COMP551 Assignment 1 Report

Joey Koay (260956108), Selina Wang (260978208), Estelle Lin (260949118)

October 5, 2023

Abstract

This research focuses on the analysis of Linear and Logistic Regression models using gradient descent, evaluated on the Boston Housing and Wine datasets. Our process involved data acquisition, preprocessing, deep data analysis, and model development from basic principles, coupled with varied experimental assessments.

For both datasets, we explored different training-test splits, and cross-validation techniques, and examined performance metrics with varying data sizes. We also studied minibatch sizes for stochastic gradient descent and optimized learning rates and parameter configurations.

Our results showed that pure analytical linear regression isn't always ideal. Enhanced methods, like mini-batch stochastic gradient descent for linear regression, and employing Gaussian basis functions yield better predictions. Additionally, binary classification in logistic regression often outperforms but is less stable than multi-class models.

This study offers insights into model performance and optimization, emphasizing the practicality of these machine-learning models in various analytical scenarios.

Keywords
Machine Learning, Logistic Regression, Linear Regression

1 Introduction

Linear Regression has been pivotal in statistics and econometrics, guiding understanding of variable relationships [1]. Its counterpart, Logistic Regression, is prominent in classification tasks, estimating categorical outcomes' probabilities [2]. These methods are precursors to advanced architectures like Neural Networks, bridging traditional statistics and modern deep learning [3].

This paper delves into logistic and linear regression's efficacy using the Boston and Wine datasets. We identify 60/40 to 80/20 train/test splits as optimal and underscore 5-fold cross-validation's reliability. Analysis indicates that smaller mini-batches yield increased precision at the cost of slower convergence. While higher learning rates speed up convergence, they also risk divergence. Gaussian basis functions enhance linear regression predictions, with mini-batch stochastic gradient descent surpassing analytical solutions.

Our binary logistic regression model with a sigmoid function displayed high accuracy, precision, recall, and f1-score. Grid-search hyperparameter tuning, based on prior research [7], achieved over 90% in performance metrics, though with noticeable variability. Conversely, the softmax-based multi-class model exhibited slightly reduced metrics but impressive stability.

In linear regression, while mini-batch stochastic gradient descent exhibited lower MSE than its analytical counterpart, predictions were more time-intensive. Employing at least 4 Gaussian basis functions to the Boston dataset further reduced the MSE.

2 Datasets

2.1 Preprocessing Dataset

In both the Boston and Wine datasets, we didn't remove any data points during outlier identification. Using the IQR (Interquartile Range) method, we found 21 outliers in the Boston dataset (as seen in Figure 9) and 3 in the Wine dataset (as seen in Figure 10). These outliers were identified based on their position and without considering data trends. Additionally, both datasets had no malformed data.

2.2 Boston Dataset

The dataset, comprising 506 samples and 13 attributes, excluding the ethically sensitive 'B' column, offers insights into factors influencing Boston housing prices. Key features include 'MEDV,' representing median home values in thousands, and 'RM,' denoting average room count per dwelling. Data preprocessing includes: 1) Loading the CSV with the Pandas library. 2) Replacing '?' with 'NA' for missing data; no rows were omitted for missing values. Descriptive statistics, including mean, median, and standard deviation, were computed using the 'describe' method.

2.3 Wine Dataset

The dataset includes chemical measurements for wines across 178 samples and 13 attributes. For pro-

cessing: 1) Pandas and other libraries were imported and dataset details were specified. 2) The dataset was loaded using Pandas with the appropriate delimiter. 3) No missing values were found.

2.3.1 Model Implementation

For the wine dataset, three models were implemented and each of these models was integrated with Stochastic Gradient Descent and Mini-batch methods later.: 1)Binary Classification Logistic Model (Figure 7). 2)Improved Binary Classification Logistic Model (with Floor/ceiling method) (Figure 8). 3) Multi-Class Softmax Model (Figure 1).

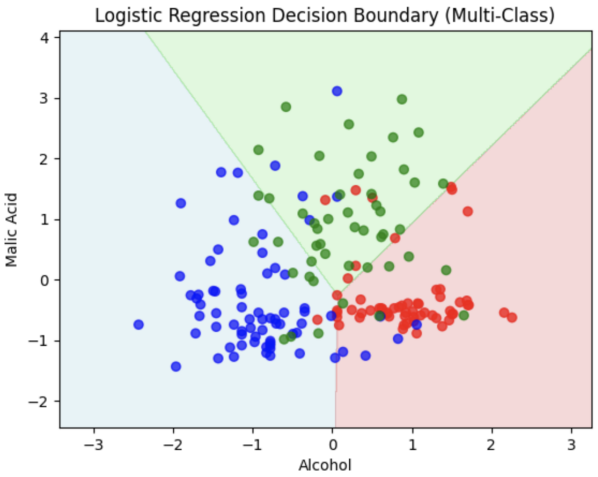

Figure 1: Multi-class Softmax Model Decision Boundary (Wine Dataset)

Our initial binary classification model exhibited limited accuracy due to the logistic function's 0-1 boundary [9]. To address this, we introduced an enhanced model, classifying values above 0.95 as 1 and those below 0.05 as 0.

For the Wine Dataset's multiple class distribution, we utilized the softmax function, tailored for more than two classes. This addressed the shortcomings of binary classification models reliant on the sigmoid function, especially given the dataset's three wine classes.

The softmax model's decision boundary for the wine data highlights three linear regions for each wine class, intersecting at a central point. Despite most data points aligning within regions, overlaps, notably in the light green region, signify potential misclassifications. Boundaries are set by normalized features in the plot.

3 Results

3.1 80/20 train/test split

In the **Boston Dataset**, we calculated the Mean Squared Error (MSE) for both training and testing data. The training data yielded an MSE of approximately 44.556, while the testing data had an MSE of roughly 39.771.

For the training data, an MSE of approximately 44.556 suggests an average error of roughly 6.708 (in thousands of dollars or $6,708) when predicting median home values. Similarly, for the testing data, the error averages about 6.325 (or $6,325) when predicting median home values. These values, which are also known as the Root Mean Squared Error (RMSE), signify the model's tendency to deviate from true median home values, with smaller RMSE values indicating better predictive performance. For further discussion about the limitations of MSE, please refer to Appendix A.

In the **Wine Dataset**, we calculated accuracy, precision, recall, and f1 score as the performance metrics for both training and testing data using the improved binary-class model and the multi-class model. Accuracy represents the percentage of correct classifications. Precision quantifies the model's ability not to label a negative instance as positive. Recall is assessing the model's capability to detect all positive instances. The F1-score measures a balance between precision and recall. The results may vary across different runs. However, both models consistently produce results around 80-90 percent. This signifies an effective class identification process with only minor errors.

The improved binary classification model used 'Alcohol' as the primary feature, with 'Class' serving as the target variable. Remarkably, this model yielded outstanding results, consistently achieving over 90 percent accuracy for the training set and occasionally achieving a perfect 100 percent accuracy for the testing set, highlighting the model's exemplary generalization.

On the other hand, the multi-class model, which utilized 'Alcohol' and 'Malic Acid' as input features and 'Class' as the target variable, demonstrated slightly lower performance metrics. It yielded approximately 80 percent accuracy for both the training and testing datasets.

3.2 5-fold cross validation

The purpose of k-fold cross-validation (where k is a positive integer) is to evaluate a machine learning model's performance and its ability to generalize effectively, all while optimizing data utilization [6]. In

this technique, the dataset is divided into k subsets (5 in our case), and in each of the k iterations, one subset serves as the 'test set' while the remaining k-1 subsets act as the 'training set'. Performance metrics are computed for each fold, allowing for statistical analysis such as calculating the mean and standard deviation of the performance data [6]. This comprehensive approach provides a robust assessment of model performance.

In the **Boston dataset**, the 5-fold performance metric for an 80/20 train/test split is displayed in Table 1.

For the **Wine Dataset**, a sample result of the 5-fold performance metric for an 80/20 train/test split is displayed in 3. This is using the binary classification model.

The model performed well across five folds, with accuracy averaging at 0.91. Notably, the second fold achieved perfect precision, and recall reached 1.00 in the third and fifth folds. The F1-score, ranging between 0.89 and 0.97, underscores the model's balance of precision and recall. This consistent performance signifies the model's avoidance of both underfitting and overfitting, affirming its reliable classification ability.

For multi-class classification using "Alcohol" and "Malic Acid" features, results are presented in 2. With softmax regression on the wine dataset, the model averaged an accuracy of 0.778, precision of 0.77, recall of 0.766, and F1-score of 0.754. Some variability in the metrics, especially lower scores in the third and fourth folds, highlights potential sensitivity to certain data subsets. While this offers a solid foundation, further optimization, possibly through feature engineering or hyperparameter adjustments, might enhance the model. The findings underscore its potential, pinpointing areas for refinement.

3.3 Training set data increment

In this section, we investigate the effect of progressively increasing the size of the training dataset from 1% to 100%.

Figures 2, 11, 14, 15, and 16 illustrate graphs using the Boston dataset. Figures 17 and 18 illustrate graphs for the binary classification in the wine dataset, while Figures 19 and 20 illustrates the multi-class graphs for the wine dataset. It's noteworthy that each code run generates different graphs due to random training set selection for both datasets.

The two datasets are producing similar trends in their performance metrics as data size increases. Notably, test performance displays significant variability initially but stabilizes as the testing size increases. Conversely, when reducing the training size, training performance exhibits pronounced fluctuations. These figures, despite having different training and testing sets, confirmed the previously mentioned trend. Now, let's delve into two specific cases:

Case 1: Large Training Set, Small Testing Set. Small testing sets lack representative samples, leading to wider error variations. Unusual or outlier test data points can disproportionately impact error amplitudes, lacking normal data points for balance.

Case 2: Small Training Set, Large Testing Set. A small training set limits the model's ability to capture data patterns effectively, leading to a less stable model. However, when combined with a large testing set, the overall error averages out, reducing the amplitude and differences in testing errors between datasets of varying sizes.

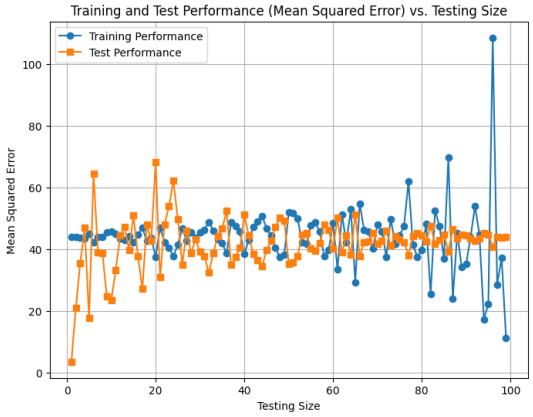

Figure 2: Impacts on mean squared error when the training size increases (Boston Dataset run 1)

3.4 Growing minibatch sizes

In this section, we will be talking about the effects of the different batch sizes on convergence speed by comparing the performance metric given at each iteration.

Beginning with linear regression on the **Boston Dataset**, Figure 21 assesses convergence speed for various mini-batch sizes. Smaller batch sizes lead to slower convergence [10], as they take smaller but more precise steps compared to larger batch sizes [11].

Additionally, we plotted the mean square error at each iteration for different batch sizes. Smaller batch sizes exhibit steeper initial slopes in Figure 22, reflecting a more rapid exponential decrease in mean square error, aligning with the notion of smaller, more precise steps.

Depending on priorities (speed or accuracy) various configurations can be selected. For models with fewer iterations, a batch size of 8 effectively reduces

MSE more rapidly. It's important to note that MSE converges to different values with different batch sizes in each run due to random test/train data selection. As a result, no definitive conclusion can be drawn regarding the batch size that consistently yields the lowest MSE overall.

A similar trend is detected for logistic regression on the **Wine Dataset**, as shown in Figure 3. The convergence speed consistently decreases as batch size increases in multi-class classification. However, in binary classification, the results fluctuate with each run. This variability may be caused by the smaller batch sizes introducing more noise into the optimization process, which can lead to faster convergence in some cases. Larger batch sizes, on the other hand, may require more epochs to converge to a similar level of performance.

Furthermore, we generated performance metric plots (including accuracy, precision, recall, and f1-score) for various batch sizes using both classification methods, as shown in Figures 23 and 24. In our experiments, it is evident that both cases exhibited their optimal performance when employing the largest batch size of 128. This finding underscores the benefits associated with using larger batch sizes.

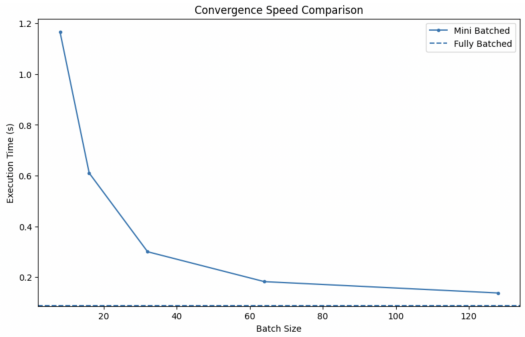

Figure 3: Convergence Speed between different batch sizes at each iteration (wine dataset multi-class)

3.5 Different learning rate

In the **Boston Dataset**, the learning rate is a critical hyperparameter that controls the convergence speed. A higher learning rate accelerates convergence but risks overshooting the minimum, potentially causing divergence and overfitting, which hampers generalization [12].

Two figures were generated for the Boston dataset: Figures 25 and 26. In the former, a learning rate of 0.01 produced the lowest MSE, and coincidentally, is the largest among the options (0.01, 0.005, 0.001). making it converge the fastest, making it the optimal choice. Conversely, in Figure 26, a learning rate of 0.05 leads to divergence and unreliable predictions.

In the **Wine Dataset** (binary class), the model achieves optimal performance at a learning rate of 0.01. Lower rates like 0.001 can lead to slower convergence, while a higher rate of 0.1 may result in overshooting or instability. Though training data at this rate indicates potential volatility (see Figure 27), the test data suggests good generalization (see Figure 28) [13].

For the multi-class scenario, training data (Figure 29) shows consistent performance across learning rates, peaking subtly at 0.001. Test data (Figure 4), however, reveals overfitting signs at higher rates. A learning rate of 0.001 strikes a balance, fitting the training data well and generalizing effectively to unseen data.

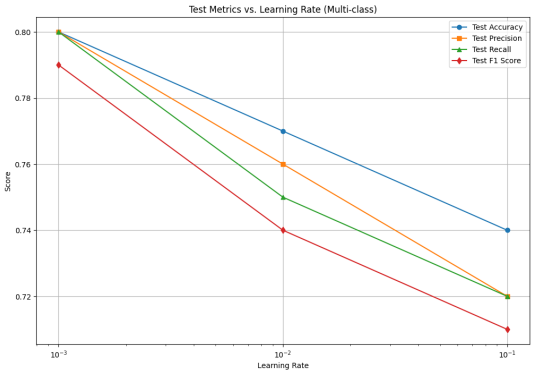

Figure 4: Testing Metrics Trend as Learning Rate Increases (Wine Dataset Multi-class)

3.6 Parameter configurations

In the **Boston Dataset**, parameter choices play a pivotal role in machine learning, directly influencing the outcomes and performance of models. In this section, we delve into the process of selecting optimal parameters for both the Boston and wine datasets.

To employ mini-batch stochastic gradient descent for linear regression, key decisions involve selecting parameters like learning rate, batch size, and maximum iterations. In Figure 22, we observe that irrespective of the mini-batch size, the model converges after 100 iterations. For optimal parameter choices, refer to Figure 30, where a batch size of 8 and a learning rate of 0.0145 yields the lowest MSE, making them the preferred selections for linear regression.

In the **Wine Dataset**, for binary classification, the F1 score was selected due to its ability to balance precision and recall, ensuring robustness against potential shifts in class distributions [14]. It offers a comprehensive evaluation, capturing both type I and II errors. Using parameters: add_bias=True, learning rate=0.01, epsilon=0.001, maximum iteration=100000.0, and batch size=32, the model

yielded training metrics of 0.91 accuracy, 0.90 precision, 0.95 recall, and 0.92 F1 score. The test metrics were 0.92 accuracy, 0.93 precision, 0.93 recall, and 0.93 F1 score.

For multi-class classification, accuracy was chosen for its straightforward interpretation, reflecting the overall proportion of correctly classified instances. In contexts like wine classification, accuracy provides a comprehensive measure across all classes. With parameters: learning rate=0.1 and number of iterations=8000, the model displayed training metrics of 0.80 accuracy, 0.79 precision, recall, and F1 score. Test metrics were 0.80 accuracy, 0.78 precision, 0.79 recall, and 0.78 F1 score.

3.7 Gaussian Basis Functions

By using Gaussian basis functions to augment the Boston dataset's features, we enable the model to create a more accurate non-linear prediction curve. Figure 5 illustrates that the MSE with Gaussian basis functions is 34.824, significantly lower than the MSE without them (43.601). Varying the number of Gaussian basis functions enriches the data, with more functions leading to smaller MSE and improved predictions, as shown in Figure 31.

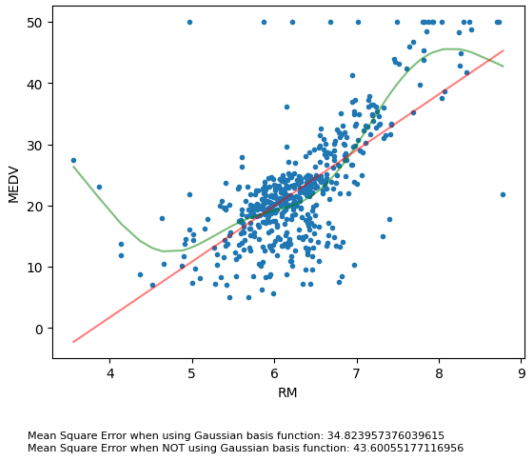

Figure 5: Regular vs Gaussian basis function enhanced dataset for Linear Regression (Boston Dataset)

3.8 Analytical vs. mini-batch stochastic gradient descent (linear regression)

When comparing analytical linear regression to mini-batch stochastic gradient descent (SGD), SGD yields a significantly lower MSE, as shown in Figure 32. This improvement is attributed to SGD's incorporation of regularization techniques, iterative optimization, and its suitability for large datasets and complex models with many parameters. It allows for fine-tuning hyperparameters like learning rate, batch size, and optimization algorithms to better adapt to the data.

4 Discussion

Our project assessed a linear regression model on the Boston dataset using MSE, recognizing its limitations and exploring alternative error metrics. In the Wine dataset, we conducted binary and multi-class classification, noting improvements with softmax (increased model stability). We studied the effects of training set size, batch size, and learning rate on performance and convergence, and found that there needs to be a balance between the amount of training set and testing set, and the optimal split is a 60/40 to 80/20 training/testing dataset split. For the linear regression model, employing Gaussian basis functions enhances the accuracy, as it does not only predict using a straight line. Finally, we favored mini-batch stochastic gradient descent over analytical linear regression for its adaptability and parameter tuning capabilities.

Conclusions

In this paper, we emphasized the significance of metric selection, classification methods, and optimization techniques in machine learning. Future investigations should explore advanced regression techniques beyond linear and logistic regression to enhance predictive accuracy. Detecting and handling outliers is crucial for improving predictions. For optimizing parameter configurations, especially in the Boston dataset, consider creating a 3D graph to pinpoint the best parameter values more effectively, as opposed to relying on a 2D table and individual epoch comparison. For the wine dataset, consider adding more features to the multi-class classifying feature sets, and creating a multi-dimensional graph could be the future improvements for the multi-class model.

Statement of Contributions

Joey Koay led the work on the Boston dataset, while Selina Wang and Estelle Lin primarily tackled the Wine dataset; all three collaborated on the write-up and experiments, ensuring quality in their respective responsibilities.

Appendices

Appendix A

In Figure 6, the blue dots represent the 506 data points in our Boston dataset, while the green line illustrates the best-fit linear regression model obtained through an analytical approach. MSE quantifies the average squared difference between predicted and actual values, serving as an indicator of model performance. A lower MSE implies a better model fit, with smaller squared errors between predictions and true values.

It's worth noting why Mean Squared Error (MSE) is employed in regression. MSE is designed to give more weight to larger errors, ensuring that predictions significantly off from the true value have a substantial impact on the overall error measure [4]. The act of squaring the errors serves the crucial purpose of ensuring that all errors are positive, eliminating the possibility of positive and negative errors offsetting each other when summed [4]. This characteristic makes MSE a natural and effective choice for evaluating regression models, as it prioritizes the magnitude of errors.

However, the utilization of Mean Squared Error (MSE) introduces certain drawbacks. It exhibits sensitivity to outliers, where even a single outlier can significantly skew the final result [5]. Furthermore, MSE provides no information about the direction of errors, as it consistently outputs positive values [5]. Moreover, optimizing a model solely for MSE can be problematic, particularly with complex models, as it may lead to overfitting [5]. Such models can fit the training data exceptionally well but struggle to generalize effectively to unseen data. To address these limitations and gain a more comprehensive understanding of model performance, we will explore alternative error measurement methods in subsequent sections of this paper.

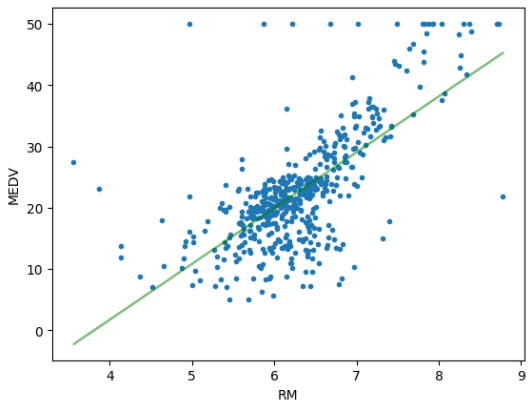

Figure 6: Analytical Linear Regression Graph (Boston Dataset)

Appendix B

To understand Table 1 better, it is best to understand what each of the error indicate and mean. MAE (Mean Absolute Error) is used for a robust measure of central error tendency [8]. An MAE of 4.5, for the Boston dataset, implies that our model's predictions are, on average, about $4,500 off in terms of median home values. MSE (Mean Squared Error) and RMSE (Root Mean Squared Error) were described in Appendix 3.1. MSE quantifies squared errors, while RMSE provides an interpretable error measure in the same units as the target variable [8]. R^2 (R-squared) indicates how well the model explains variance in median home values. An R^2 of 0.47 suggests that 47% of the variance is predictable from our model. A higher R^2 signifies a better fit to the data [8]. MAPE (Mean Absolute Percentage Error) represents the average percentage difference between predicted and actual values [8]. For instance, if our model predicts $300,000 for a home, the average difference is 26%, ranging from $222,000 to $378,000. While additional metrics like adjusted R-squared and median absolute error exist, these 5 metrics provide a concise overview of our model's accuracy.

5-fold method Errors					
Fold number	MAE	MSE	RMSE	R^2	MAPE
1	4.553	47.477	6.890	0.461	25.951
2	4.397	41.874	6.471	0.448	23.364
3	4.199	35.374	5.948	0.459	25.240
4	4.360	41.583	6.448	0.587	27.019
5	4.971	55.977	7.482	0.389	28.552
total	4.496	44.457	6.648	0.469	26.025

Table 1: Specific error values for each fold and across all folds for Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared (R^2) and Mean Absolute Percentage Error (MAPE) (Boston Dataset)

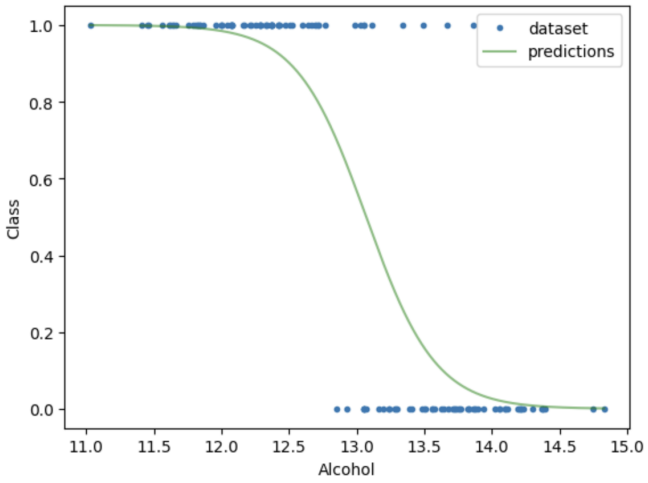

Figure 7: Binary Classification Logistic Regression Model Prediction (Wine Dataset)

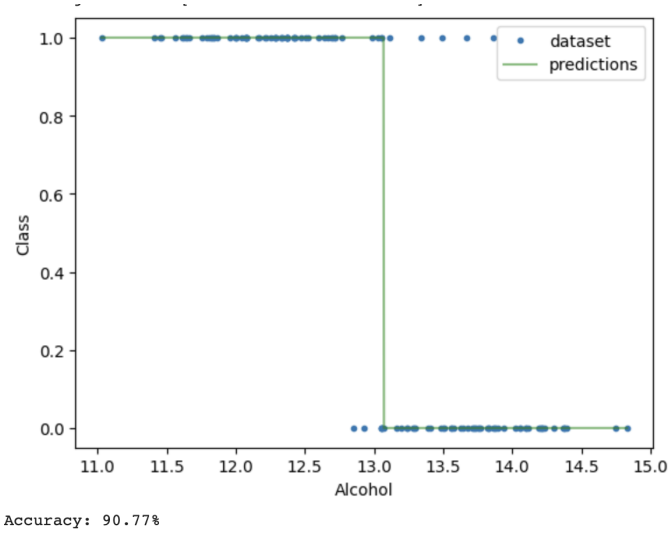

Figure 8: Binary Classification Logistic Regression Model (improved) (Wine Dataset)

5-fold method Errors				
Fold number	ACC	PRE	Recall	F1-score
1	0.80	0.81	0.81	0.80
2	0.86	0.85	0.82	0.83
3	0.74	0.69	0.71	0.69
4	0.69	0.70	0.71	0.68
5	0.80	0.80	0.78	0.77
total	0.78	0.77	0.77	0.76

Table 2: Specific error values for each fold and across all folds for accuracy (ACC), PRE (Precision) (Wine Dataset)

5-fold method Errors				
Fold number	ACC	PRE	Recall	F1-score
1	0.92	0.93	0.93	0.93
2	0.96	1.00	0.94	0.97
3	0.92	0.87	1.00	0.93
4	0.88	0.92	0.86	0.89
5	0.88	0.80	1.00	0.89
total	0.91	0.90	0.95	0.92

Table 3: Specific error values for each fold and across all folds for accuracy (ACC), PRE (Precision) (Wine Dataset)

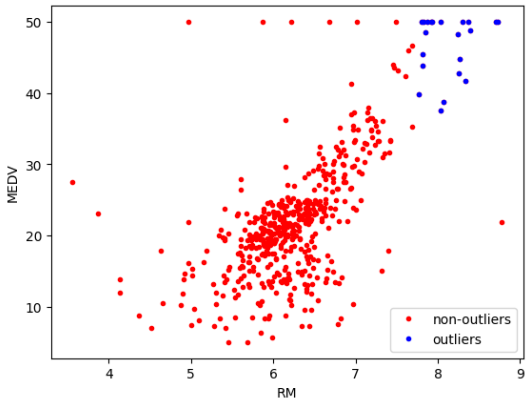

Figure 9: Outliers identified by IQR method (Boston Dataset)

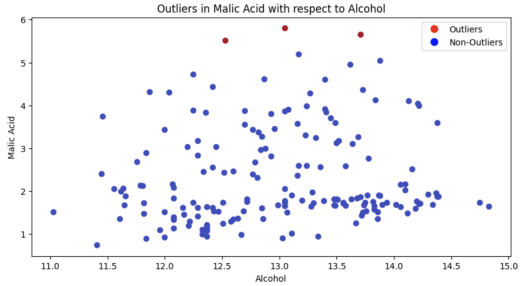

Figure 10: Outliers identified by IQR method (Wine Dataset)

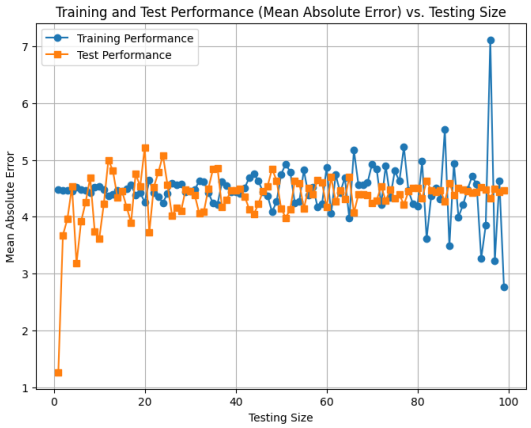

Figure 11: Impacts on mean absolute error when the training size increases (Boston Dataset run 1)

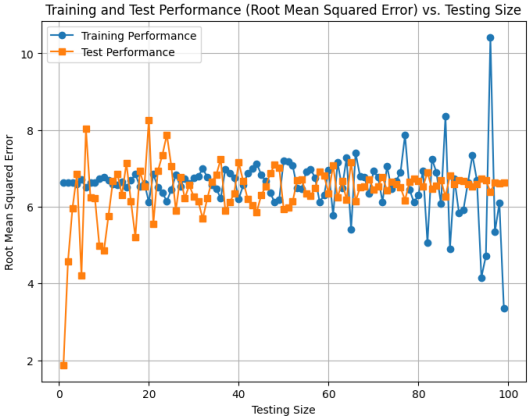

Figure 14: Impacts on root mean squared error when the training size increases (Boston Dataset run 1)

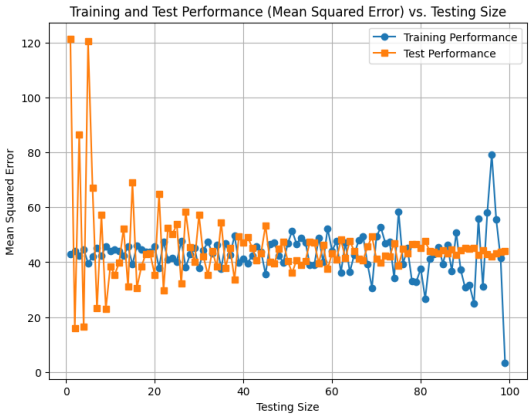

Figure 12: Impacts on mean squared error when the training size increases (Boston Dataset run 2)

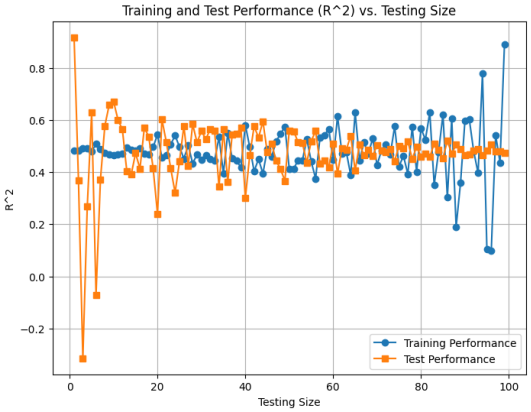

Figure 15: Impacts on R^2 when the training size increases (Boston Dataset run 1)

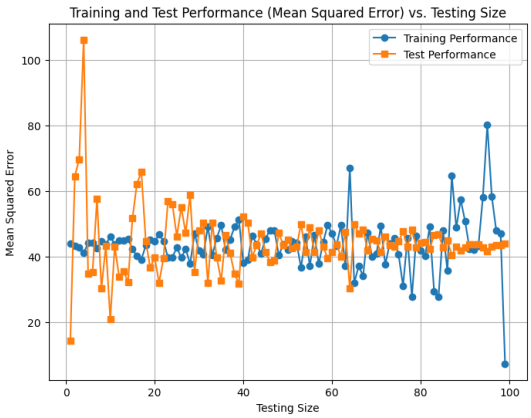

Figure 13: Impacts on mean squared error when the training size increases (Boston Dataset run 3)

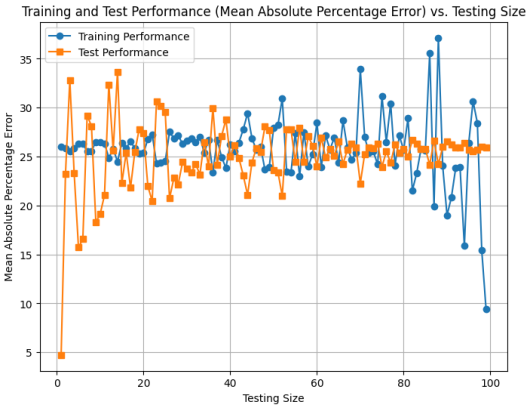

Figure 16: Impacts on mean absolute percentage error when the training size increases (Boston Dataset run 1)

Figure 17: Impacts on training performance when the data size increases (wine dataset binary-class)

Figure 18: Impacts on test performance when the data size increases (wine dataset binary-class)

Figure 19: Impacts on training performance when the data size increases (wine dataset multi-class)

Figure 20: Impacts on testing performance when the data size increases (wine dataset multi-class)

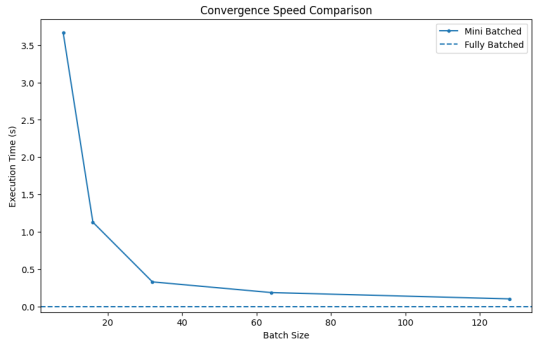

Figure 21: Convergence Speed with different batch sizes (Boston Dataset)

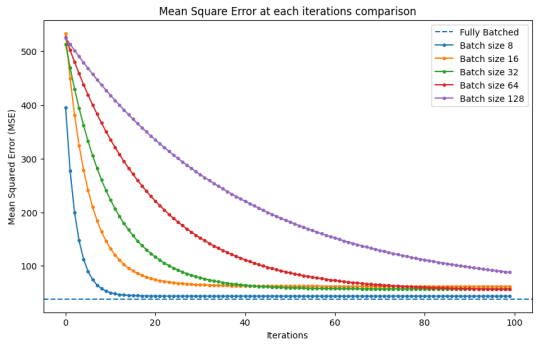

Figure 22: Convergence Speed between different batch sizes with respect to the performance metric at each iteration (Boston Dataset)

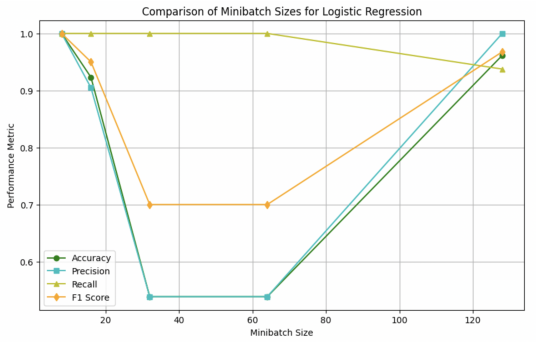

Figure 23: Performance metric between different batch sizes at each iteration (wine dataset binary-class)

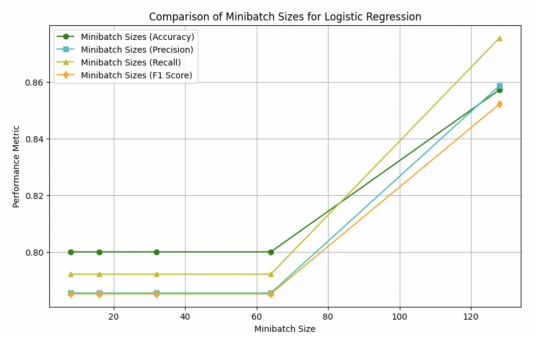

Figure 24: Performance metric between different batch sizes at each iteration (wine dataset multi-class)

Figure 25: Learning rate comparison (Boston Dataset)

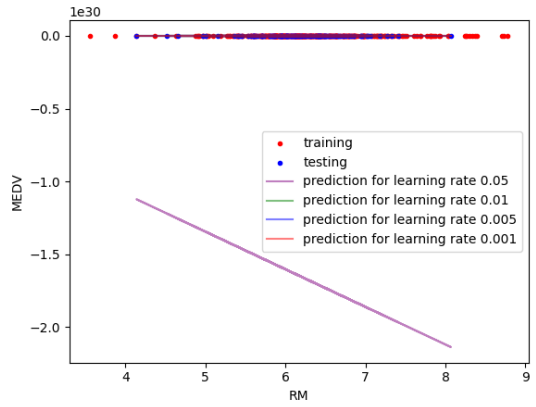

Figure 26: Learning rate comparison (Boston Dataset)

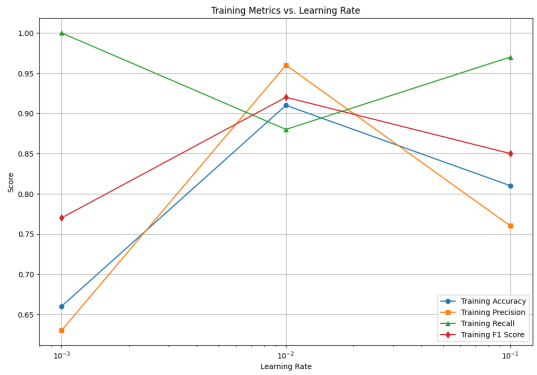

Figure 27: Training Metrics Trend as Learning Rate Increases (Wine Dataset)

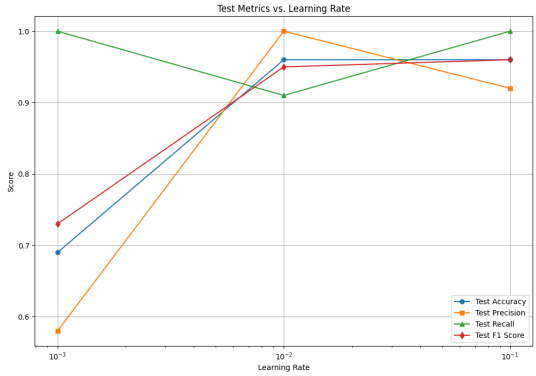

Figure 28: Testing Metrics Trend as Learning Rate Increases (Wine Dataset)

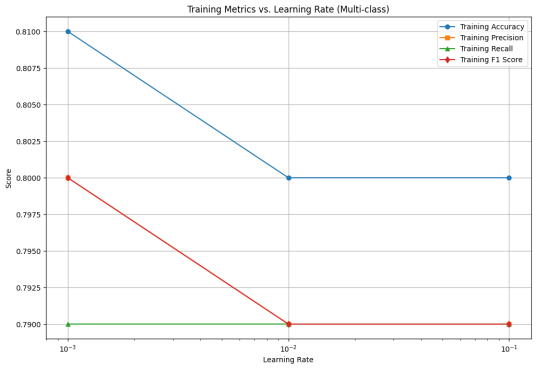

Figure 29: Training Metrics Trend as Learning Rate Increases (Wine Dataset Multi-class)

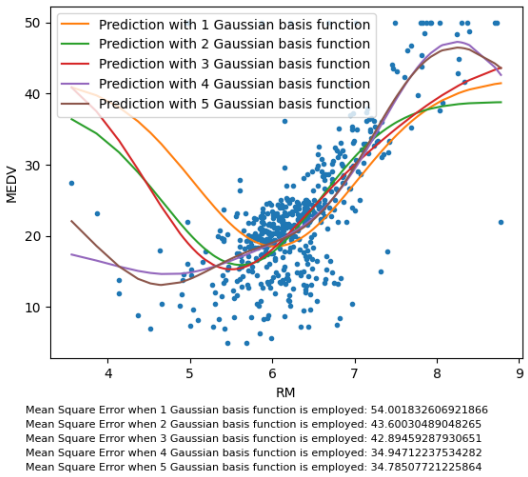

Figure 31: Effects of different amounts of Gaussian basis function enhanced dataset (Boston Dataset)

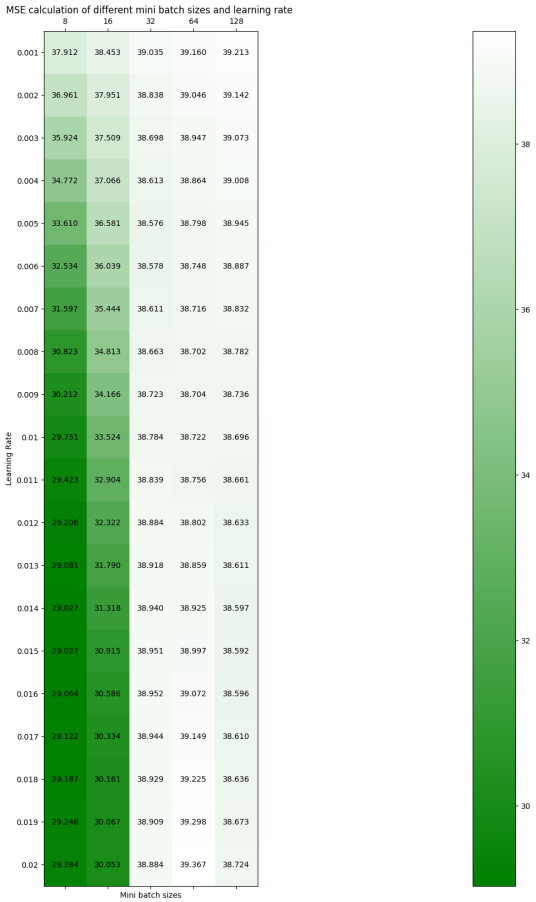

Figure 30: MSE calculation of different mini-batch sizes and learning rates (Boston Dataset)

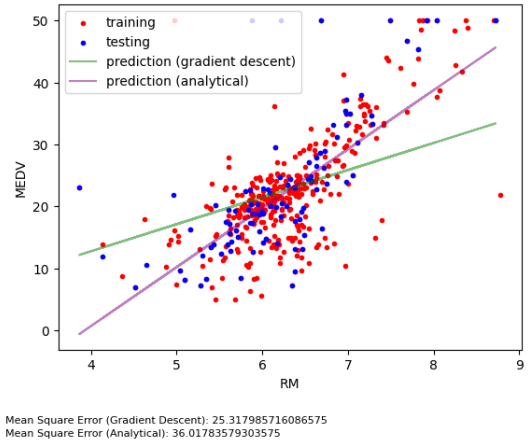

Figure 32: Analytical vs. SGD linear regression (Boston Dataset)

References

[1] Draper, N. R., & Smith, H. (1966). Applied regression analysis. John Wiley & Sons.

[2] Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression. John Wiley & Sons.

[3] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[4] Frost, J. (2023). Mean squared error (MSE). Retrieved from https://statisticsbyjim.com/regression/mean-squared-error-mse/

[5] (2023). Understanding mean squared error (MSE) - a key metric in data analysis! Retrieved from https://databasecamp.de/en/statistics/mean-squared-error

[6] Brownlee, J. (2023). A gentle introduction to k-fold cross-validation. Retrieved from https://machinelearningmastery.com/k-fold-cross-validation/

[7] Rohan, K. (2021). Wine Quality Prediction Model Using Machine Learning Techniques. https://www.diva-portal.org/smash/get/diva2:1574730/FULLTEXT01.pdf

[8] (2022). Interpretation of evaluation metrics for regression analysis (MAE, MSE, RMSE, MAPE, R-squared, and... Retrieved from https://medium.com/@ooemma83/interpretation-of-evaluation-metrics-for-regression-analysis-mae-mse-rmse-mape-r-squared-and-5693b61a9833

[9] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

[10] Patrikar, S. (2019). Batch, Mini Batch & stochastic gradient descent. Retrieved from https://towardsdatascience.com/batch-mini-batch-stochastic-gradient-descent-7a62ecba642a

[11] (2023). Epochs, Batch Size, Iterations - How are They Important to Training AI and Deep Learning Models. Retrieved from https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations

[12] Brownlee, J. (2020). Understand the Impact of Learning Rate on Neural Network Performance. Retrieved from https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/

[13] Brownlee, J. (2020). Understand the dynamics of learning rate on deep learning neural networks. Machine Learning Mastery. https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/

[14] Serokell. (2023). A guide to F1 score. Retrieved from https://serokell.io/blog/a-guide-to-f1-score