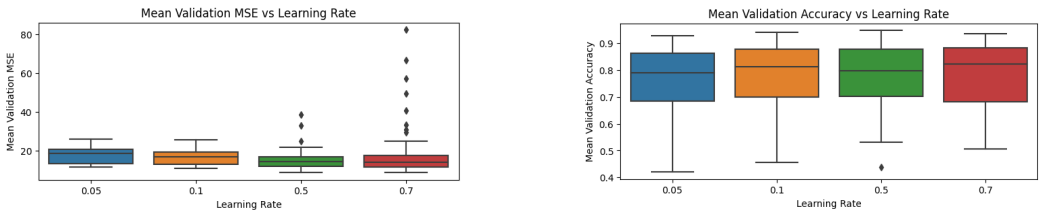


Figure B.10. Distribution of results with different learning rates across a variety of parameter configurations with Linear Regression (left) and Logistic Regression (right).

| Data | Original | Original + Gaussian | Original + Sigmoid | Original + Gaussian + Sigmoid |
|---|---|---|---|---|
| MSE Train | 6.17920 | 5.99688 | 5.93278 | 5.41358 |
| MSE Test | 8.51491 | 8.00070 | 7.51599 | 8.40242 |

TABLE B.10 *Data Augmentation with Gaussian basis and Sigmoid Transformation of the Input Features. Five Gaussian transformations were applied to the "ZN" feature and five Sigmoid basis transformations were applied to the "DIS" feature.*