Comparative Analysis of Regression Models and Approaches for Temperature Prediction and Diabetes Classification

261250317 Léo Valette
261091403 Kayvan Dharsee
261111350 Kohei Okabayashi

Abstract

In this assignment, we analyzed the performance of linear and logistic regression models on two datasets. We used linear regression on the Infrared Thermography Temperature dataset to predict average oral temperature (aveOralM) and logistic regression on the CDC Diabetes Health Indicators dataset to classify diabetes status (Diabetes_binary). We identified various key features impacting both models performances. For predicting oral temperature, the "Ethnicity" feature was notably important. For diabetes classification, blood pressure, income, and age were crucial factors. To improve model performance, we balanced the CDC dataset via downsampling, removed duplicates, and fine-tuned learning rates and batch sizes. Our analysis highlights the importance of effective preprocessing, thorough experimentation, and tuning techniques to enhance model accuracy across both datasets.

Introduction

Machine learning models like linear and logistic regression are essential for predictive and classification tasks. In this assignment, we implemented these models to analyze the Infrared Thermography Temperature dataset Wang et al. (2023) and the CDC Diabetes Health Indicators dataset.

Thermographic imaging and oral temperature readings help detect elevated body temperatures (EBT) in clinical settings Wang et al. (2021). The Infrared Thermography dataset includes physiological and environmental data such as ambient temperature (T_atm), humidity, and facial temperature. Using linear regression, we predicted oral temperatures (aveOralM) and found that "Ethnicity," especially being "White," was the most significant predictor.

The CDC Diabetes dataset highlights health factors like high blood pressure, BMI, and physical activity. Our logistic regression model classified diabetes status (Diabetes_binary), with "HighBP," "Income," and "Age" being key predictors, showing both direct and inverse correlations.

We applied data preprocessing methods like downsampling and duplicate removal to enhance performance and optimized learning rates and batch sizes through running various experiments with a validation set to improve convergence. This analysis emphasizes the role of feature selection, preprocessing, and tuning for accurate predictions.

Datasets

When working with the Infrared Thermography Temperature and CDC Diabetes Health Indicators datasets, ethical considerations are crucial, particularly regarding privacy and data security due to the sensitive health information involved. To protect participant privacy, personal identifiers like "subjectID" must be carefully anonymized. Post training, we also found some concerning results with the features with high weights in the linear regression model. For this model, ethnicity was a large factor in determining oral temperature. With the dataset only consisting of 1020 points, and the ethnicity feature not being balanced, this clearly raises concerns in the ethics and true accuracy of our predictive model outside the data. In addition, we could not find any resources online to explain this racial bias.

The Infrared Thermography dataset contains 1020 samples with 32 continuous, 4 categorical variables, 2 target variables, and 1 ID variable. There were no duplicate rows, and two missing values in the "Distance" column were removed to maintain data quality, as their small number made removal the most effective option. An outlier in "Distance," 100 times larger than the mean, was corrected from 79 to 0.79, as it was likely a decimal error. We reached said conclusion as no other features had any outliers, implying that all other entries in the outlier data

point were not outliers.

Categorical variables such as "Gender," "Age," and "Ethnicity" were encoded to numerical/binary values. "Age" was transformed into numerical values representing the mean of each range, and one-hot encoding was applied to "Gender" and "Ethnicity," with the first category dropped to prevent collinearity. One-hot encoding was chosen to avoid creating an ordinal relationship between categories, which label or binary encoding might introduce.

Initial exploration showed a mostly balanced gender distribution, with slightly more females, and a predominantly "White" ethnicity. The correlation matrix revealed strong correlations among many continuous features, suggesting redundancy and potential for feature reduction. Most continuous variables followed a bell-curve-like distribution, except "Humidity," which showed high variability. Scatter plots of features most correlated with oral temperature ("aveOralM") confirmed their predictive value. The target variable was centered around 37°C, resembling a normal distribution. This informed our feature preprocessing and modeling strategies.

The CDC Diabetes Health Indicators dataset includes 15 integer and 8 binary variables, one target variable, 21 feature variables, and an ID variable. Initially, we removed 24,206 duplicate rows (nearly 10% of the data), improving the balance of the target variable, with diabetes cases increasing from 13.9% to 15.3%. Since there were no categorical features, no encoding was needed. Feature correlations were low, with none exceeding 0.55 in absolute value. However, the dataset is imbalanced, with fewer entries representing individuals with diabetes or prediabetes.

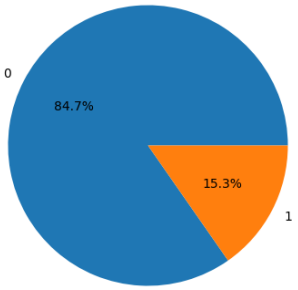


Figure 1: Distribution of the Target After Removal of Duplicate Rows

The most correlated features with the target variable were "General Health," "High Blood Pressure," "Difficulty Walking," "BMI," and "High Cholesterol." These variables showed significant differences between the positive and negative classes as shown in our box plots, making them key predictors for diabetes. While most integer and binary features were balanced, exceptions included "Physical Health," "Mental Health," "Stroke," and "Cholesterol Check." Despite a slight outlier in BMI, we retained it to observe its impact on model accuracy.

After cleaning the datasets, we split the feature set into an 80-20 train/test split, with the option to create a validation set. The features were then standardized for clear weight comparison across models.

Results

Our first experiment analyzed the performance of analytical linear regression and fully-batched logistic regression using an 80-20 train/test split. As shown in our metrics table, the linear regression model performed well, while the logistic regression model's performance was subpar. For both models, the metrics across the training and testing sets were very similar, indicating the models were likely not overfitting to the training data. This also suggests that the models are able to generalize well to unseen data.

For the linear regression model, predictions on average differed from the true values by around 0.2°C, as indicated by the MAE. Given that the target variable averages around 37°C and is tightly clustered, the model demonstrated good predictive accuracy, as shown in our histogram. The RMSE was similar in magnitude to the MAE, suggesting no large errors from outliers. Additionally, the R^2 score of approximately 0.77 on the testing set confirmed strong model fit and no overfitting.

For the logistic regression model, the accuracy score of around 0.85 initially seemed high, but the class imbalance (85% of the data in the negative class) meant a model predicting all negatives would achieve similar accuracy. The imbalance caused far more false negatives than false positives, as shown by the lower recall compared to precision. The F1 score of 0.22 provided a more reliable performance measure than accuracy, revealing the model's struggle to predict diabetes due to the imbalance. The confusion matrix supported this, showing very few true positives.

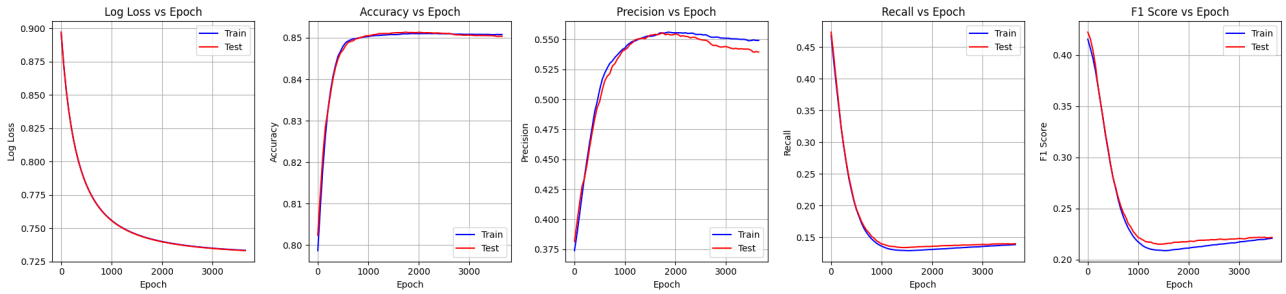


Figure 2: Training vs Testing Performance Metrics for Diabetes Classification Before Downsampling

To improve performance, we addressed dataset imbalance by experimenting with upsampling and downsampling. Downsampling significantly improved the F1 score to 0.72 on the testing set, though accuracy decreased. We visualized the overall performance with metric graphs and a confusion matrix. We acknowledged that accuracy wasn't a reliable metric for the imbalanced dataset and shouldn't be directly compared to the model trained on balanced data. The success of downsampling was likely due to the large amount of available data.

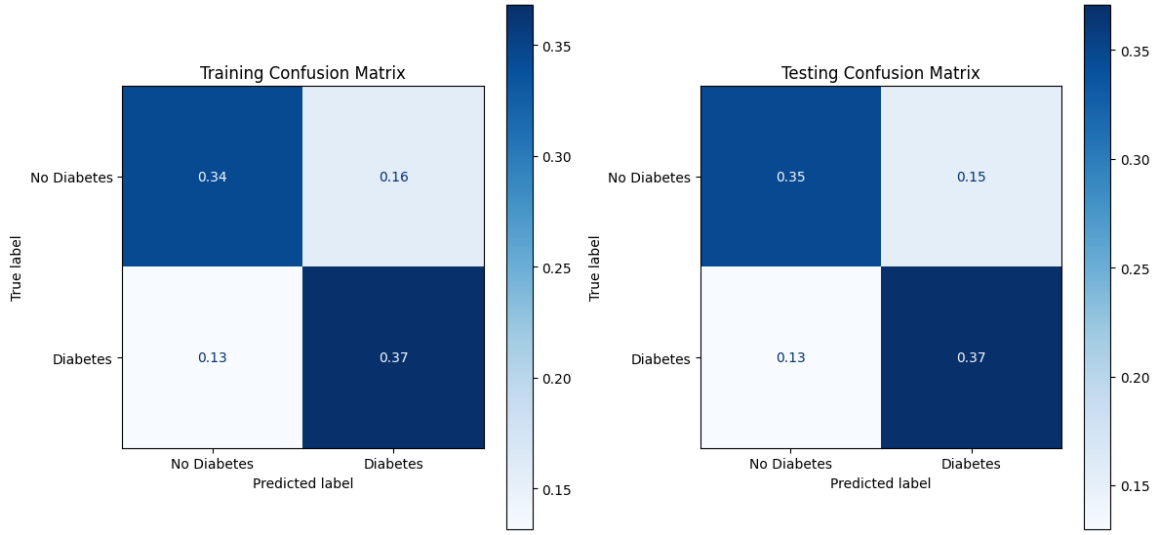
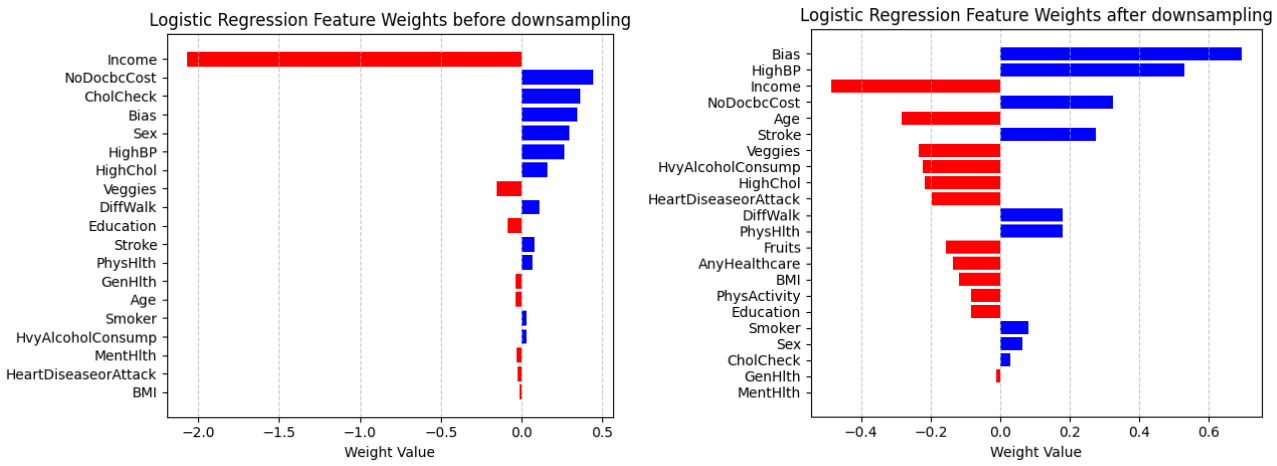


Figure 3: Confusion Matrices for Diabetes Classification After Downsampling

The feature weights of both linear and logistic regression models provided valuable insights. Since the features were standardized, we could compare their weights directly. For the linear model, we displayed only the weights with an absolute value greater than 10^{-2}, and excluded the weight for the White Ethnicity feature. Weights below this threshold had little impact on predictions due to their small magnitudes. The White Ethnicity weight was excluded because it was around 37, making it difficult to display other weights. This feature was by far the most influential, indicating a strong direct relationship between being White and having a higher oral temperature. The negative class for the White feature (representing all other ethnicities) had a value around -1 due to standardization, suggesting an inverse relationship between belonging to any other ethnicity and high oral temperature. Many continuous features were highly correlated as seen during preprocessing, which we believe is what caused some features to have negligible weights, as their predictive information was captured by more influential features.

For the logistic model, income initially had the highest absolute weight, around -2.06, but after downsampling, high blood pressure (HighBP) became the most influential feature, with a weight around 0.5. This showed a direct relationship between high blood pressure and diabetes. Income remained influential, with an inverse relationship, indicating that higher income is associated with lower diabetes likelihood. Other important features included having had a stroke, being unable to afford a doctor's visit, and having difficulty walking, all of which directly contributed to being classified as diabetic. Additionally, the model relied heavily on bias, with the bias term having a greater weight than any individual feature. We also observed that heavy alcohol consumption and high cholesterol surprisingly had an inverse relationship to diabetes classification. We found this rather odd, as we all associated these traits with a higher likelihood of having diabetes.



(a) Before

(b) After

Figure 4: Comparison of Logistic Regression Feature Weights Before and After Downsampling

When experimenting with different train/test splits, we observed similar trends for both models. For the linear model, as the training size increased, the gap between training and testing performance narrowed. Training performance slightly decreased while testing performance improved, suggesting that larger training sizes (around 70-80%) resulted in better overall model performance.

For the logistic model, we saw similar results, though with more variance due to fewer splits being tested. Based on F1 score, log loss, and recall, a 70-80% train/test split appeared to give optimal performance. Although precision and accuracy showed mixed results, the small change in values (as shown on the y-axis) confirmed that these findings were still consistent with our conclusion.

When analysing the y-axis of the second dataset, we see that although we can observe some improvement, it is on a small scale. We believe that the improvement in increased training size is small, as the dataset contains more than enough data points to reach the optimal weights regardless of the training size. Although this possibly would not be true for extremely small training sizes such as 5%, for the 20-80% range we believe that there is still an adequate amount of data points to train a near-optimal model.

We also experimented with different mini-batch sizes for both linear and logistic regression using mini-batch SGD as the optimizer. The results showed that larger batch sizes led to slightly worse performance, while smaller batches, around 8-16, maximized performance. As expected, smaller batch sizes required more iterations to converge, but the trade-off was better performance. The logistic regression graphs showed more variance, likely due to the randomness of mini-batch SGD, where a small gradient norm can end the optimization loop. Using the loss function as a loop condition was tested but significantly slowed down training, so we opted for the gradient norm.

Varying learning rates for mini-batch SGD revealed that lower rates led to smoother convergence, while higher rates risked not reaching optimal weights or diverging. In the linear regression model, any learning rate greater than or equal to 0.9 resulted in divergence, leading to NaN predictions. The SSE loss change as a function of learning rate followed a trajectory similar to the exponential function, showing how enlarging the learning rate can quickly degrade performance. Smaller learning rates required more iterations, but as the learning rate approached the divergence point, oscillation increased, causing more steps to reach convergence. This is why in the case of linear regression, our number of iterations increases as we increase the learning rate, while it decreases when we do so for logistic regression. Our results suggested that for both models, a learning rate of 0.0001 was near optimal.

Finally, comparing analytical and mini-batch SGD linear regression, we found that both approaches can achieve similar performance. This is provided mini-batch SGD's hyperparameters, specifically the learning rate and maximum number of epochs, are carefully selected. For the metrics displayed in Table 1, we used a learning rate of 0.0001, a mini-batch size of 8, a max_epochs value of 10,000, and an epsilon value of 0.1, as we found these to be optimal hyper-parameters from our previous experimentation. Excessively high learning rates or too few epochs resulted in sub-optimal performance, far from the analytical model's results. This experiment highlighted the importance of hyperparameter tuning, as correct values can bring mini-batch SGD performance close to the closed-form solution,

while improper values result in significantly worse outcomes.

Metric	Analytical		Mini Batch SGD	
	Training	**Testing**	**Training**	**Testing**
SSE Loss	0.0304	0.0358	0.0311	0.0346
RMSE	0.2464	0.2676	0.2493	0.2629
MAE	0.1937	0.2070	0.1959	0.2021
R^2	0.7516	0.7733	0.7450	0.7841

Table 1: Metrics of Analytical vs Mini-Batch SGD Linear Regression

Originality/Creativity

As a group, we took several steps and experimented with various techniques to build strong ML models. In preprocessing, we tested different methods to optimize model training. We used a validation set to tune the hyper-parameters of gradient descent and mini-batch SGD for both models, ensuring no overfitting to the test sets. This allowed us to achieve strong results while ensuring our models could generalize well to unseen data. We also standardized the data, which had no noticeable effect on performance but enabled us to compare feature weights and assess feature importance for prediction and classification. Additionally, we removed many duplicate rows in the CDC dataset, ensuring all data points were considered equally and slightly improving dataset balance. To fully balance the data, we experimented with upsampling and downsampling, opting for downsampling, which gave better results due to the large amount of data available. We also analyzed correlation matrices and used various graphs to identify feature collinearity, imbalances, and outliers in both datasets.

In model implementation, we added verbose options for detailed insights into the training process, allowing us to make meaningful adjustments for improvement. We also experimented with using the models corresponding loss functions as the termination condition for both gradient descent algorithms. We incorporated multiple metrics to give a clearer picture of model performance: RMSE, MAE, and R^2 for the linear model, and accuracy, precision, recall, F1-score, and confusion matrices for the logistic model. These experiments and adjustments led to strong performance, particularly with the logistic regression model, where we improved the F1-score from 0.22 to 0.72.

Conclusion

Our experiments highlighted key insights into the performance of linear and logistic regression models. Hyperparameter tuning, including learning rate and batch size, significantly impacted convergence and model performance. Balanced train/test splits were crucial for reliable metrics, and the utilization of a validation set proved valuable for hyperparameter optimization without overfitting to the testing data. Addressing data imbalance also improved model evaluation, particularly for precision, recall, and F1 scores.

For future work, experimenting with advanced optimization techniques like momentum or RMSProp could enhance model performance. We could also experiment with feature removal, particularly on the first dataset, as there was high correlation between many of the features. Although our model accounted for this with setting the weights of these features close to zero, having less features would allow for quicker and less computationally intensive training. Overall, careful data preprocessing, balanced data, and proper hyperparameter selection are essential for building effective regression models, and these potential enhancements could further boost model accuracy and interpretability.

Statement of Contributions

We all wrote our own code and reports, met up together to discuss different experiments, results, and methods to display our results. We chose what worked best from each of our reports to create a final notebook and report. Kayvan's data preprocessing strategies and insights, Léo's graphics and creative ideas to visualize and express the datasets and results were the highlights of the assignment and Kohei worked to summarize it all together at the end to finalize the report.

References

Wang, Q., Zhou, Y., Ghassemi, P., Chenna, D., Chen, M., Casamento, J., ... McBride, D. (2023). *Facial and oral temperature data from a large set of human subject volunteers (version 1.0.0).* PhysioNet. Retrieved from https://doi.org/10.13026/3bhc-9065 doi: 10.13026/3bhc-9065

Wang, Q., Zhou, Y., Ghassemi, P., McBride, D., Casamento, J. P., & Pfefer, T. J. (2021). Infrared thermography for measuring elevated body temperature: Clinical accuracy, calibration, and evaluation. *Sensors*, *22*.

Appendix

	Datset 1		Dataset 2 Before Downsampling		Dataset 2 After Downsampling	
Metric	**Training**	**Testing**	**Training**	**Testing**	**Training**	**Testing**
SSE Loss	0.0304	0.0358	-	-	-	-
RMSE	0.2464	0.2676	-	-	-	-
MAE	0.1937	0.2070	-	-	-	-
R^2	0.7516	0.7733	-	-	-	-
Log Loss	-	-	0.7333	0.7331	0.6744	0.6731
Accuracy	-	-	0.8508	0.8503	0.7123	0.7178
Precision	-	-	0.5488	0.5389	0.7023	0.7086
Recall	-	-	0.1384	0.1394	0.7366	0.7407
F1 Score	-	-	0.2210	0.2214	0.7190	0.7243

Table 2: Metrics of the Linear and Logistic Regression Models After Training on an 80-20 Training/Test Split

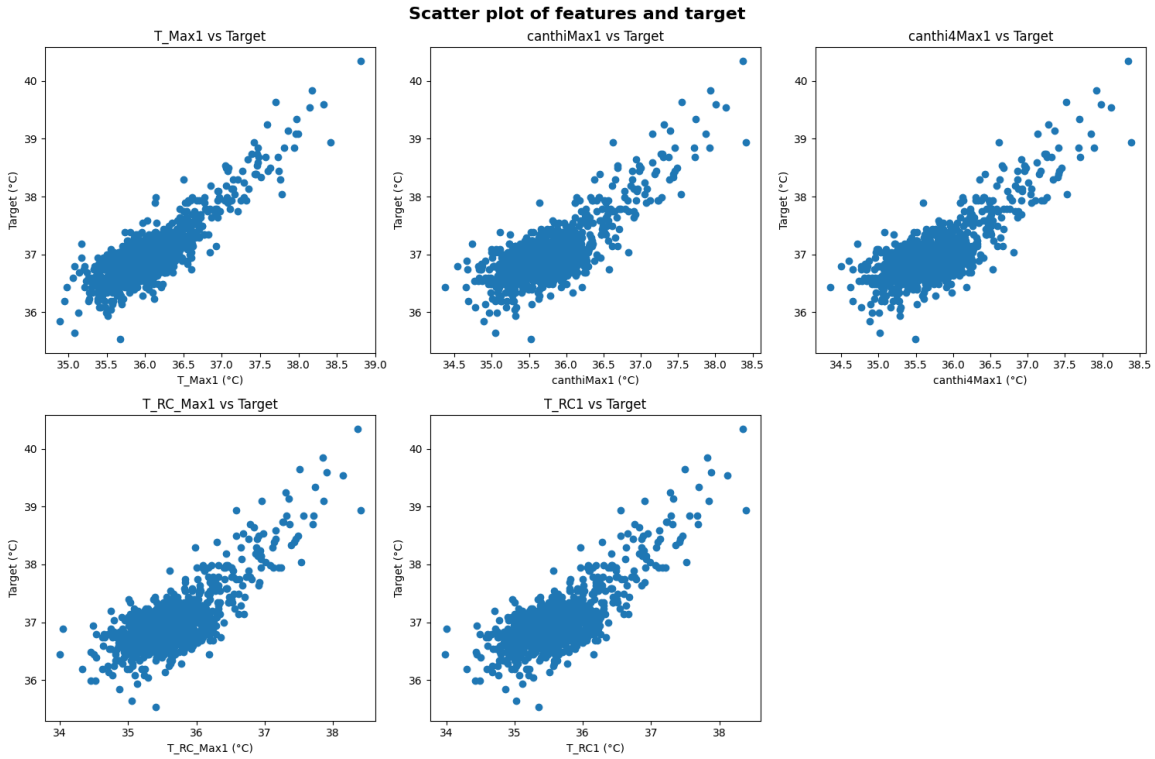


Figure 5: Scatter Plots of Features With High Correlation to the Target of Dataset 1

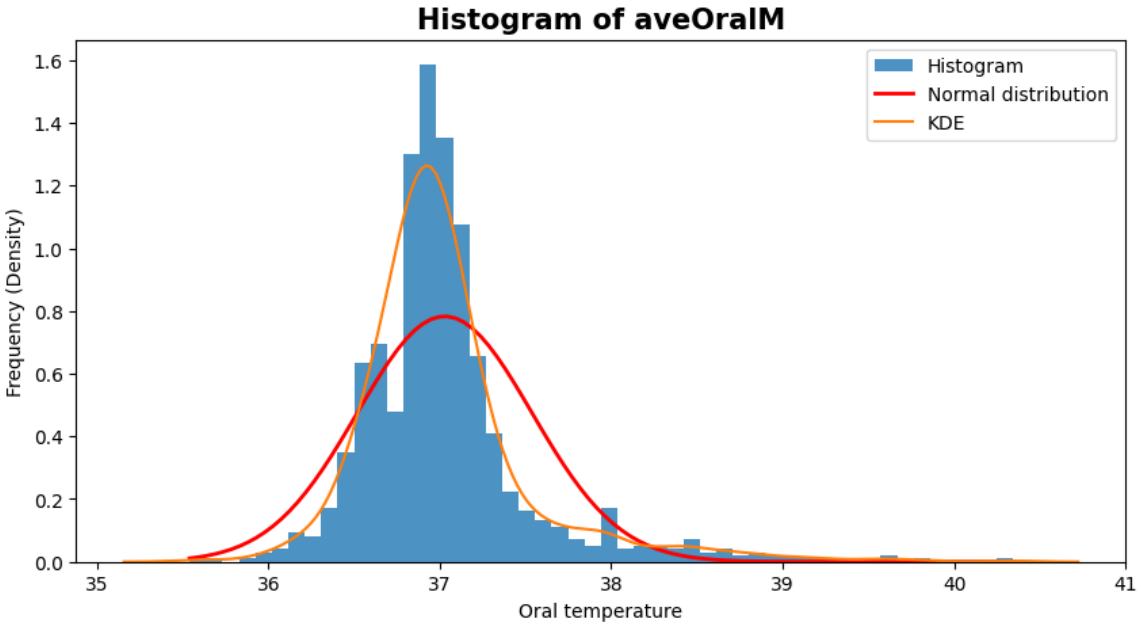


Figure 6: Distribution of the Target Variable of Dataset 1

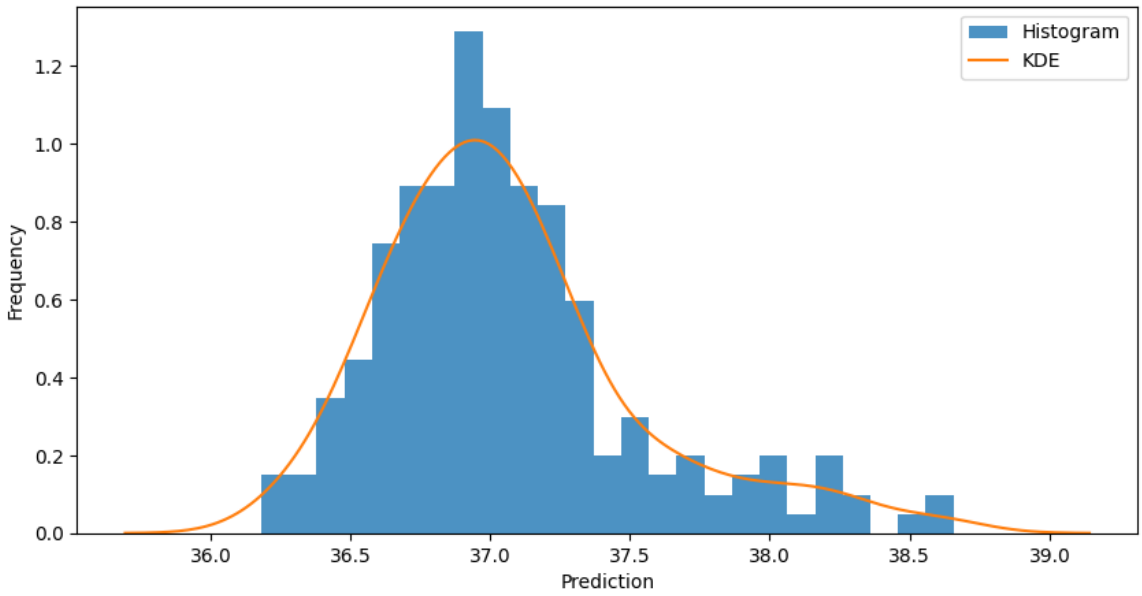


Figure 7: Distribution of the Predicted Values of the Test Split from Dataset 1

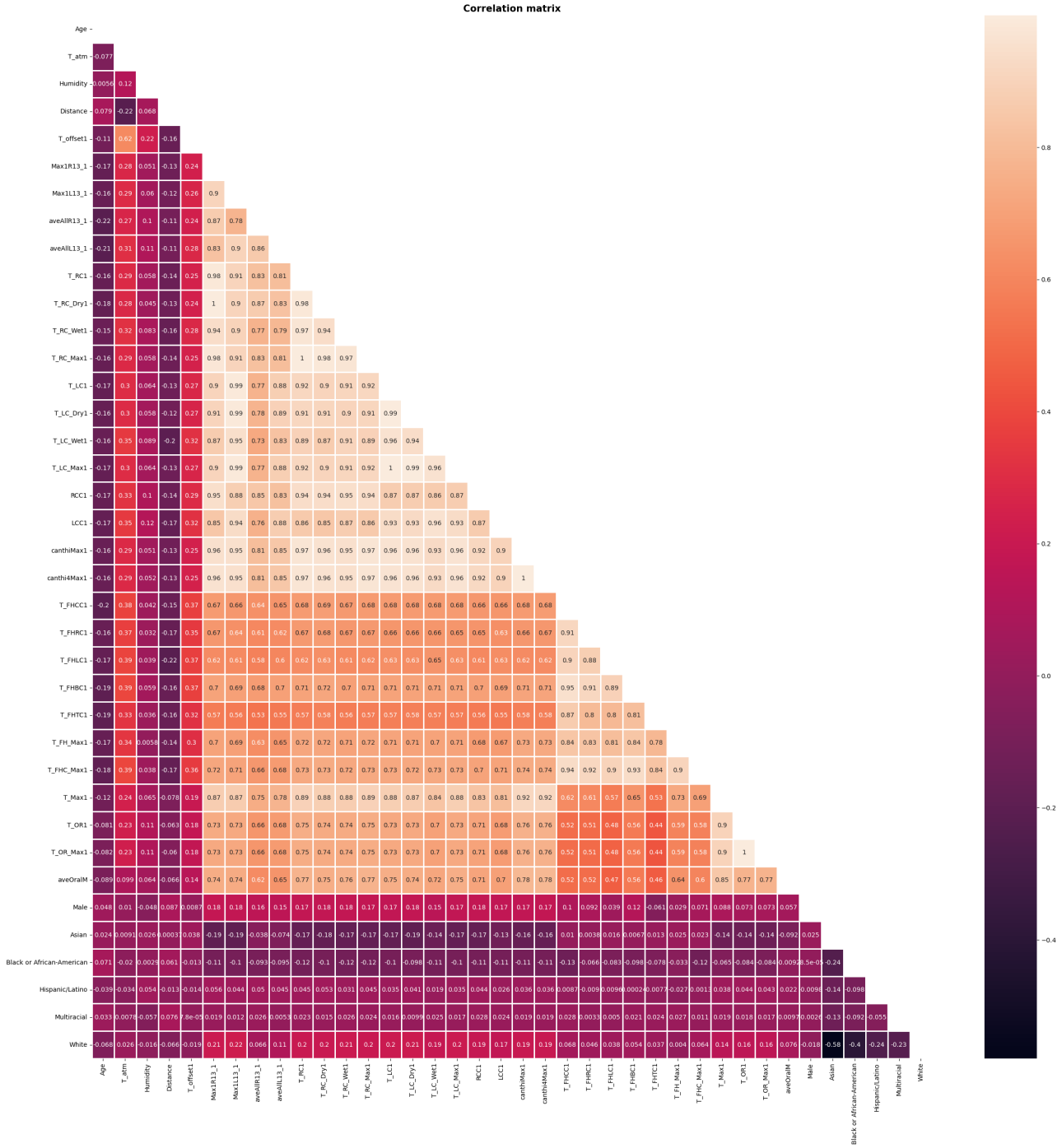


Figure 8: Correlation Matrix of Dataset 1

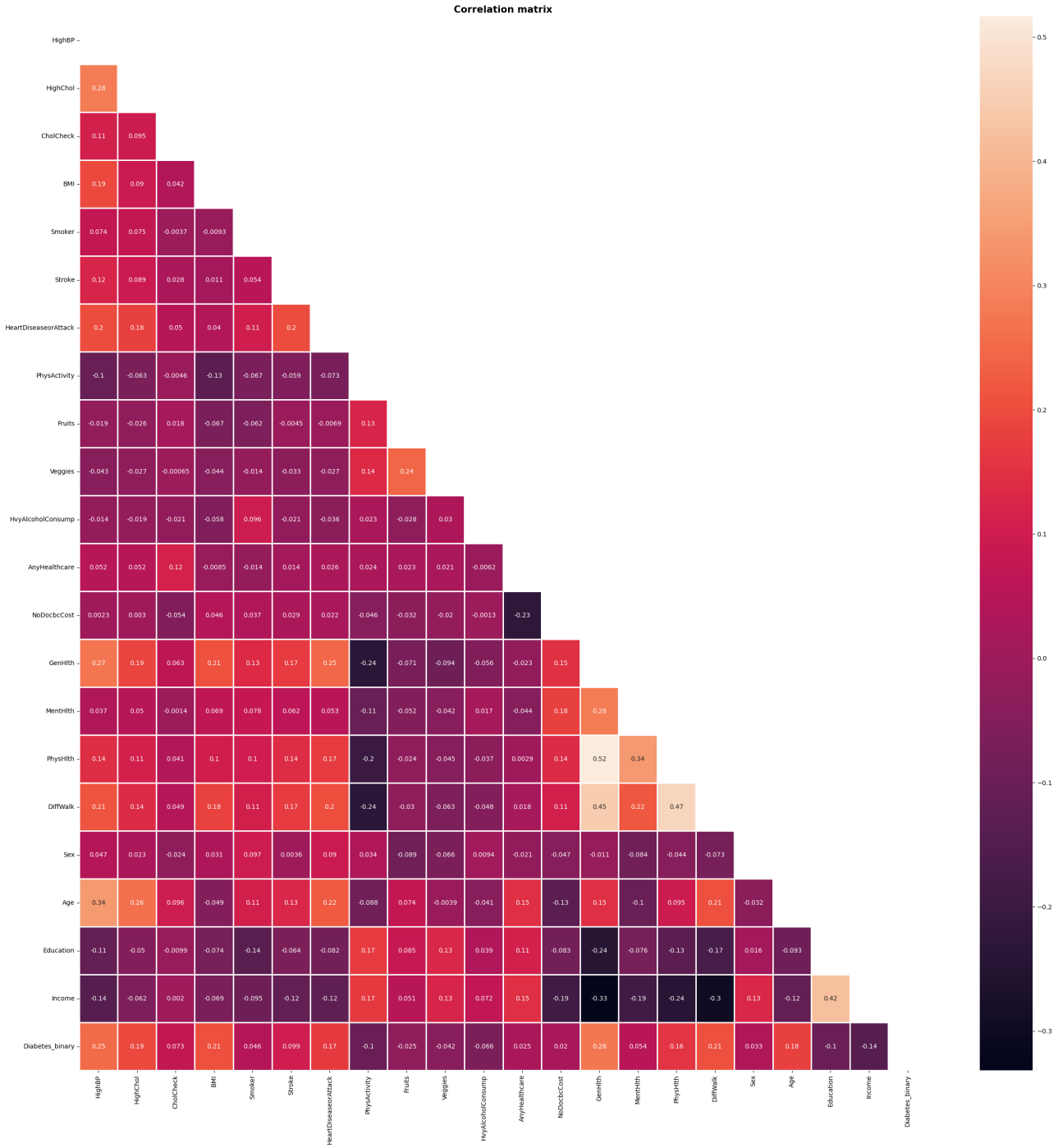


Figure 9: Correlation Matrix of Dataset 2

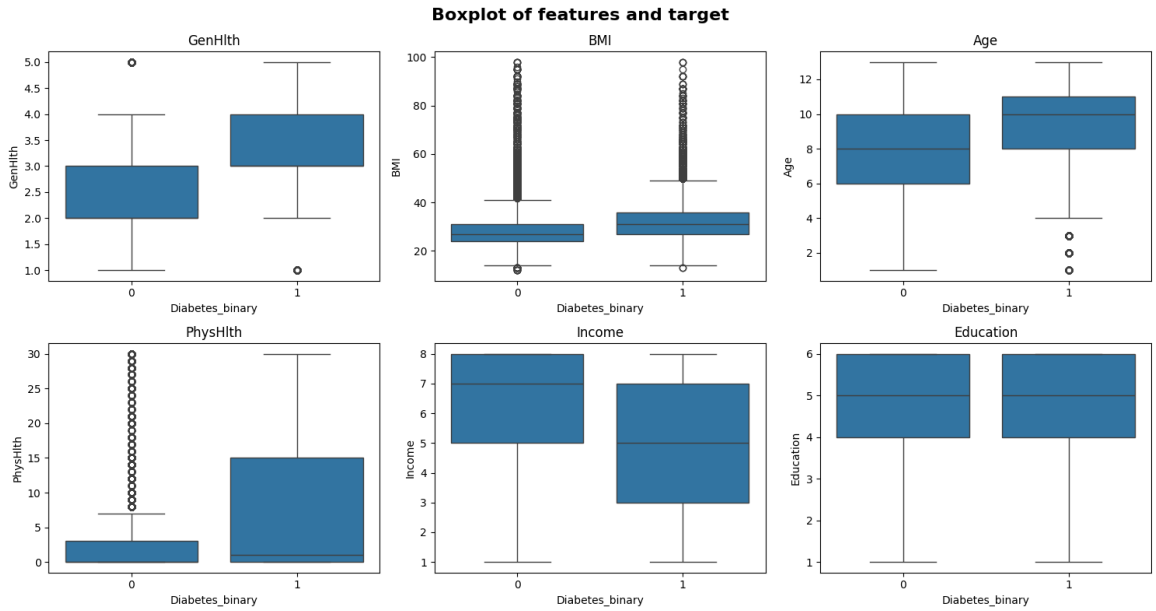


Figure 10: Box Plots of Features with High Correlation to the Target of Dataset 2

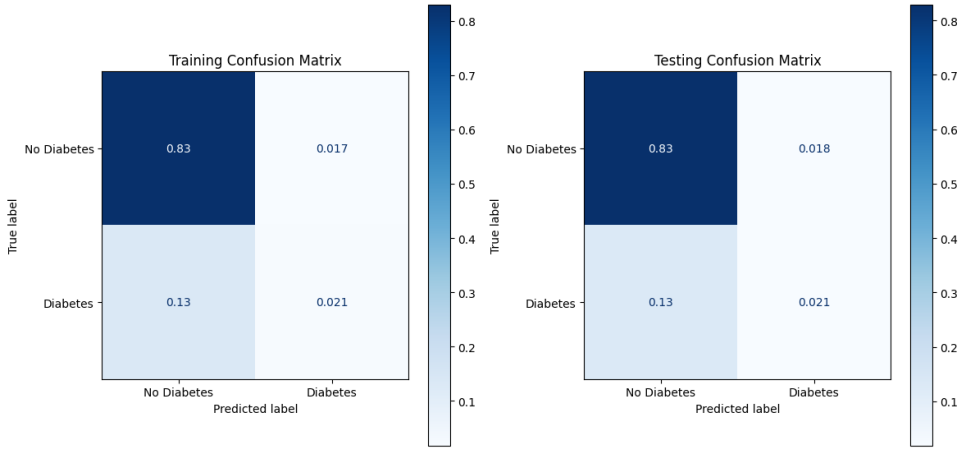


Figure 11: Confusion Matrices for Diabetes Classification Before Downsampling

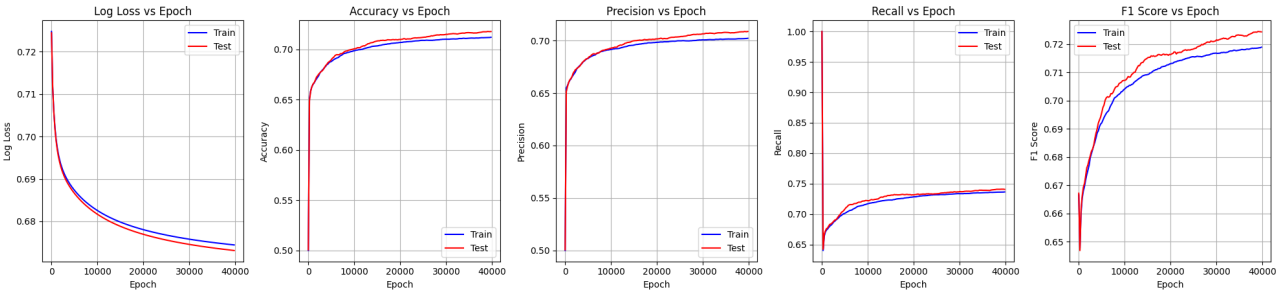


Figure 12: Training vs Testing Performance Metrics for Diabetes Classification After Downsampling

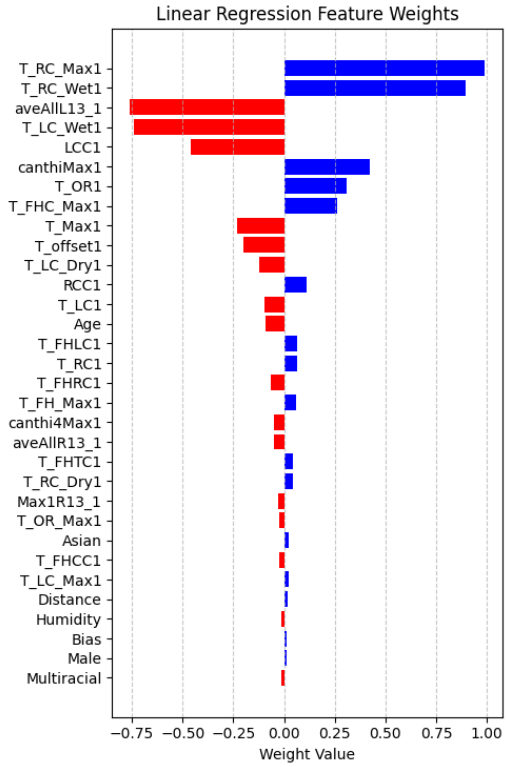


Figure 13: Linear Regression Model Feature Weights for Temperature Prediction

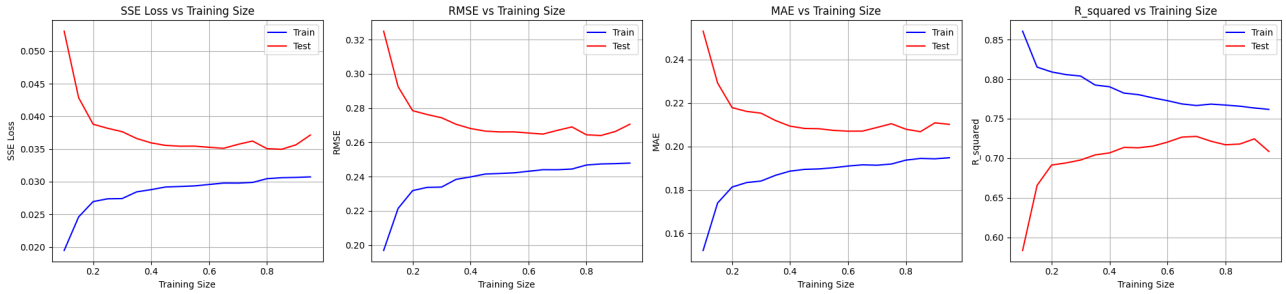


Figure 14: Effect of Training Size on Linear Regression Performance Metrics

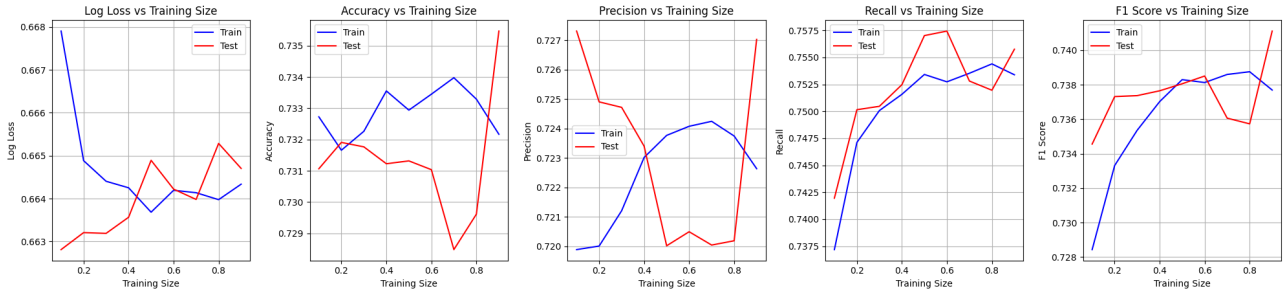


Figure 15: Effect of Training Size on Logistic Regression Performance Metrics

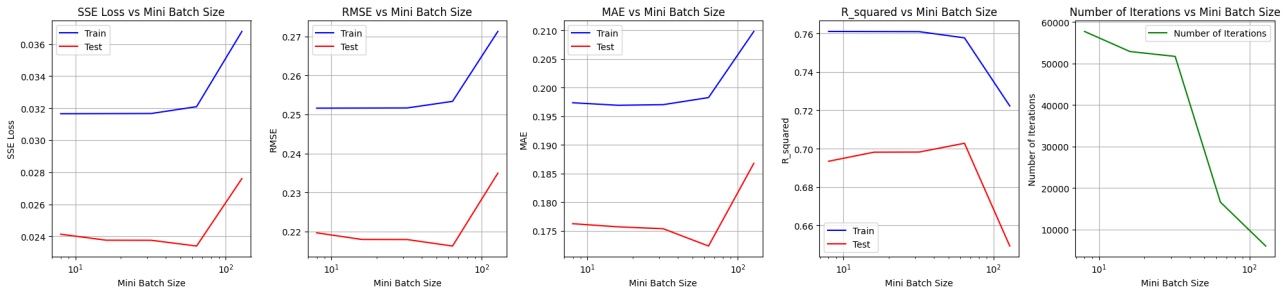


Figure 16: Effect of Mini Batch Size on Linear Regression Performance and Iterations

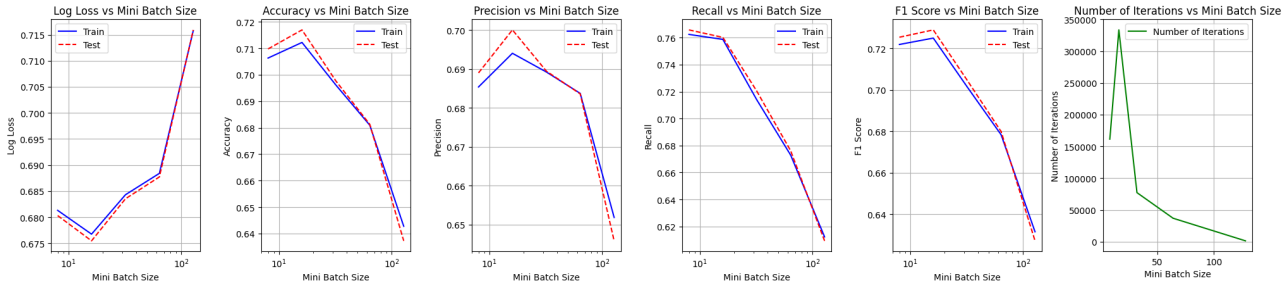


Figure 17: Effect of Mini Batch Size on Logistic Regression Performance and Iterations

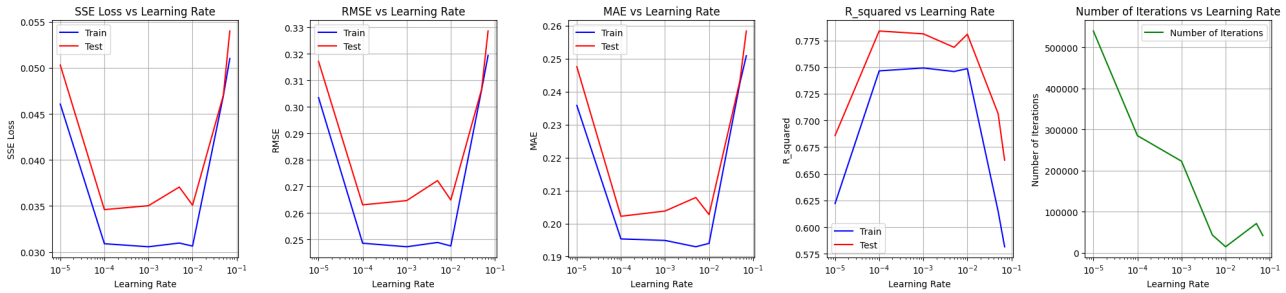


Figure 18: Effect of Learning Rate on Linear Regression Performance and Iterations Using Mini-Batch SGD

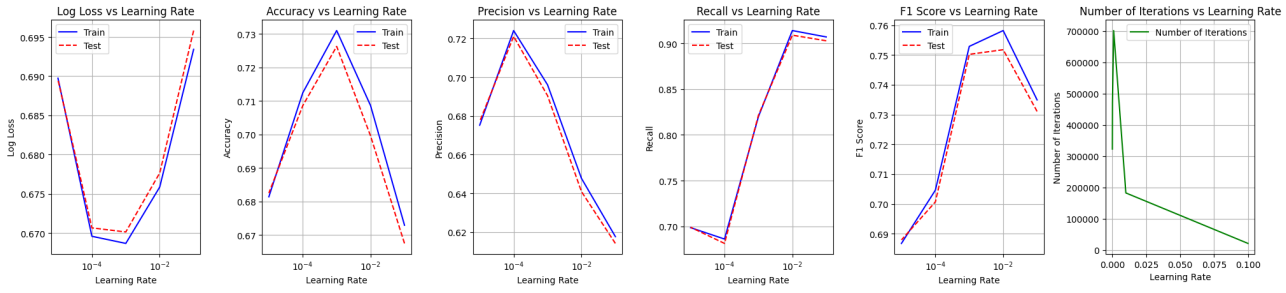


Figure 19: Effect of Learning Rate on Logistic Regression Performance and Iterations Using Mini-Batch SGD