COMP 551 ASSIGNMENT 1 REPORT - FALL 2024

Hathaway Hao
261071268

Yifan Lin
261078741

Michael Yu
261070826

October 1, 2024

ABSTRACT

Statistical learning is key to Machine Learning, so we evaluated linear regression and logistic classification on the Infrared Thermography and CDC Diabetes datasets. We implemented analytical least squares, gradient descent, and mini-batch SGD. Our experiments analyzed performance, feature importance, and the effects of training size, mini-batch size, and learning rates. Cross-validation was used to prevent overfitting, and we compared analytical and iterative approaches for linear regression. Key findings show the importance of learning rates, batch size, and computational trade-offs. This study marks our first application of machine learning and identifies areas for future research.

1 Introduction

This study applies machine learning techniques to two datasets: the Infrared Thermography Temperature dataset (regression) and the CDC Diabetes Health Indicators dataset (classification). We implemented and compared linear and logistic regression, exploring the effects of hyperparameters and training strategies. This provided a solid foundation for developing more advanced models.

The project focused on three key algorithms: (1) analytic linear least squares, (2) gradient descent for logistic classification, and (3) mini-batch gradient descent. We evaluated model performance, analyzed feature importance, and examined the impact of training data size, mini-batch size, and learning rates on model efficiency. Cross-validation was used to prevent overfitting, and we compared analytical and iterative approaches for linear regression.

The two datasets used are cited in the references. Both datasets have been thoroughly exploited, and numerous works that use the datasets already exist, and can be found directly from the link in the references 1 2.

2 Datasets

We tested the linear regression model on a dataset called *Infrared Thermography Temperature*1 and the logistic classification model on a dataset called *CDC Diabetes Health Indicators*2.

2.1 Dataset 1: Infrared Thermography Temperature

The *Infrared Thermography Temperature* dataset originally sourced from the UCI Machine Learning Repository, consists of multiple infrared thermography readings. It has a sample size of 99 and includes 6 attributes along with the target response 'aveOralM', which represents the average oral temperature. Our goal is to use linear regression to predict 'aveOralM' based on these attributes. This dataset is used in a regression context to model the relationship between the infrared thermography readings and oral temperature.

2.2 Dataset 2: CDC Diabetes Health Indicators dataset

The *CDC Diabetes Health Indicators* dataset, also sourced from the UCI Machine Learning Repository, consists of 253,680 data points. Each data point is characterized by 21 attributes, and the task is to predict whether an individual is diabetic (1 for diabetic, 0 for non-diabetic). Our goal is to use logistic regression to predict the presence of diabetes based on the individual's health indicators.

2.3 Data Processing

Both datasets were preprocessed with a structured pipeline to ensure compatibility with the machine-learning models. The preprocessing steps are detailed below:

- **Data Preparation and Exploration for both Datasets**
  Initially, data were loaded from the UCI repository. Basic data exploration was performed, which revealed the presence of some missing values in the dataset. For the *Infrared Thermography Temperature* dataset, we conducted further exploration of each feature column, which led us to apply appropriate preprocessing techniques. Categorical variables such as 'Age', 'Gender', and 'Ethnicity' were **one-hot encoded** to convert them into numerical form. Boolean variables were also transformed into integers (0 or 1) to facilitate model training.
- **Handling Missing Values**
  After analyzing the dataset, we identified rows with missing values. These rows were removed to ensure the integrity of the data, leaving us with a clean dataset without null entries.
- **Feature-Target Exploration**
  Since the target variable for the *Infrared Thermography Temperature* dataset ('aveOralM') is continuous, we focused on exploring the relationship between the features and the target through visualizations such as scatter plots to understand how each feature impacts the target temperature.
- **Data Splitting and Normalization**
  Before training, both datasets were split into training and testing sets using an 80-20 split. After splitting, we applied standardization (scaling the data) to the feature sets using the `StandardScaler` to ensure that the features are on a similar scale. This helps the model converge faster during training. For the *Infrared Thermography Temperature* dataset, the target variable 'aveOralM' was separated from the features after ensuring that all missing values were handled appropriately.

3 Results

As stated in the introduction section, our study encompasses a series of experiments. To test our models, we conducted a series of experiments that are fundamental in machine learning research and practice. We evaluated performance metrics (MSE for regression, accuracy for classification) to establish baseline model effectiveness 3. We tested the impact of training data size to assess model scalability and data efficiency, as well as mini-batch size experiments, which helped optimize the takeaways between computational efficiency and convergence speed 4. Cross-validation was employed to obtain robust performance estimates and avoid overfitting 5. Lastly, comparing analytical and iterative solutions provided insights into the strengths and limitations of different optimization approaches.

3.1 Experiment 1: Report performance of linear regression for data1 and logistic regression for data2

In this experiment, we evaluated the performance of the linear regression model using the Mean Squared Error (MSE), R-squared score, and Mean Absolute Error (MAE) for both the training and test sets.

Set	MSE	R^2	MAE
Training	0.0617	0.7756	0.1950
Test	0.0667	0.6646	0.2032

Table 1: MSE, R^2, and MAE for Training and Test Sets of the Linear Regression Model

Set	Accuracy	Precision	Recall	F-1 score
Training	0.8627	0.5241	0.1683	0.2547
Test	0.8637	0.5332	0.1692	0.2569

Table 2: Accuracy, Precision, Recall, and F-1 score for Training and Test Sets of the Logistic Regression Model

From the results shown in Table 1 and Table 2, we can draw the following conclusions:

The Training MSE is 0.0617, while the Test MSE is 0.0667. The small difference between these values indicates that the model generalizes well, with minimal overfitting. The R-squared values showing that the model explains 77.56% of the variance in the training data and 66.46% in the test data. Although the test set R-squared is slightly lower, this is expected and suggests a slight reduction in model performance on unseen data. The MAE values for the training and test sets are also very close, shows that the average error in the model's predictions is consistent across both sets. Overall, the linear regression model demonstrates reasonable performance on both the training and test sets, with a slight decrease in accuracy on the test set, which suggests potential for further optimization or regularization.

The **Logistic Regression model** achieved a training accuracy of 0.8627 and test accuracy of 0.8637. Precision was moderate, at 0.5241 for training and 0.5332 for testing, while recall was low at around 0.17 for both sets, resulting in F1-scores of 0.2547 for training and 0.2569 for testing. Regularization did not improve performance, suggesting the need for more advanced techniques like handling class imbalance or feature engineering to boost recall and predictive performance.

3.2 Experiment 2: Linear Regression vs. SGD Linear Regression

In this experiment, we investigated the weights of the top 10 features in both traditional Linear Regression and Stochastic Gradient Descent (SGD) Linear Regression models. By examining feature weights, we gain insights into which variables most strongly influence the model's predictions, potentially uncovering non-obvious relationships in the data. Comparing top features from both approaches allows us to assess the consistency of feature importance across optimization methods, providing insights into the stability of our rankings. Additionally, this analysis can help detect potential biases in the dataset or model, prompting further investigation when unexpected features show large weights. The data evaluation is shown in Table 3 and Table 4:

Feature	Weight	Absolute Weight
canthi4Max1	0.357622	0.357622
canthiMax1	-0.352782	0.352782
T_Max1	0.301124	0.301124
T_LC1	0.300552	0.300552
T_RC_Max1	0.230246	0.230246
T_OR1	0.212044	0.212044
Max1R13_1	-0.209065	0.209065
T_RC1	-0.159100	0.159100
T_OR_Max1	-0.157372	0.157372
T_RC_Dry1	0.143177	0.143177

Table 3: Top 10 Features by Absolute Weights in Linear Regression

Feature	Weight	Absolute Weight
T_FHLC1	-0.199022	0.199022
T_RC_Wet1	0.195939	0.195939
RCC1	-0.188584	0.188584
Max1L13_1	-0.163420	0.163420
T_Max1	0.142402	0.142402
aveAllL13_1	0.108773	0.108773
T_FHBC1	0.106191	0.106191
T_atm	-0.099799	0.099799
T_RC_Dry1	0.085703	0.085703
canthiMax1	0.078252	0.078252

Table 4: Top 10 Features by Absolute Weights in SGD Linear Regression

3.3 Experiment 3: Influence of Training Data Size on Linear Regression Loss

Experiment 3 investigates the influence of training data size on model performance for both linear and logistic regression. This analysis is crucial for understanding how much data is needed for effective model training and how model performance scales with increased data availability. By varying the proportion of data used for training from 20% to 80%, we can observe the learning curve of each model and identify potential overfitting or underfitting scenarios. From the results shown in Table 6 7 and Figure 12, we can draw the following key conclusion:

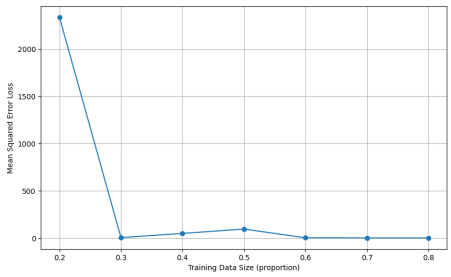

Figure 1: Impact of Training Data Size on Linear Regression Model Performance

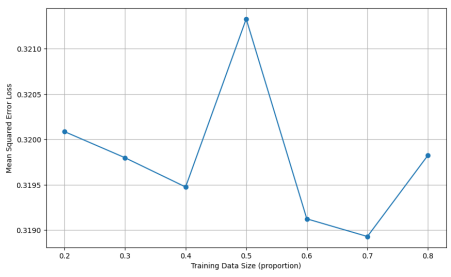

Figure 2: Impact of Training Data Size on Logistic Regression Model Performance

As the **training data size increases**, the **loss decreases**. More specifically:

For the Linear Regression model, performance is highly volatile at smaller training sizes (20-50%). Then A significant improvement occurs at 60% training size, with loss decreasing to 7.1. Finally, The model achieves its best performance at 70-80% training size, with losses of 0.065 and 0.0635 respectively. For the Logistic Regression model, performance

is remarkably consistent across all training sizes, with losses ranging narrowly from 0.3189 to 0.3213. There's a slight trend of decreasing loss as training size increases, but the difference is minimal.

3.4 Experiment 4: Mini-Batch Size Impact on Convergence and Performance

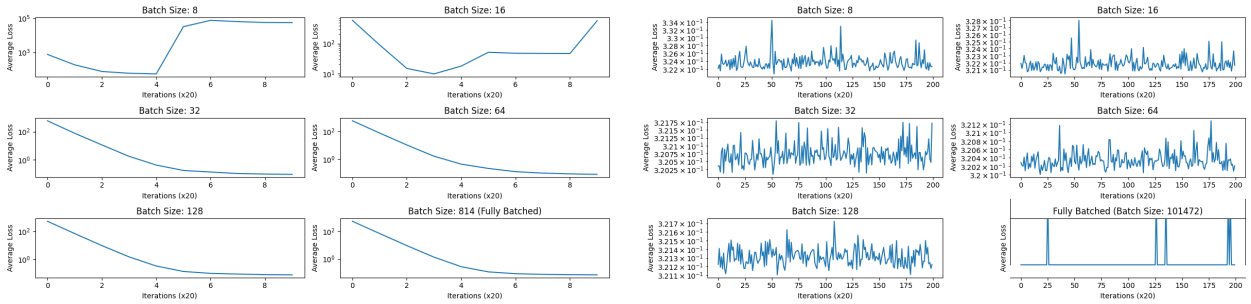

Figure 3: Influence of Mini-batch Size on our Linear Regression Loss (smoothed) with Mini-Batch Stochastic Gradient Descent

In Figure 3, these line graphs show that larger mini-batch sizes generally lead to better performance in SGD linear regression. As batch size increases from 8 to full batch, we see more stable convergence and lower final loss values. Compared to a fully batched approach, the larger mini-batches seem to approach similar performance levels. While larger batches converge more slowly initially, they ultimately provide better stability and final results. Among the tested configurations, full batch size 814 works best, offering optimal performance and stability.

3.5 Experiment 5: Try different learning rates

In this experiment, we investigated the impact of different learning rates on the performance of a Stochastic Gradient Descent (SGD) Linear Regression model. We tested five different learning rates: 1e-1, 1e-2, 1e-3, 1e-4, and 1e-5, over 1500 iterations with a batch size of 32. The experiment data visualization is shown below 5:

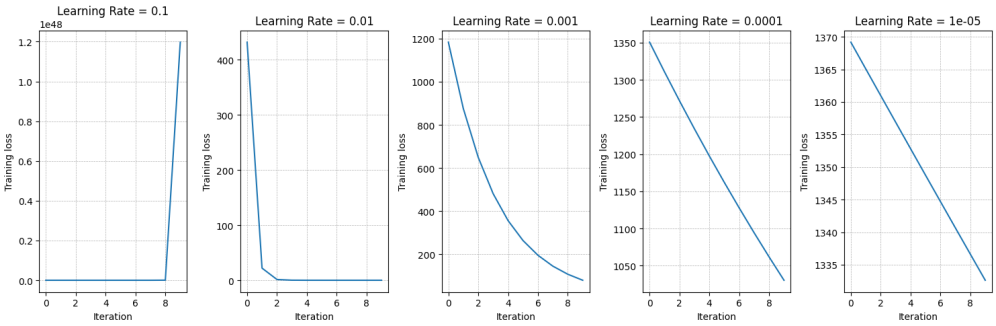

Figure 4: Smoothed Loss Function of Linear Regression with Mini-Batch Stochastic Gradient Descent with Learning Rate: 1e-1, 1e-2, 1e-3, 1e-4, 1e-5

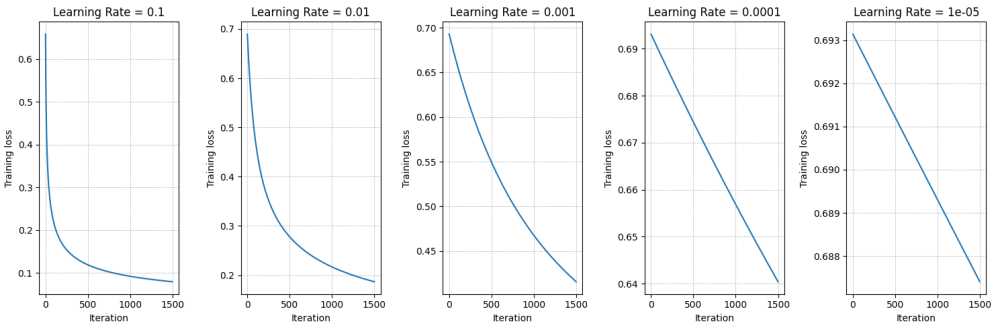

Figure 5: Smoothed Loss Function of Logistic Regression with Learning Rate: 1e-1, 1e-2, 1e-3, 1e-4, 1e-5

For this problem and dataset, a learning rate of 1e-2 appears to be optimal among the tested values. It provides the best balance between convergence speed and stability. Learning rates that are too high cause (1e-1) the model to diverge, while rates that are too low (1e-4 & 1e-5) cause the model to converge too slowly.

In this experiment, we evaluated the effect of different learning rates on the loss in a logistic regression model. The learning rate significantly impacts how fast or slow the model converges, as seen in the results across various rates. Overall, the learning rate of 0.1 was optimal for this experiment, leading to the lowest loss and fastest convergence, while smaller learning rates hindered the model's performance and slowed its learning.

3.6 Experiment 6: Perform cross-validation using different learning rates

In this experiment, we evaluated the performance of both our linear regression and logistic regression models using optimal parameters. The performance metrics for the linear regression model, as shown in Table 5a, include the Mean Squared Error (MSE) for both training and test sets. The model demonstrates an MSE of 0.264627 on the training set and 0.387184 on the test set, indicating that the model generalizes well without significant overfitting.

Set	MSE
Training Set	0.264627
Test Set	0.387184

(a) **Table 4** 5-Fold Cross-Validation in linear

Set	**Accuracy**	**Precision**	**Recall**	**F1-score**
Training	0.862781	0.862067	0.864027	0.863005
Test	0.863076	0.863464	0.863181	0.863128

(b) **Table 5** 5-Fold Cross-Validation in logistic

Additionally, Table 5b presents the accuracy, precision, recall, and F1-score for the logistic regression model on both the training and test sets. The model achieves an accuracy of 0.862781 on the training set and 0.863076 on the test set, with similar high values for precision, recall, and F1-score. These results suggest that the logistic regression model performs consistently across both datasets, confirming its robustness and generalization ability.

3.7 Experiment 7: Compare analytical solution with mini-batch SGD

Here in figure 6, the analytical solution provides a constant loss value, serving as a baseline for optimal performance. In contrast, the SGD approach shows an evolution of performance over iterations. Initially, the SGD loss is significantly higher than the analytical solution, reflecting the random initialization of weights. As the number of iterations increases, the SGD loss rapidly decreases and then begins to oscillate around the analytical solution's loss value. This oscillation is characteristic of SGD, resulting from the stochastic mini-batch sampling and the fixed learning rate.

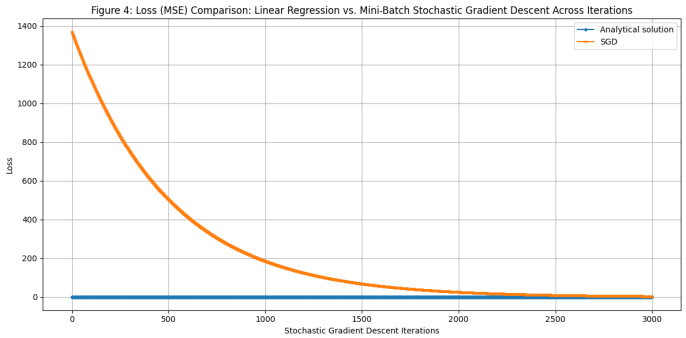

Figure 6: The comparison between linear regression and Mini-batch SGD

4 Discussion and Conclusion

Here are our takeaways from all the experiments we have made:

- **Experiment 1**
  The linear regression model generalized well, while logistic regression had high accuracy but struggled with precision-recall balance. Regularization was ineffective, and further improvements should focus on class

imbalance and feature engineering. Both models performed adequately but need refinement. Limitation: Performance may degrade on larger datasets due to computational constraints.

- **Experiment 2**
  We compared feature importance between Linear and SGD Linear Regression. The top features were consistently ranked, though some biases in weights suggest further investigation is needed.Limitation: Results may vary with larger datasets or different SGD hyperparameters.
- **Experiment 3**
  We examined how training size affects performance. Linear regression was volatile with small data but improved with more. Logistic regression remained stable across all sizes, suggesting it handles small datasets better. Limitation: Larger datasets may reveal different behaviors or performance issues not observed with smaller samples.
- **Experiment 4**
  Using a full batch size of 814 provided the best balance between stability and performance, resulting in the lowest loss. Smaller batches had more volatile convergence, while larger batches converged more slowly but more consistently. The limitation is that only a few batch sizes were tested; further tuning and testing with different datasets could offer additional insights.
- **Experiment 5**
  A learning rate of 0.1 provided the best balance between speed and stability in this experiment, leading to the lowest loss and fastest convergence. Higher rates caused divergence, while lower rates led to slow convergence. The limitation is that we only tested a few rates; further tuning and additional iterations could provide deeper insights.
- **Experiment 6**
  We assessed the performance of linear and logistic regression with optimal parameters. Linear regression generalized well, showing minimal overfitting, while both models benefited from improved stability and reduced errors. Logistic regression demonstrated consistent accuracy across datasets, highlighting the value of parameter optimization. Limitation: Further tuning may be needed for larger or more complex datasets.
- **Experiment 7**
  We compared analytical and SGD solutions for linear regression. SGD initially showed higher loss but converged towards the analytical solution over iterations. Limitation: The number of iterations was preset and longer runs might show different long-term behavior.

Future investigations could involve delving into deep neural networks to leverage their capability to capture complex patterns and perform comparisons with our current models. Additionally, probing into parameter tuning for these deep learning models, including factors such as learning rate, training data size, and mini-batch size, holds promise for enhancing their robustness on the datasets.

5 Statement of Contributions

All members contributed equally to this project. We each completed the lab independently and then combined our individual work and code to produce the final report.

References

[1] Infrared Thermography Temperature Wang et al., 2021 https://api.semanticscholar.org/CorpusID:245585208

[2] CDC Diabetes Health Indicators. 2021. https://archive.ics.uci.edu/dataset/891/cdc+diabetes+health+indicators

[3] Hastie, Trevor, et al. The elements of statistical learning: data mining, inference, and prediction. Vol. 2. New York: springer, 2009.

[4] Prémont-Schwarz, I. (2023). Gradient Descent. COMP 551, McGill University.

[5] James, Gareth. "An introduction to statistical learning." (2013)

Appendix A Default Model Parameters

Model	Max iteration	Batch Size	Learning Rate	Epsilon	Regularization Term
Linear Regression with SGD	1500	16	1e-2	1e-5	0
Logistic Regression	2000	16	1e-3	1e-7	0.1

Appendix B Experiment 3

B.1 Training Loss History of Linear Regression with data size

Training Size (Proportion)	**MSE Loss**
0.2	233.8436
0.3	100.6199
0.4	162.3182
0.5	11.3937
0.6	7.1098
0.7	0.0650
0.8	0.0635

Table 6: Linear in Different Training Set Sizes

Training Size (Proportion)	**MSE Loss**
0.2	0.3201
0.3	0.3198
0.4	0.3195
0.5	0.3213
0.6	0.3191
0.7	0.3189
0.8	0.3198

Table 7: Logistic in Different Training Set Sizes

B.2 Final Loss, Mean Loss and Min Loss in Linear Regression model with different mini-batch SGD

Batch Size	**Final Loss**	**Mean Loss**	**Min Loss**
8	55391.013203	28565.609963	52.265444
16	2085.653474	149.216125	3.461185
32	0.101154	67.321424	0.086003
64	0.083188	65.046168	0.080201
128	0.068877	64.398326	0.068812
814	0.068739	63.514423	0.068739

Table 8: Summary of Mini-batch Size Influence on SGD Linear Regression Loss