COMP 551
Assignment 1

Jessie Kurtz, Marc-Antoine Nadeau, Baicheng Peng

2024-09-30

Abstract

Our project involved two machine learning tasks: predicting average oral temperature using an Infrared Thermography Temperature dataset, and diagnosing diabetes based on a Diabetes Health Indicators dataset. We began by analyzing the dataset characteristics and performing necessary data preprocessing, including handling missing and duplicate values, scaling features, addressing class imbalance, and removing highly correlated features.

We then implemented linear and logistic regression models, utilizing mini-batch stochastic gradient descent for regression and classification. The models were evaluated using various metrics, and experiments were conducted to assess their behavior under different conditions. For both linear and logistic regression models, the mini-batch stochastic gradient presented the best trade-off between performance and convergence speed. These experiments also revealed meaningful patterns and opportunities for model optimization.

Table of Contents

1 Introduction

The Infrared Thermography Temperature dataset contains facial temperature measurements obtained using two infrared thermography (IRT) devices. It includes oral temperature recordings of each patient, captured in both fast mode and monitor mode, with the latter being used as the **target** due to its higher accuracy. The primary goal of the dataset is to evaluate the effectiveness of IRT in detecting elevated body temperature (EBT) and to provide metrics for assessing its clinical accuracy.

The Diabetes Health Indicators dataset includes health statistics and lifestyle data. The **target** variable is binary, where 0 indicates no diabetes, and 1 indicates pre-diabetes or diabetes. The primary goal of the dataset is to understand diabetes risk factors and health outcomes.

Key metrics such as compute time, mean squared error (MSE), R-squared (R^2) for linear regression, and accuracy, precision, recall, and F1 score for logistic regression were analyzed to evaluate the efficiency and predictive power of each regression method detailed in the report.

In the linear regression task, Mini-Batch Stochastic Gradient Descent (MBSGD) offered a good balance between speed and accuracy, while MBSGD with Adam Optimization was slightly less accurate but required more time. The analytical method had the quickest compute time and matched Adam in accuracy. For logistic regression, MBSGD provided high accuracy and a strong F1 score with relatively fast performance, while Adam optimization took much longer and had slightly lower accuracy and F1 score. Full-batch logistic regression had similar performance to MBSGD but took more time.

These findings demonstrate that MBSGD offers a balance between computational efficiency and predictive performance, while the Adam optimizer, though slower, may offer benefits in some complex scenarios. Full-batch methods remain competitive, especially for tasks where batch processing is preferred.

For the Infrared Thermography Temperature Dataset, the metric Mean Absolute Error is similar to that of other studies, where it was found to be between 0.2 to 0.25 **[6]**. For the Diabetes Dataset, there is still a gap in performance when comparing our dataset to ones online, which have F1 scores around 0.84 from logistic regression and 0.87 from random forest modelling **[7]**.

2 Datasets

2.1 Infrared Thermography Temperature Dataset - Linear Regression

2.1.1 Understanding the Data

1. Plot Histograms for Each Feature

Histograms plot the distribution of values for each feature, highlighting any bias or outliers. This figure can be found in the Appendix as Figure 1 and the primary findings in Table 1.

2. Understand Correlation and Relationships

The correlation coefficient was used to examine the relationship between 'aveOralM' and other features. Additionally, the Variance Inflation Factor (VIF) was employed to detect multicollinearity. A high VIF value signals that a variable is highly correlated with others in the dataset, leading to inflated variance and reduced predictive power [4]. High VIF and high correlation together indicate redundancy, suggesting that such variables should be considered for removal from the dataset to avoid multicollinearity [5]. Many temperature features presented very high VIF and high correlation with the others in the dataset, making them candidates for removal, as shown in Figures 2 and 3. The final features used are shown in Table 2.

2.2 Diabetes Health Indicators Dataset - Logistic Regression

2.2.1 Understanding the Data

1. Plot Histograms for Each Feature

The histograms can be found in the Appendix as Figure 4. As mentioned in Table 3, the **target** value is skewed towards zero. The methods used to correct for this imbalance are described in 4.1.

2. Understand Correlation and Relationships

No features presented both high VIF and high correlation, as shown in Figures 5 and 6. However, features with low correlation to the target typically have low predictive power and add more noise to the model. As such, the final selected features are shown in Table 4.

2.3 Preparing the Data

The steps for preparing the linear regression dataset are outlined in Table 5 of the Appendix. The same steps outlined in Table 5 were used for logistic regression, except that oversampling was done before splitting the dataset into training and testing data. There were also no missing data, outliers or categorical feature encoding.

3 Results

3.1 Performance Evaluation of Linear Regression and Fully Batched Logistic Regression

As shown in Table 6, the Mean Absolute Error and Mean Squared Error are comparable between the training and testing datasets; however, the R^2 value is higher in the training model suggesting that the model fits the training data more accurately. Overall, the model demonstrates reasonable performance. Furthermore, we observe in Figures 7a and 7b that our trained linear regression model accurately represents the dataset, as the residuals in both figures are centred around the zero residual line (i.e., perfect correlation, indicated by the red dashed line). Similarly, we see in Figure 8 that the predicted values are close to the real values.

As observed in the confusion matrices shown in Figure 9a and 9b, our trained logistic regression model accurately classifies the majority of instances in both the training and testing datasets. Specifically, as illustrated in Table 7 the model achieves an accuracy, precision, recall, and F1 score of approximately 0.73 for both the training and testing datasets, indicating consistent performance.

3.2 Model Weights and Feature Performance

For the linear regression model, the bias term is the most significant weight, with a value around 37 across the analytical, mini-batch stochastic, and ADAM optimization techniques. Aside from this, as shown in Table

2, the parameters with the largest weights are T_OR_Max1 and Max1L13_1, approximately 0.27 and 0.21, respectively. All other weights are very close to 0, indicating that they contribute minimally to the model's predictions.

From Table 4 we observe that the most significant positive weights associated with the logistic regression model are GenHlth, BMI, Age, HighBP, HighChol, and CholCheck, with respective values of approximately 0.60, 0.50, 0.40, 0.36, and 0.27. Therefore, the model suggests that the best predictors for detecting diabetes are general health, BMI, age, cholesterol, and blood pressure. Additionally, the largest negative weight, indicating an inverse correlation with diabetes, is HvyAlchoholConsump, with a value of approximately -0.17.

3.3 Impact of Training Data Size on Model Performance

As expected, increasing the size of the training dataset leads to finer and more accurate models for both analytical and mini-batch stochastic gradient descent optimization techniques, when applied to both training and testing datasets. This trend is evident in Figure 10 where the explained variance score (10a) and R^2 (10c) tend towards 1, whereas the mean squared error (10b) approaches 0.

Similar to the case of linear regression, we observe that our logistic regression model achieves increasing accuracy and a more refined model as the dataset size grows. However, this increase is quite subtle, as seen in Figures 11a, 11b, and 11c, all of which exhibit an upward trend with a minimal growth rate. This subtle increase can be attributed to the size of the dataset and our implementation of oversampling, which further enlarges the dataset. This means that even with just 20% of the training dataset, the model is still exposed to a significant amount of data.

3.4 Batch Size Metric

3.4.1 Linear and Logistic Regression

Based on the training loss and R^2 score graphs for both linear and logistic regression models, shown in Figure 12 of the Appendix, the following conclusions can be drawn regarding the impact of batch size on convergence speed and final performance:

- **Convergence Speed:**
  - **Smaller batch sizes (8, 16, 32)** consistently converge the fastest in terms of both training loss and R^2 score (for linear regression). Batch size 32 provides the best balance in both models, achieving rapid convergence with minimal oscillations.
  - **Larger batch sizes (64, 128, 772, 183579)** exhibit slower convergence. In both models, full-batch sizes (e.g., 183579, 772) take longer to reduce the loss initially but provide more stable learning overall. This effect is more noticeable in linear regression, where larger batch sizes significantly slow down convergence, making it take longer to reach the final loss. In contrast, logistic regression tends to converge faster, so the difference in convergence speed across batch sizes is less pronounced.
- **Final Performance:**

– All batch sizes converge to similar final loss values in both models. Linear regression loss stabilizes around 10^{-1}, while logistic regression stabilizes around 3.5×10^{-1}. Smaller batch sizes (16, 32) reach these values faster, while larger batch sizes converge more slowly but provide smoother final performance.

- **Best Configuration:**
  – In both models, **batch size 32** strikes the best balance between fast convergence and stable performance, outperforming both smaller and larger batch sizes. It provides rapid learning while avoiding oscillations, making it the optimal choice overall.

3.5 Learning Rate Metric

3.5.1 Linear Regression

Based on the training loss and R^2 score graphs for both linear and logistic regression models, shown in Figure 13 and 14 of the Appendix, respectively, the following conclusions can be drawn regarding the impact of the learning rate on model performance:

- **MSE:** For Mini-Batch Stochastic Gradient Descent (MBSGD), the mean-squared error(MSE) increases consistently as learning rates rise. Both training and test set errors grow larger with higher learning rates, indicating that the model struggles to fit the data properly.
- **R^2 Score:** The R^2 score decreases sharply as learning rates increase, reflecting a loss in the model's ability to explain the variance in the data. This decline is observed for both the training and test sets.
- **Suitable Learning Rate:** A learning rate of 0.01 provides the best balance between minimizing MSE and maintaining a high R^2 score. As learning rates increase beyond 0.1, both error and explanatory power suffer, making higher learning rates unsuitable for linear regression with MBSGD.

3.5.2 Logistic Regression

- **Accuracy:** Accuracy provides a broad view of model performance but may not reflect class imbalances. With MBSGD, accuracy on the training set decreases as learning rates increase, while test set accuracy remains stable.
- **F1 Score:** The F1 score balances both false positives and false negatives. For MBSGD, the F1 score drops significantly as learning rates increase, signaling an imbalance between precision and recall.
- **Precision:** Precision is the proportion of true positive predictions out of all predicted positives. In MBSGD, precision improves at a learning rate of 0.02 but sharply declines afterwards.
- **Recall:** Recall measures the proportion of actual positives correctly identified. In MBSGD, recall improves as the learning rate increases, with the highest recall at 0.05. However, this comes at the expense of precision and F1 score, indicating a trade-off at high learning rates.

- **Suitable Learning Rate:** A learning rate of 0.01 offers the best overall performance across all metrics (accuracy, F1 score, precision, and recall) for MBSGD. Higher learning rates improve recall but degrade precision, accuracy and F1 score, making 0.01 the most effective for balanced performance.

3.6 Comparison of Linear Regression Methods

In testing the performance of the two models (Analytical and mini-batch) on **training set**, the Analytical model always gives the better weights to minimize MSE than the mini-batch model, which is reasonable because Analytical model give the closed form solution which would give optimal solution on the training data it has seen and fit.

However, on the **test set**, the mini-batch model sometimes outperform the analytical model since in the mini-batch explore a more variability of weights during the updating process. Thus for unseen data, mini-batch may give better solution than analytical ones in support of more possibility in updating the weight given by the nature of mini-batch gradient descent. This is illustrated in Table 8 of the Appendix.

4 Originality / Creativity

4.1 Methods to Correct for Imbalanced Dataset

Different methods were used to correct for the highly imbalanced **Diabetes Dataset**.

4.1.1 Adjusting Class Weights

Adjusting class weights was used to account for the disparity between the majority and minority classes. Given the class ratio of approximately 21:3 (Class 0 to Class 1), the respective class weights were calculated to be 1.15 for Class 0 and 7.63 for Class 1. Adjusting class weights improved the model compared to fitting it on the raw imbalanced data, but because the dataset was still highly imbalanced in terms of actual counts, evaluation metrics like Precision, Recall, and F1 score remained relatively poor.

4.1.2 Oversampling/ Under-sampling the Dataset

By definition, oversampling increases the number of instances in the minority class by duplicating existing examples, whereas under-sampling reduces the number of majority class instances to balance with the minority class by randomly discarding some of the majority class data. Both methods led to significant and similar improvements in evaluation metrics. Oversampling is typically a better choice for small datasets since discarding data in under-sampling could lead to a lack of information. However, this method is computationally expensive. Given the size of the dataset and the desire for better performance, under-sampling was selected.

4.2 Adaptive Moment Estimation (ADAM)

ADAM (Adaptive Moment Estimation) was implemented to determine if it could improve on standard optimization techniques, given that it adapts learning rates and uses momentum. The ideal hyper-parameters found through experiments 3.1 to 3.6, shown in Table 9, were used to compare these regression methods. The final results are shown in Table 10 for linear regression, and Table 11 for logistic regression.

5 Discussion and Conclusion

In conclusion, our analysis indicates that the Linear Regression model achieves optimal performance with Mini-Batch Stochastic Gradient Descent as the optimization algorithm, as shown in Table 10. This setup converged in just 0.03 seconds and resulted in an R^2 value of 0.69, demonstrating its effectiveness and accuracy.

Similarly, the Logistic Regression model also achieves optimal performance using the Mini-Batch Stochastic Gradient Descent as the optimization algorithm, as shown in Table 11. It converged in approximately 2 seconds and yielded an F1-score of 0.74. Although ADAM is theoretically expected to improve convergence speed, we observed that it yielded less accurate results and was slower to converge in both linear and logistic regression.

In the future, we hope to experiment with different machine learning algorithms, such as K-Neighbors Classifier and Decision Tree Classifier, to see if we can enhance convergence speed and model accuracy. Additionally, we aim to expand our analysis to include a broader set of metrics to deepen our understanding of the predictions.

6 Statement of Contributions

All team members contributed equally by independently implementing both linear regression and logistic regression models. Once all models were completed, we analyzed each one and merged them by selecting the best aspects of each. From these two merged models, we collaboratively compiled our findings into this report.

References

[1] Q. Wang et al., “Facial and oral temperature data from a large set of human subject volunteers,” *Physionet.org*, May 24, 2023. Available: https://physionet.org/content/face-oral-temp-data/1.0.0/. [Accessed: Sep. 17, 2024].

[2] A. Gholamy, V. Kreinovich, and O. Kosheleva, “Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation,” *ScholarWorks@UTEP*, 2018. Available: https://scholarworks.utep.edu/cs_techrep/1209/. [Accessed: Sep. 17, 2024].

[3] Towards AI Team, “How, When, and Why Should You Normalize/Standardize/Rescale Your Data,” *Towards AI*, May 16, 2019. Available: https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff. [Accessed: Sep. 17, 2024].

[4] C. Team, “Variance Inflation Factor (VIF),” *Corporate Finance Institute*, Aug. 19, 2024. Available: https://corporatefinanceinstitute.com/resources/data-science/variance-inflation-factor-vif/. [Accessed: Sep. 17, 2024].

[5] P. Allison, “When Can You Safely Ignore Multicollinearity? | Statistical Horizons,” *Statistical Horizons*, Sep. 10, 2012. Available: https://statisticalhorizons.com/multicollinearity. [Accessed: Sep. 17, 2024].

[6] V. Untilova, “Oral Temperature Prediction with Python - Data And Beyond - Medium,” *Medium*, Apr. 16, 2024. [Online]. Available: https://medium.com/data-and-beyond/oral-temperature-prediction-with-python-using-infrared-thermography-data-ce40a2361d45. [Accessed: Sep. 30, 2024].

[7] “Diabetes-Health-Indicators/README.md at main · Helmy2/Diabetes-Health-Indicators,” GitHub, 2024. Available: https://github.com/Helmy2/Diabetes-Health-Indicators/blob/main/README.md. [Accessed: Sep. 30, 2024].

7 Appendix

Feature(s)	**Findings**
Humidity, Distance, Age	These features show skewed distributions with most values clustered toward the lower end. The age skewing is influenced by the fact that 95% of the subjects were under 30 years old, with about half in the 18–21 age range **[1]**.
Temperature features (e.g., T_FHC1, T_OR1)	Many temperature-related features do not have extreme outliers and are approximately normally distributed.

Table 1: Summary of Histogram Findings for IRT Dataset

Weight Name	**Optimization Techniques**		
	Analytical	**Mini-Batch Stochastic**	**ADAM**
Bias	37.0377	37.0376	37.0293
Gender	-0.0221	-0.0217	-0.0165
Age	0.0002	0.0009	-0.0047
Ethnicity	-0.0538	-0.0514	-0.0453
T_atm	-0.0986	-0.0978	-0.0864
Humidity	0.0038	0.0033	0.00197
Distance	-0.0099	-0.0095	-0.02998
T_offset1	0.0217	0.0210	0.0361
Max1L13_1	0.2293	0.2159	0.1924
aveAllR13_1	-0.0161	-0.0038	-0.0061
T_FHCC1	0.0238	0.0241	0.0245
T_OR_Max1	0.2728	0.2730	0.2638

Table 2: Weight Values for Linear Regression

Feature(s)	Findings
Diabetes_binary, HighBP, HighChol, Smoker, Stroke, HeartDiseaseorAttack, CholCheck, PhysActivity	These binary features are heavily skewed towards '0', indicating that most individuals do not have these conditions or habits.
BMI	The distribution is roughly normal but slightly right-skewed, with most individuals having BMIs in the 20–40 range.
GenHlth	The feature has a relatively even distribution across its categories, with more frequent values in the mid-range (2 to 3).
MentHlth, PhysHlth	Both features are highly right-skewed, indicating that most individuals report very few days of poor mental or physical health, though a few report worse health.
HvyAlcoholConsump	Most individuals are not heavy alcohol consumers, as the distribution is extremely skewed towards '0'.
DiffWalk	Most people do not have difficulty walking, as the majority of values are '0'.
Age	The distribution is slightly bimodal, with concentrations in certain age groups.
Education	More individuals have higher education levels, with a gradual increase towards the higher categories.
Income	The income feature shows a positive skew, with more individuals having higher income levels.

Table 3: Summary of Histogram Findings for Diabetes Dataset

Weight Name	**Optimization Techniques**		
	Full-Batch	**Mini-Batch Stochastic**	**ADAM**
Bias	-0.0382	-0.0320	-0.0367
HighBP	0.3587	0.3649	0.3473
HighChol	0.2792	0.2476	0.2299
CholCheck	0.2086	0.2358	0.2542
BMI	0.5079	0.4988	0.5348
Smoker	0.0084	0.0352	0.0042
Stroke	0.0384	0.0367	0.0366
HeartDiseaseorAttack	0.1026	0.0948	0.1395
PhysActivity	-0.0054	-0.0087	-0.0079
HvyAlcoholConsump	-0.1633	-0.1995	-0.2119
GenHlth	0.5918	0.6351	0.6234
MentHlth	-0.0447	-0.0347	-0.0497
PhysHlth	-0.0692	-0.0905	-0.0871
DiffWalk	0.0382	-0.0043	-0.0192
Age	0.4072	0.3909	0.3954
Education	-0.0298	-0.0544	0.0441
Income	-0.0752	-0.0640	-0.0816

Table 4: Weight Values for Logistic Regression

Ordered Steps	Description
Remove Duplicate Entries	Duplicate entries can skew data, so they were removed to ensure accuracy.
Handle Missing Data	To handle missing values in the 'Distance' feature and avoid data loss, the mean of the 'Distance' column was used to fill in the gaps, instead of removing rows with missing values. This approach prevents bias in the results.
Remove Outliers	Outliers were detected in features such as 'Age', 'Humidity', and 'Distance', and removed using the Inter-quartile Range (IQR) method.
Label Encode Data	Categorical features ('Ethnicity', 'Gender', and 'Age') were label-encoded to convert them into numerical form for linear regression.
Split Dataset into Training and Testing Data	The dataset was split into 80% training and 20% testing, ensuring independence to avoid overfitting or data leakage [**2**].
Normalize Data	Normalization was applied to scale features evenly, preventing larger features from dominating and improving model convergence [**3**].

Table 5: Steps in Preparing the Dataset

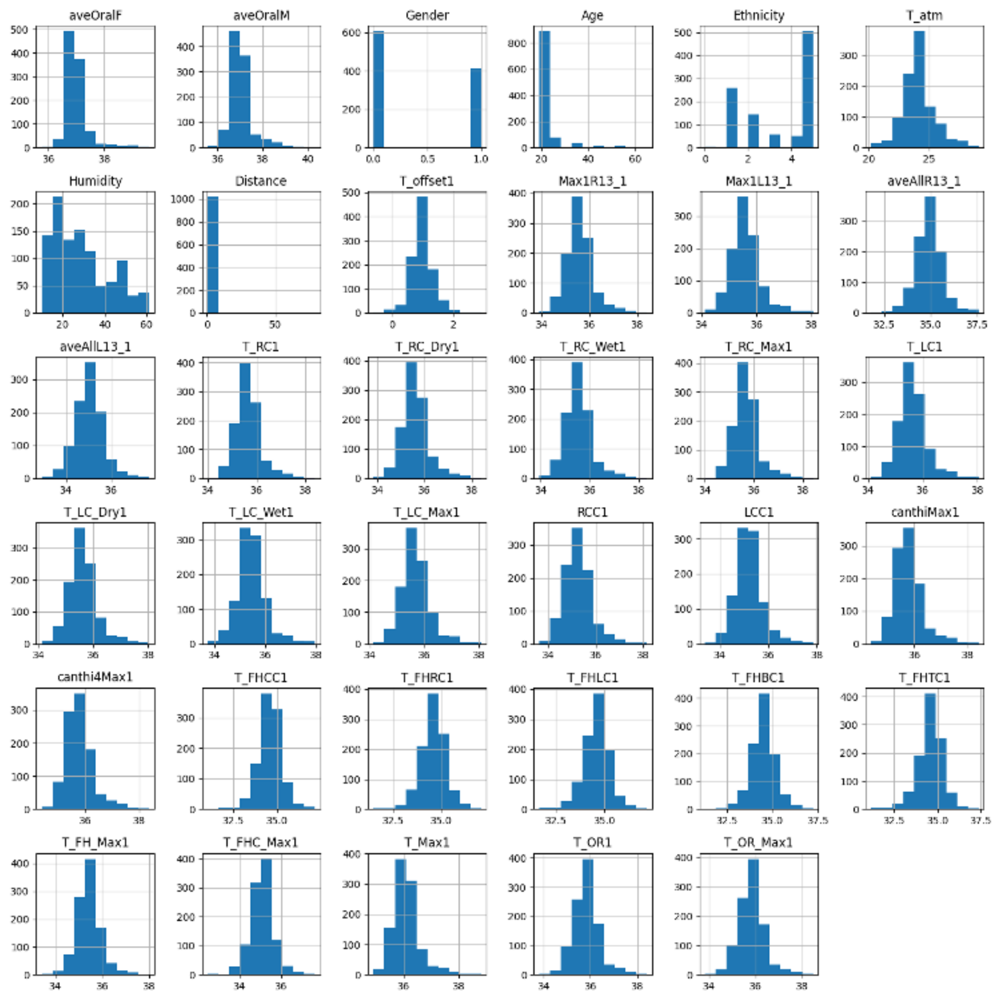


Figure 1: Histogram IRT Dataset

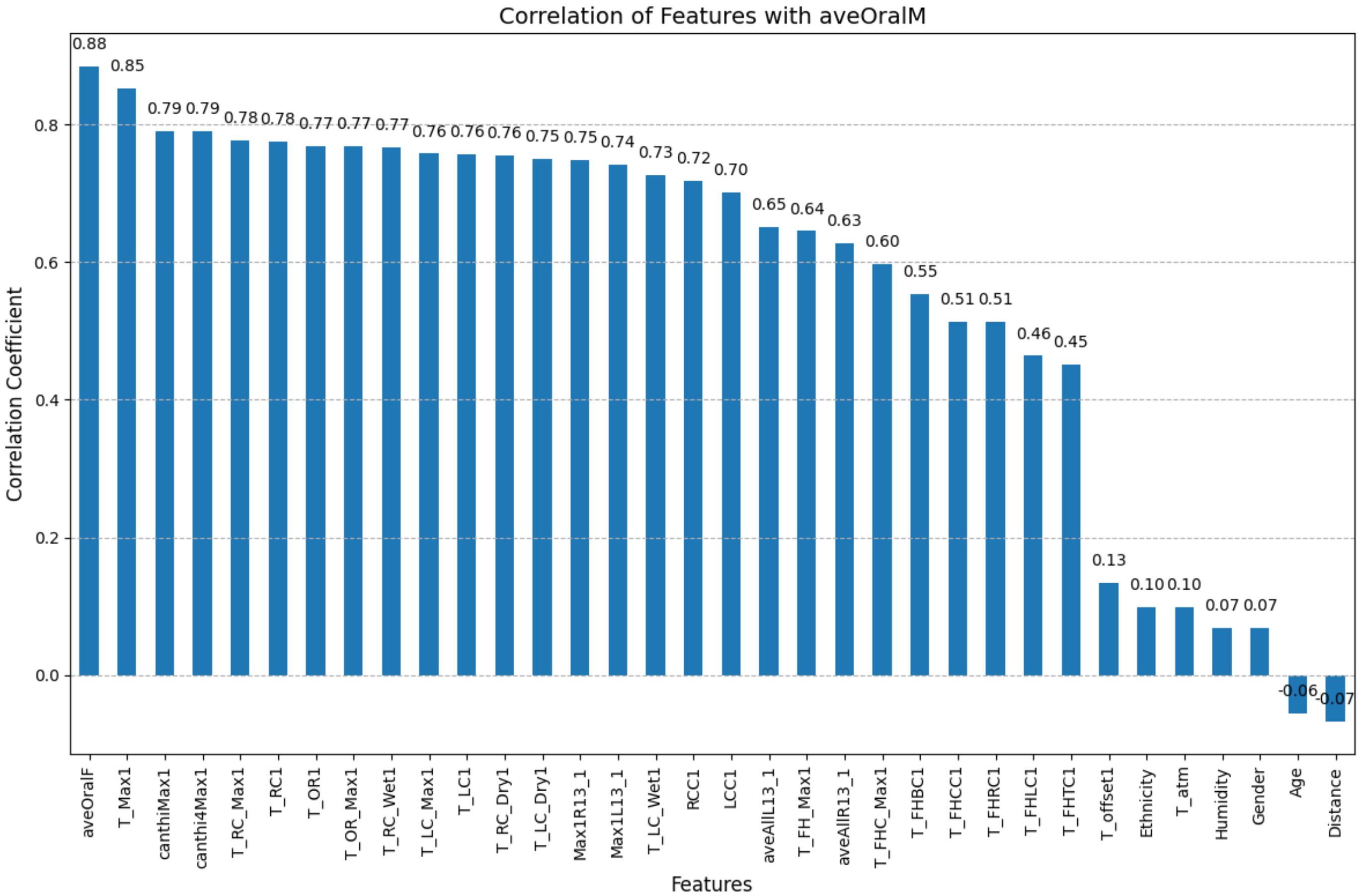


Figure 2: Correlation Coefficient vs Feature IRT Dataset

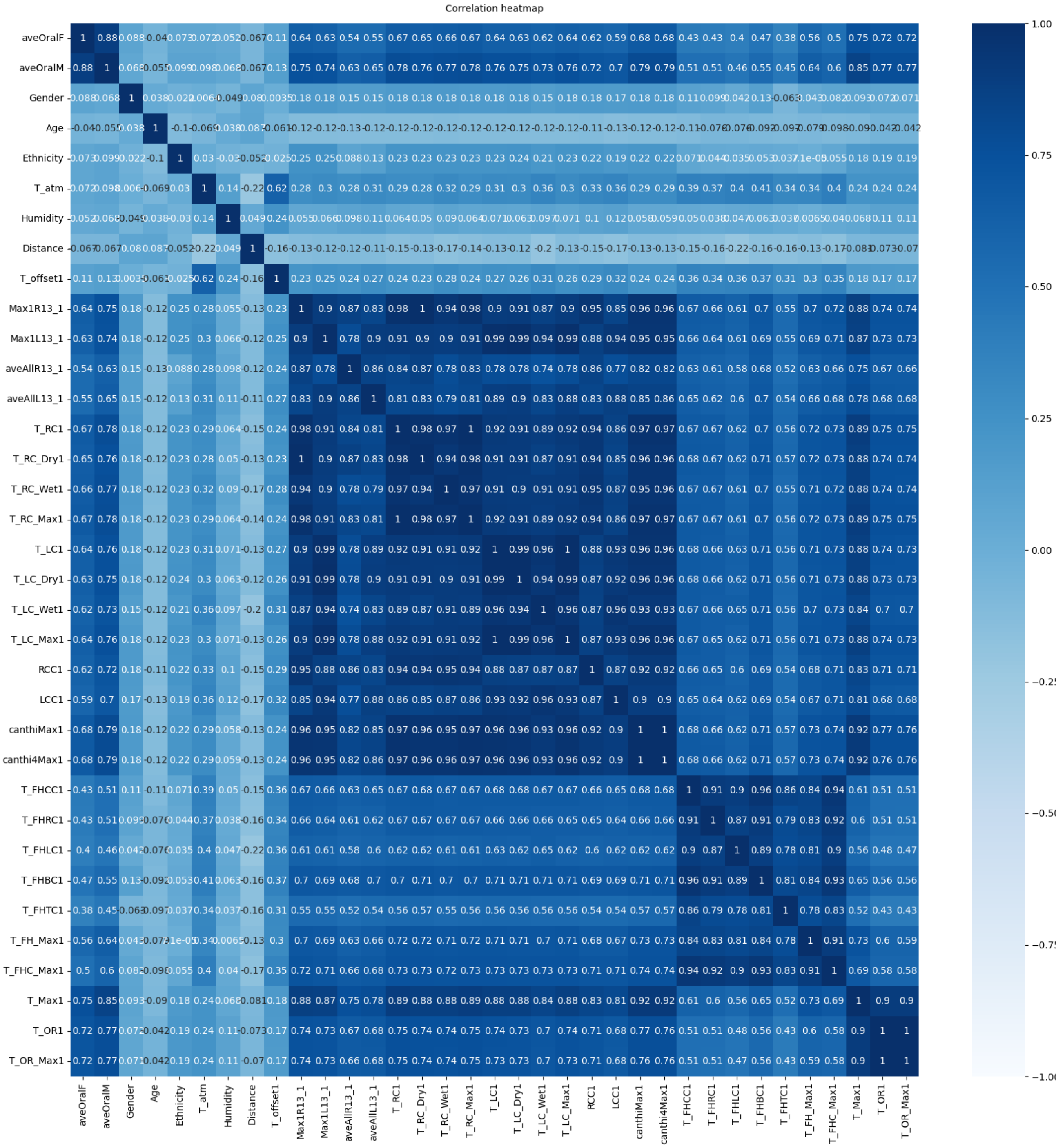


Figure 3: Correlation Heat Map for Multicollinearity in IRT Dataset

Figure 4: Histogram Diabetes Dataset

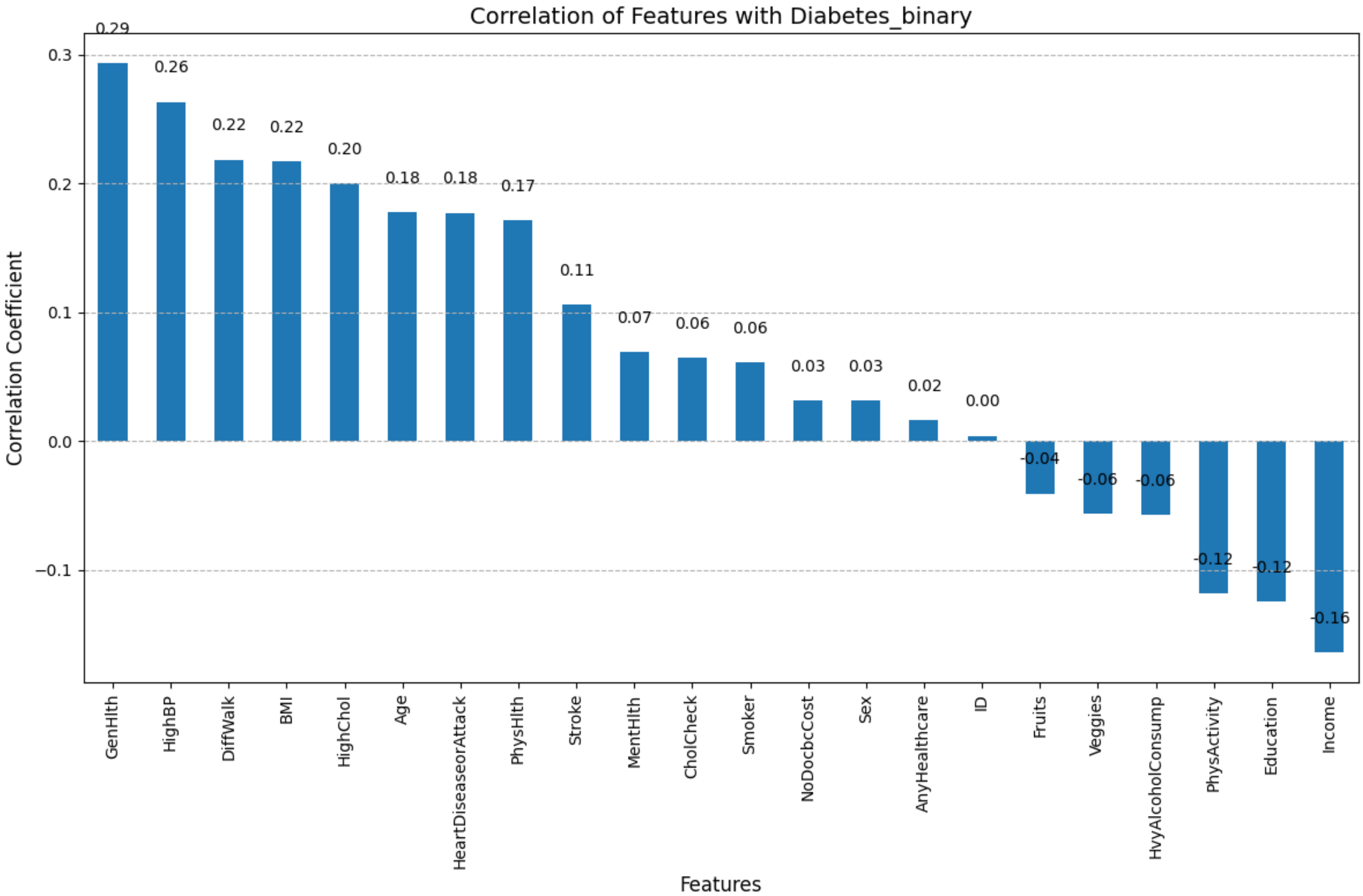


Figure 5: Correlation Coefficient vs Feature Diabetes Dataset

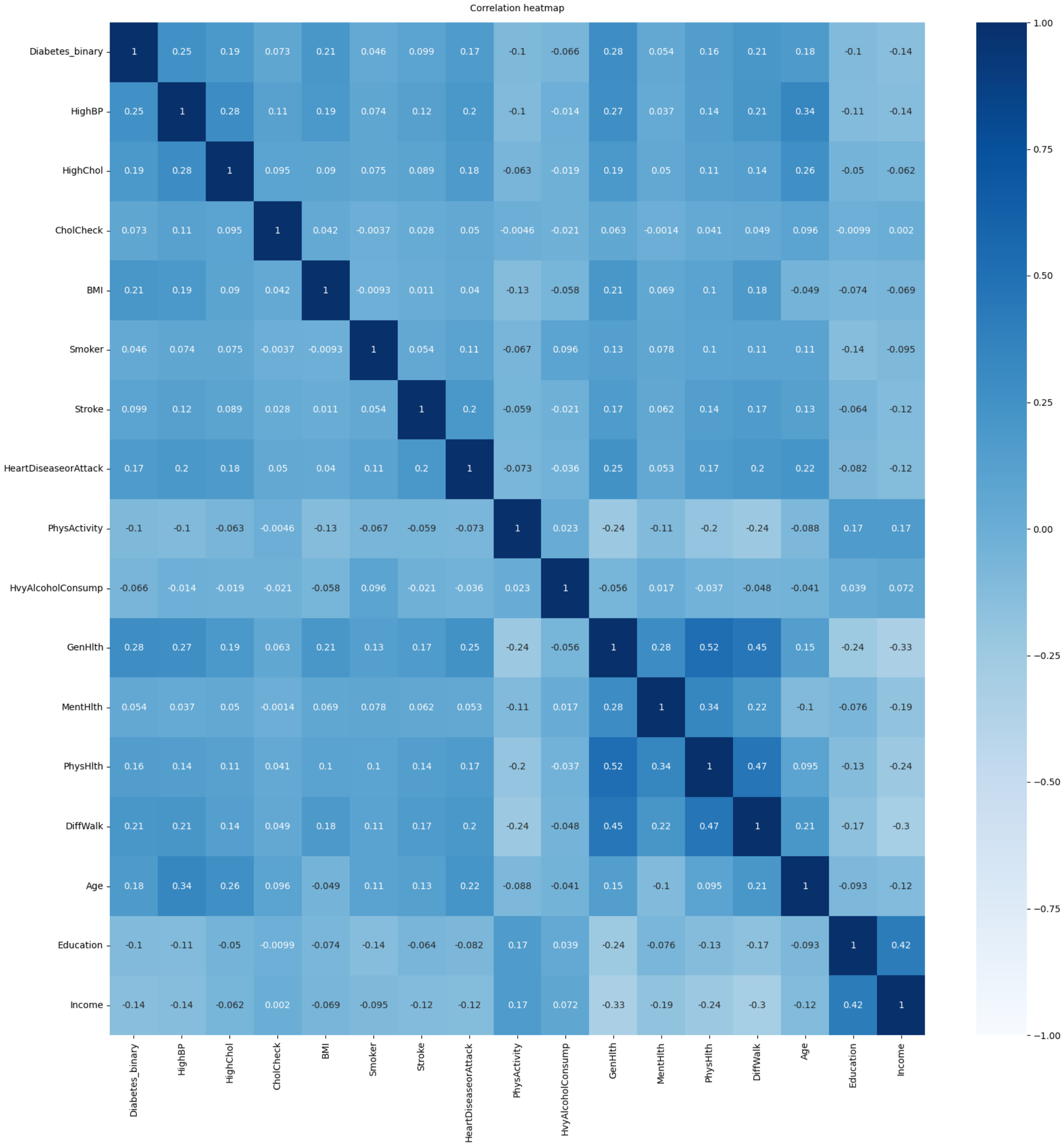
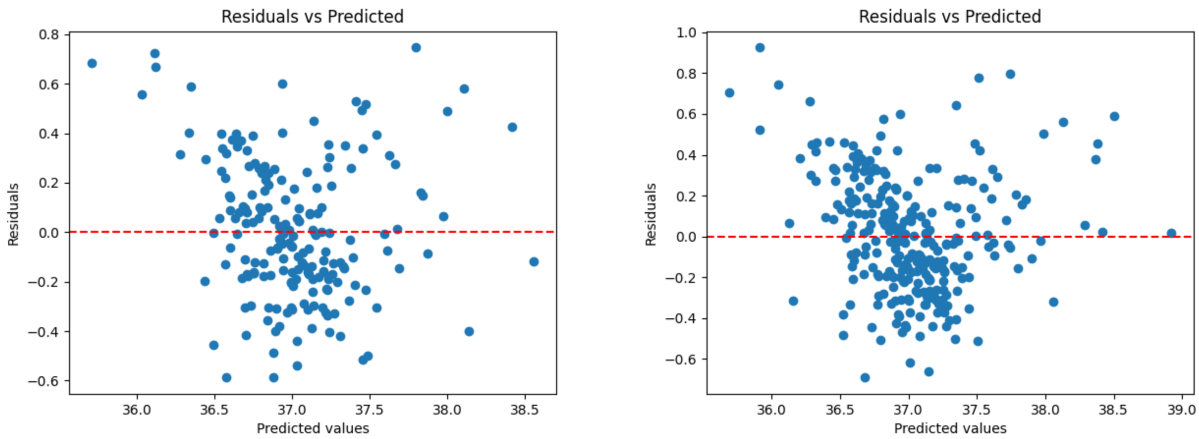
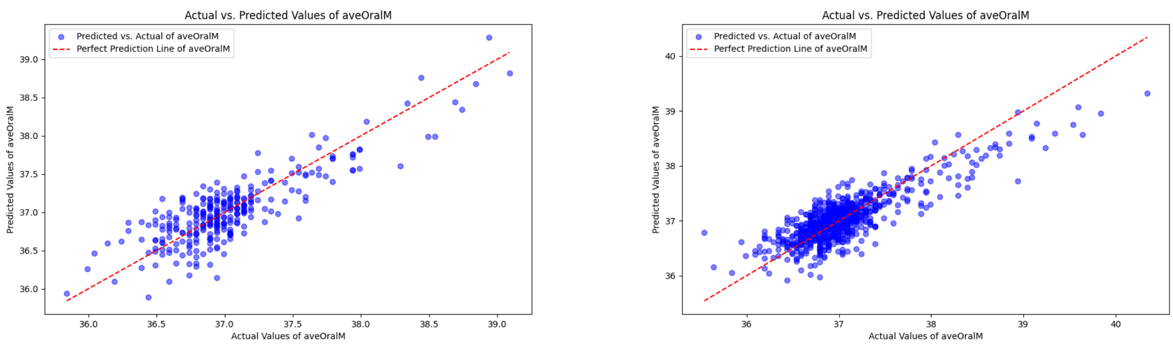


Figure 6: Correlation Heat Map for Multicollinearity in Diabetes Dataset



(a) Plot of Residual of Linear Regression (Testing) (b) Plot of Residual of Linear Regression (Training)

Figure 7: Comparing Residual Values from Linear Regression



(a) Actual and Predicted Values of Target aveOralM (Testing) (b) Actual and Predicted Values of Target aveOralM (Training)

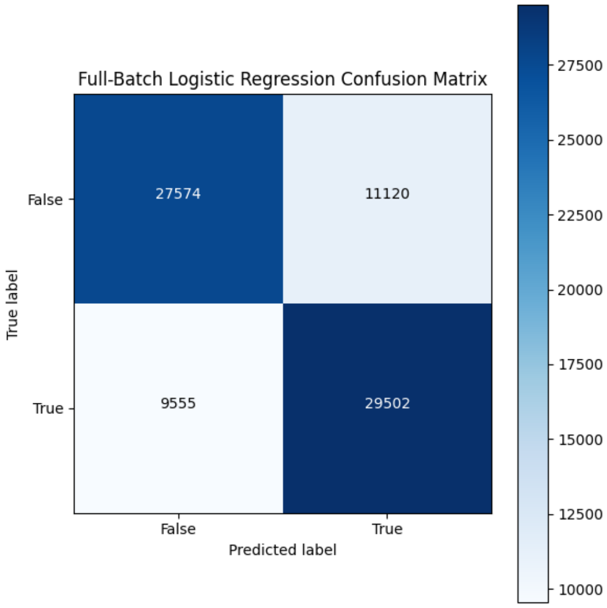
Figure 8: Comparing Actual and Predicted Values of Target aveOralM

Metric	Testing	Training
Mean Absolute Error	0.2279	0.2209
Mean Squared Error	0.0805	0.0827
R-squared	0.6441	0.7050
Adjusted R-squared	0.6379	0.7008

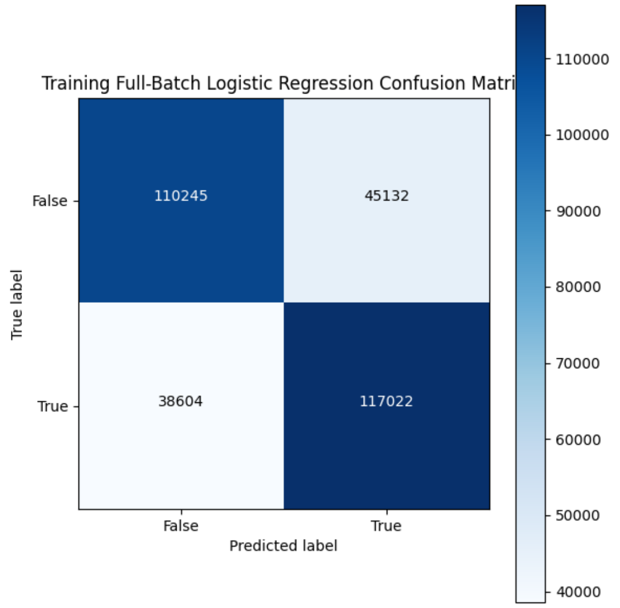
Table 6: Evaluation Metrics for the Model

Metric	Testing Dataset	Training Dataset
Accuracy	0.7307	0.7332
Precision	0.7206	0.7227
Recall	0.7520	0.7557
F1 Score	0.7360	0.7388

Table 7: Performance Metrics for 80-20 Logistic Regression

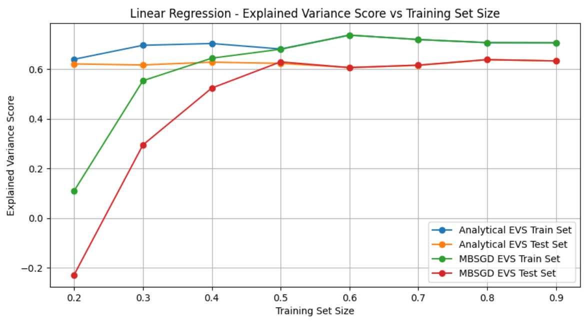


(a) Confusion Matrix for Full-Batch Logistic Regression (Testing)



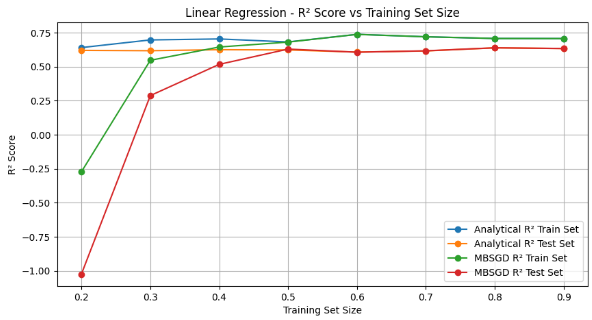
(b) Confusion Matrix for Full-Batch Logistic Regression (Training)

Figure 9: Confusion Matrix for 80-20 Training/Testing Dataset for Logistic Regression



(a) Explained Variance Score

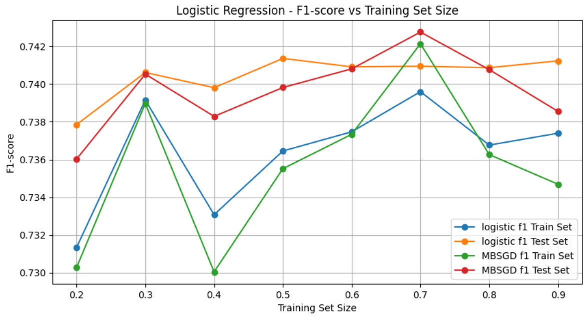
(b) Mean Squared Error



(c) R-squared

Figure 10: Performance Metrics of the Linear Regression Model with Increasing Training Dataset Size

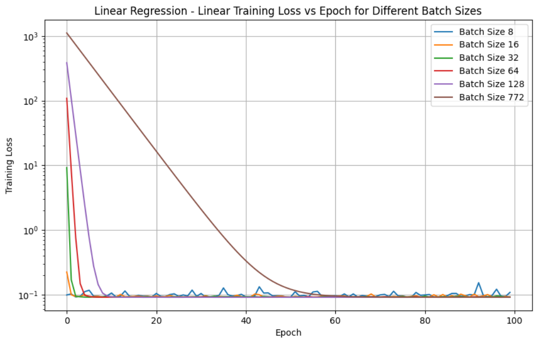
(a) Accuracy



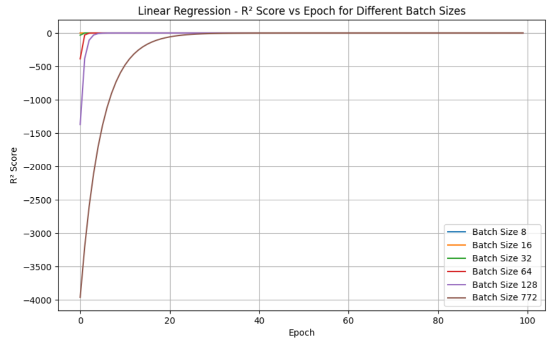
(b) F1 Score

(c) Precision

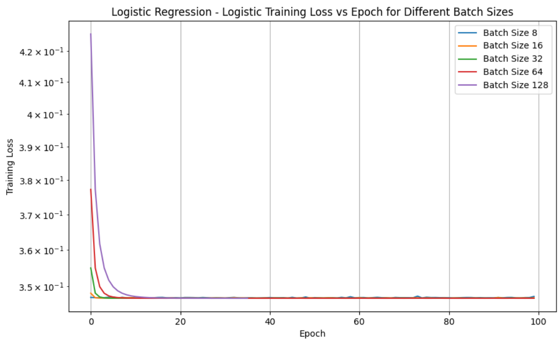
Figure 11: Performance Metrics of the Logistic Regression Model with Increasing Training Dataset Size



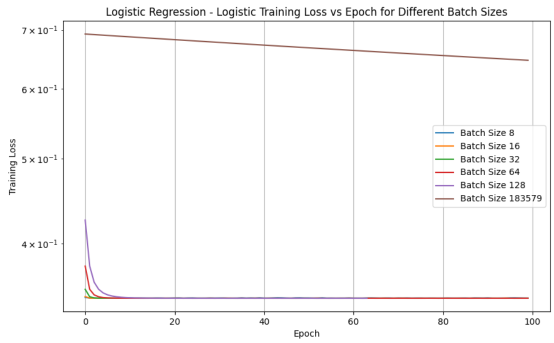
(a) Linear Regression - MSE vs Epoch



(b) Linear Regression - R^2 vs Epoch

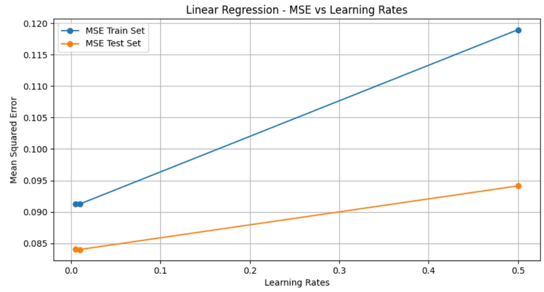


(c) Logistic Regression - Log-loss vs Epoch

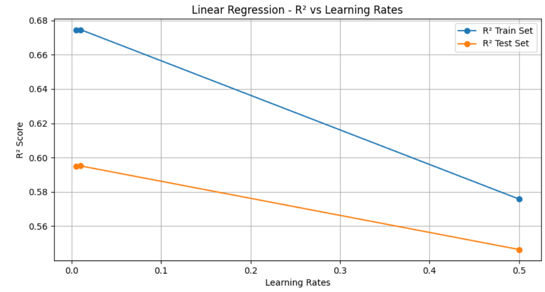


(d) Log-loss vs Epoch with full batch

Figure 12: Linear and Logistic Regression - Convergence Speed Using MSE, R^2 and Log-loss (cross-entropy for binary class)

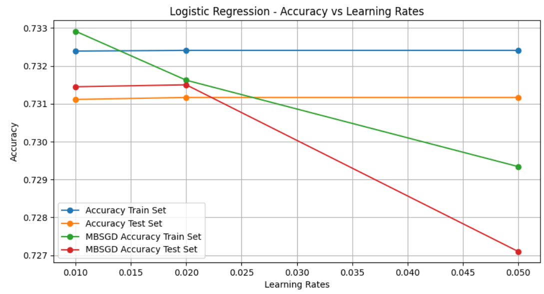


(a) MSE vs Learning Rates



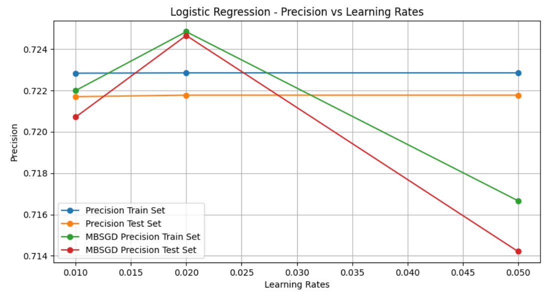
(b) R^2 vs Learning Rates

Figure 13: Linear Regression - Various Metrics vs Learning Rates



(a) Accuracy vs Learning Rates

(b) F1 Score vs Learning Rates



(c) Precision vs Learning Rates

(d) Recall vs Learning Rates

Figure 14: Logistic Regression - Various Metrics vs Learning Rates

Metric	Analytical	Mini-batch
Mean Absolute Error	0.2145	0.2143
Mean Squared Error	0.0722	0.0721
R-squared	0.6807	0.6811
Adjusted R-squared	0.66811	0.6685

Table 8: Compare Metrics for the Model

Method	learning_rate	max_iter	epsilon	batch_size	epoch	beta1	beta2
MBSGD - Logistic	0.01	1000	1×10^{-8}	32	32	N/A	N/A
MBSGD+Adam - Logistic	0.01	1000	1×10^{-8}	32	32	0.9	0.999
Full-batch Logistic	0.01	1000	1×10^{-8}	32	10	N/A	N/A
MBSGD - Linear	0.001	100	1×10^{-6}	32	100	N/A	N/A
MBSGD with Adam - Linear	0.001	100	1×10^{-6}	32	100	0.9	0.999

Table 9: Hyper-parameters for Linear and Logistic Regression

Method	**Compute Time (s)**	**Mean Absolute Error (MAE)**	**R-squared (R^2)**
Mini-Batch Stochastic Gradient Descent (MBSGD)	0.03153	0.2166	0.6902
MBSGD with Adam Optimization	1.1713	0.2194	0.6755
Analytical Regression	0.00222	0.2194	0.6755

Table 10: Metrics for Linear Regression Methods

Method	**Compute Time (s)**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Mini-Batch Stochastic Gradient Descent (MBSGD)	2.1783	0.7368	0.7251	0.7565	0.7405
MBSGD with Adam Optimization	21.2911	0.7122	0.7194	0.6890	0.7039
Full-Batch Logistic Regression	8.5703	0.7364	0.7240	0.7579	0.7406

Table 11: Metrics for Logistic Regression Methods