Getting Starting with Machine Learning

Abstract

In this assignment, we implemented two fundamental machine learning models—linear regression and logistic regression—on two benchmark datasets: the Infrared Thermography Temperature dataset and the CDC Diabetes Health Indicators dataset. Our experiments compared full-batch and mini-batch gradient descent, explored varying batch sizes and learning rates, and analyzed the impact of training set sizes on performance. Results showed that larger training sizes improved model accuracy. Mini-batch gradient descent accelerated convergence while maintaining performance similar to full-batch gradient descent. Linear regression benefited from larger batch sizes and higher learning rates, while logistic regression performed better with smaller batch sizes and lower learning rates. The analytical solution for linear regression produced comparable results to gradient descent.

Introduction

As mentioned above, this assignment focuses on the implementation and performance analysis of two machine learning models — linear regression for predicting continuous outcomes and logistic regression, more suitable for binary classification tasks. We used linear regression onto the Infrared Thermography Temperature dataset to predict the average oral temperature, and logistic on the CDC Diabetes Health Indicators dataset to predict diabetes. The objective is to understand how these models behave when trained using gradient descent techniques and to evaluate their performance through a series of experiments, providing insights into the impact of training set size, batch size, and learning rate on model performance. For linear regression, we also compared the analytical solution with gradient descent methods. Our findings revealed that an 80% training data size was optimal for both models. Mini-batch gradient descent significantly accelerated convergence without sacrificing accuracy. For linear regression, larger batch sizes and a higher learning rate of led to better results, while logistic regression favored smaller batch sizes and lower learning rates. Furthermore, the analytical solution for linear regression closely matched the results from gradient descent, highlighting the effectiveness of both methods.

Dataset

The Infrared Thermography Temperature dataset used for linear regression consists of 33 features such as gender, age, ethnicity and other temperature readings from thermal images. The goal is to predict oral temperature measured in monitor mode using linear regression. To prepare the data, we handled missing continuous values through interpolation and removed rows with missing categorical values. A notable overlap was found in the Age feature, where the categories 21-25, 26-30, and 21-30 conflicted, leading to the removal of 10 rows containing the ambiguous 21-30 range seeing as how keeping it as a separate category did not affect the performance of our model. We then applied label encoding over one-hot encoding to the three categorical features—Age, Ethnicity, and Gender—for cleaner, easier to interpret and marginally better results, before dropping the irrelevant SubjectID column.

The Diabetes Health Indicators Dataset comprises healthcare statistics and lifestyle survey data, including individuals' diabetes diagnoses. It features 35 attributes, which include demographic information, lab test results, and survey responses from each patient. The target variable for classification is the patient's health status, indicating whether they have diabetes/are pre-diabetic, or are healthy. Preprocessing involved separating the target variable from the features, and minimal additional cleaning was required.

These datasets contain sensitive health information, such as diabetes status and lifestyle habits. While anonymized, the possibility of re-identification via linkage with other data sources (e.g., geographic or demographic information) poses potential privacy risks.

Results

Experiment 1

Using an 80-20 train/test split, the linear regression model's performance on the training data is generally higher and more consistent, with an average R^2 score of 0.76313 over 50 runs, compared to an average R^2 score of 0.72113 on the testing data. The same trend can be observed for the average F1 score for logistic regression. This outcome is expected, as the model is specifically trained and optimized on the training set.

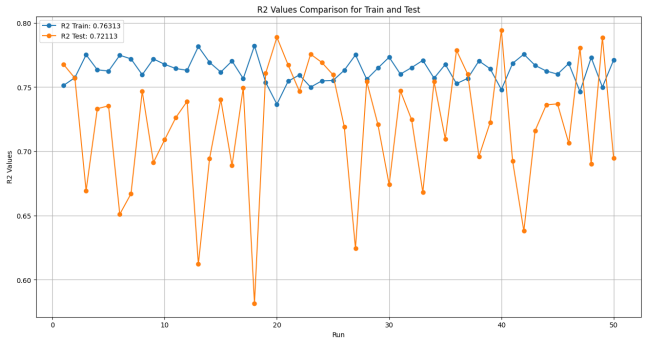

Figure 1: Learning Curve for Linear Regression

Figure 2: Learning Curve for Logistic Regression

Experiment 2

The weight of each feature can be seen in the graphs below.

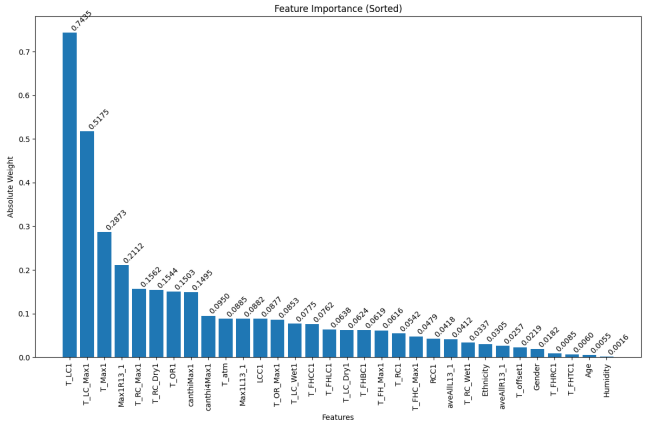

Figure 3: Feature Weights for Linear Regression

The feature importance analysis for predicting oral temperature shows that T_LC1 is the most influential feature with a weight of 0.7485. T_LC_Max1 and T_FC_Max1 also play significant roles, with weights of 0.5175 and 0.2873,

respectively. This indicates that specific thermal readings are crucial for accurate predictions, while features like Humidity and Age have minimal impact.

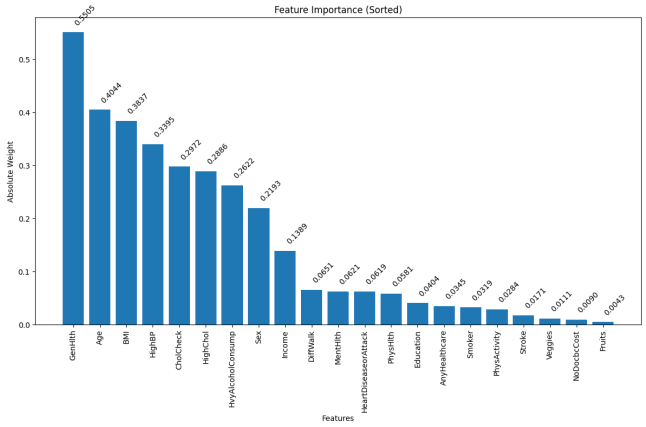

Figure 4: Feature Weights for Logistic Regression

The feature importance analysis for the Diabetes Health Indicators Dataset shows that GenHlth is the most influential feature with a weight of 0.5505, indicating its strong impact on predicting diabetes. Other significant features include Age and BMI, with weights of 0.4044 and 0.3837, respectively, highlighting the role of general health and demographic factors. In contrast, features like Fruits and NoDocbcCost have minimal impact, suggesting they contribute little to diabetes prediction. This underscores the importance of health status and age in the dataset.

Experiment 3

We evaluated the performance of both machine learning models based on different training and testing data sets and observed how size impacted the results. We progressively increased the size of the training data in increments of 10% from 20% to 80% and measured the performance of the models at each step for both the training set and test set. With smaller training subsets, the model is likely to overfit the training data, leading to high performance on the training set but poor generalization on the test set. As the training set grows, the model starts to perform better on the test set, but slightly decreases on the training set as it no longer overfits it. As the training set approaches 80%, both performances converge indicating that the model has enough data to learn from and can generalize as well.

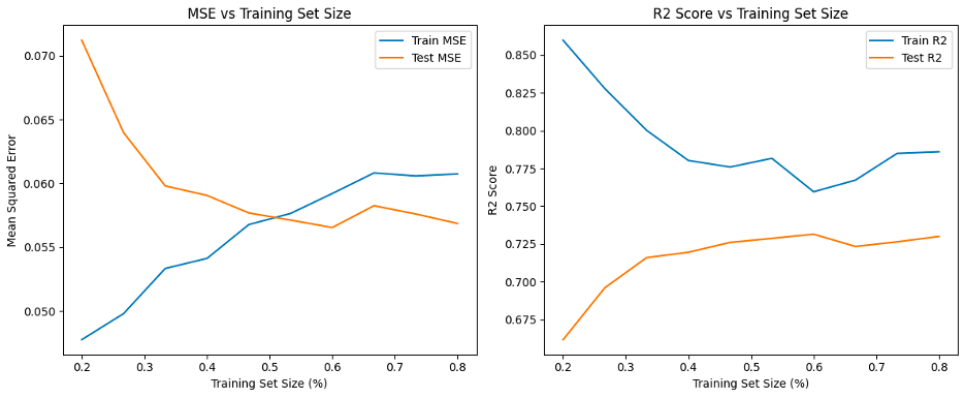

Figure 5: Linear Regression Performance as Training Set Size Increases

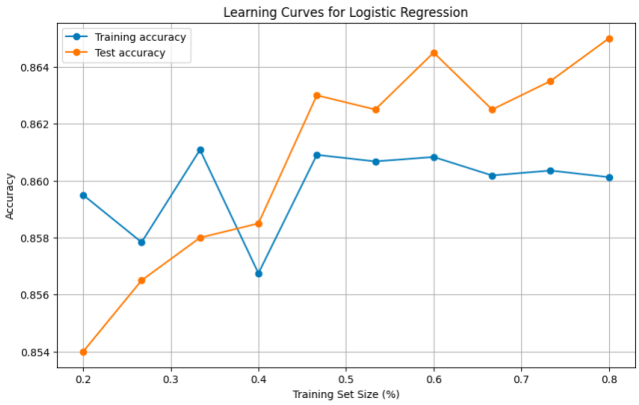

Figure 6: Logistic Regression Performance as Training Size Increases

Experiment 4

We tested both linear regression and logistic regression using SGD with growing mini-batch sizes. For each batch size, we evaluated the model's performance using multiple metrics including mean squared error (MSE), root mean squared error (RMSE), mean absolute error and R^2 score. Below is a summary of the results for different batch sizes.

Table 1: Linear Regression Results

Batch Size	MSE	RMSE	MAE	R Squared
8	0.0740	0.2716	0.2096	0.7442
16	0.0723	0.2684	0.2075	0.7510
32	0.0725	0.2688	0.2067	0.7508
64	0.0708	0.2656	0.2062	0.7558
128	0.0705	0.2651	0.2063	0.7570
256	0.0710	0.2660	0.2063	0.7554

For linear regression, as the batch size increases, there is a slight improvement in the performance of the model. However, the R^2 score closest to the full batch model of 0.7486 is the closest is obtained when we take samples of size 16. The gradient never completely converges to 0, as it always reaches the maximum number of iterations, but the R^2 score seems to indicate that the method performs well.

Table 2: Logistic Regression Results

Batch Size	Accuracy	Precision	Recall	F1 Score
8	0.8321	0.4175	0.4106	0.4114
16	0.8476	0.4206	0.3040	0.3520
32	0.8539	0.4825	0.2025	0.2840
64	0.8595	0.5226	0.1394	0.2163
128	0.8654	0.5454	0.1227	0.1997
256	0.8608	0.5535	0.1578	0.2449

For logistic regression, smaller batch sizes provide a better balance between precision and recall, yielding the highest F1 score. This could be because of several reasons. For instance, smaller batch sizes introduce more noise in the gradient descent estimates which can help the model avoid local minima. As batch size increases, this beneficial noise is lost, leading to lower F1 scores. Larger batches can also cause the model to overfit the training data, leading to faster convergence, but potentially at the expense of generalization. However, the larger batch sizes results in an F1 score closer to the full batch R^2 score of 0.2487.

Experiment 5

To assess the impact of varying learning rates on model performance, we tested 4 different leaning rates: 0.0001, 0.0005, 0.01 and 0.05. Low learning rates (0.0001 and 0.0005) for linear regression resulted in overall higher errors and lower R^2 scores. The learning rates 0.0001, 0.0005, 0.01 and 0.05 resulted in R^2 values of

0.6467, 0.6560, 0.6707 and 0.6708 respectively. In contrast, those learning rates performed better than the higher ones in logistic regression, yielding F1 scores of 0.4363 for a learning rate of 0.0001 and 0.3788 for a learning rate of 0.0005 versus 0.1791 and 0.1900 for learning rates of 0.05 and 0.1 respectively. This is to be expected since a lower rate is more adaptable and is able to get more precision.

Experiment 6

When comparing the analytical solution for linear regression with the mini-batch stochastic gradient descent approach, we observe similar performance, with the analytical method achieving an R^2 value of 0.75942, closely matching the best R^2 value of 0.7554 obtained in Experiment 4. This demonstrates that while both methods are equally effective in terms of accuracy, the gradient descent approach is more time-efficient.

Originality/Creativity

As previously mentioned in experiment 4, our gradient failed to converge before reaching the maximum number of iterations. To address this issue, we explored the use of momentum in the gradient descent implementation to accelerate the convergence in flatter areas and potentially avoid local minima. Momentum also smooths out oscillations by averaging the gradient over time, allowing the algorithm to move more consistently towards the global minimum. However, we found that implementing momentum slightly decreased the R^2 score from 0.7324 to 0.7224, likely due to overshooting the optimal solution.

Discussion and Conclusion

Through this assignment, we gained hands-on experience with implementing linear and logistic regression models from scratch and training them using gradient descent optimization techniques. We analyzed the effect of various parameters on the models' performance. Our results showed that the optimal training data size was 80%. Mini-batch gradient descent proved to be a faster alternative to standard gradient descent while delivering comparable results. For linear regression, larger batch sizes improved performance although the smaller batch sizes yielded results closer to the fully batched model, whereas logistic regression performed better with smaller batch sizes. Higher learning rates were more effective for linear regression, while lower learning rates worked best for logistic regression. Additionally, the analytical solution for linear regression closely matched the results obtained through gradient descent. Lastly, we explored the benefits and drawbacks of using momentum in gradient descent, which though accelerated convergence, negatively affected the accuracy of prediction.

Future work could involve addressing the imbalance in the CDC dataset as we suspect the reason our F1 scores for the logistic regression model was consistently low is because the target feature was imbalanced, leading to an over representation of healthy subjects, especially for SGD. This could be resolved through techniques such as class weighting or oversampling to accommodate skewed datasets could improve the model's generalization.

Statement of Contributions

William: logistic regression implementation
Guillaume: linear regression implementation
Jessica: assignment write-up