A Comparative Study of Classical and Modern Boosting Algorithms for Supervised Learning

Tony Luo, Will Kim, Sichen Li, Shang Peng, Jiaheng Guo*

Hi team, I am using the template provided by Prof. Zhong. We can for now ignore the references in .bib file and focus on this single .tex file for synchronization.

1 Introduction

Our code is available at: https://github.com/jiaheng-guo/stor565

2 Related Works

Review of existing studies in related fields.

3 Methodology

Please be aware that the paper structure Prof. Zhong provided in this template (Section 2 Related Works, Section 3 Methodology) is different from what we proposed in the group chat. Please move the mathematics explanations to Section 3, and reserve Section 2 for literature review, which covers the historical development of boosting algorithms, their applications, and comparisons in previous studies, with minimal mathematical details.

Describe the models, techniques, or algorithms you developed or compared.

---

*Please correspond to jiaheng@unc.edu for any questions regarding this paper and our code.

4 Experiments Results

This section describes the experimental setup used to compare AdaBoost, GBM, XGBoost, and LightGBM. All models were evaluated under a unified and reproducible framework with consistent preprocessing, hyperparameter tuning, and cross-validation procedures. Comparative analysis are then carried out, focusing on predictive performance, computational efficiency, and feature importance stability.

4.1 Datasets and Preprocessing

We evaluate the four boosting algorithms on several publicly available datasets from *Kaggle* and *UCI Machine Learning Repository* spanning 9 classification tasks and 4 regression tasks across various domains. Table 1 and 2 summary the characteristics of the datasets we used.

dataset	#samples	#features	#classes	domain	type	link
Adult Income	48,842	14	2	Social Science	Mixed	UCI
Heart Disease	303	13	2	Health	Mixed	UCI
Mushrooms	8,124	22	2	Biology	Categorical	Kaggle
Telco Customer Churn	7,043	33	2	Business	Mixed	Kaggle
Breast Cancer	569	30	2	Health	Numerical	UCI
Credit Card Fraud	284,807	30	2	Business	Numerical	Kaggle
IMDB Movie Reviews	49,582	-	2	Entertainment	NLP	Kaggle
MNIST	70,000	784	10	Computer Science	Image	UCI
HIGGS	11,000,000	28	2	Physics	Numerical	Kaggle

Table 1: Characteristics of the Classification Datasets Used

dataset	#samples	#features	domain	link
California Housing	20,640	8	Real Estate	Kaggle
Ames House Prices	1,460	80	Real Estate	Kaggle
Wine Quality	4,898	11	Business	UCI
Superconductivity	21,263	81	Physics	UCI

Table 2: Characteristics of the Regression Datasets Used

These classification and regression datasets vary in sample size, feature size, and subject domain, providing different scenarios for comprehensive evaluation of the boosting algorithms. The different data types involved, including numerical, categorical, images, texts, and time-series, also contribute to the diversity of the evaluation. Specifically, we included two datasets, IMDB Movie Reviews, which consists of text data of audience attitudes towards movies, and MNIST, the classical handwritten digits dataset, to examine the versatility of these algorithms on NLP and CV classification tasks. Moreover, the HIGGS dataset contains 11 million samples of particle collision data, making it a significant dataset for evaluating the performance of boosting algorithms on large-scale numerical data. The Credit Card Fraud dataset is highly imbalanced, with only 492 frauds out of 284,807 transactions. All datasets were first processed through the following shared pipeline and were subsequently adjusted as needed for each dataset:

(1) Missing numerical features were imputed using medians; missing categorical features were imputed using the most frequent category.

(2) Categorical variables were one-hot encoded for AdaBoost, GBM, and XGBoost, while LightGBM additionally took advantage of its built-in categorical support.

(3) Numerical features were left unscaled, consistent with tree-based model characteristics.

(4) Training and testing splits were generated using fixed random seeds with an 80/20 ratio.

We further processed the NLP and CV datasets. For IMDB Movie Reviews, we utilized TF-IDF vectorization to convert text data into numerical features suitable for boosting algorithms. For MNIST, we flattened the 28x28 pixel images into 784-dimensional vectors. For MNIST, we adopted the following preprocessing strategies: we first flattened the 28x28 pixel images into 784-dimensional vectors, then we applied PCA to reduce dimensionality while retaining 95% variance.

4.2 Hyperparameter Tuning and Evaluation Metrics

Hyperparameter optimization for each boosting algorithm was performed using *Bayesian Optimization*, which adaptively explores the hyperparameter space by modeling the relationship between hyperparameters and validation performance. The search focused on the key parameters that most strongly influence boosting performance, including maximum tree depth, number of estimators, and learning rate. For classification tasks, logistic or exponential loss functions were used as appropriate, while regression models employed squared-error loss. To prevent information leakage and ensure unbiased estimates, we adopted a nested 5-fold cross-validation framework, where the inner loop performed Bayesian optimization and the outer loop produced final evaluation scores.

For classification problems, we evaluated model performance using accuracy, F1-score, and ROC–AUC, with the one-vs-rest formulation applied to multiclass settings. For regression tasks, we reported RMSE, MAE, and R^2. All evaluations were conducted using 5-fold cross-validation with fixed random seeds to ensure reproducibility, and deterministic algorithm settings were used when supported.

Additionally, we measured training time and inference latency across datasets of varying scales. Finally, we extracted feature-importance rankings from each model to examine the consistency and stability of feature attributions across folds and candidate algorithms.

4.3 Comparative Analysis on Classification Tasks

In this subsection, we present the comparative results of AdaBoost, GBM, XGBoost, and LightGBM on the classification datasets described in Section 4. The AUC scores across different datasets and algorithms are summarized in Figure 1.
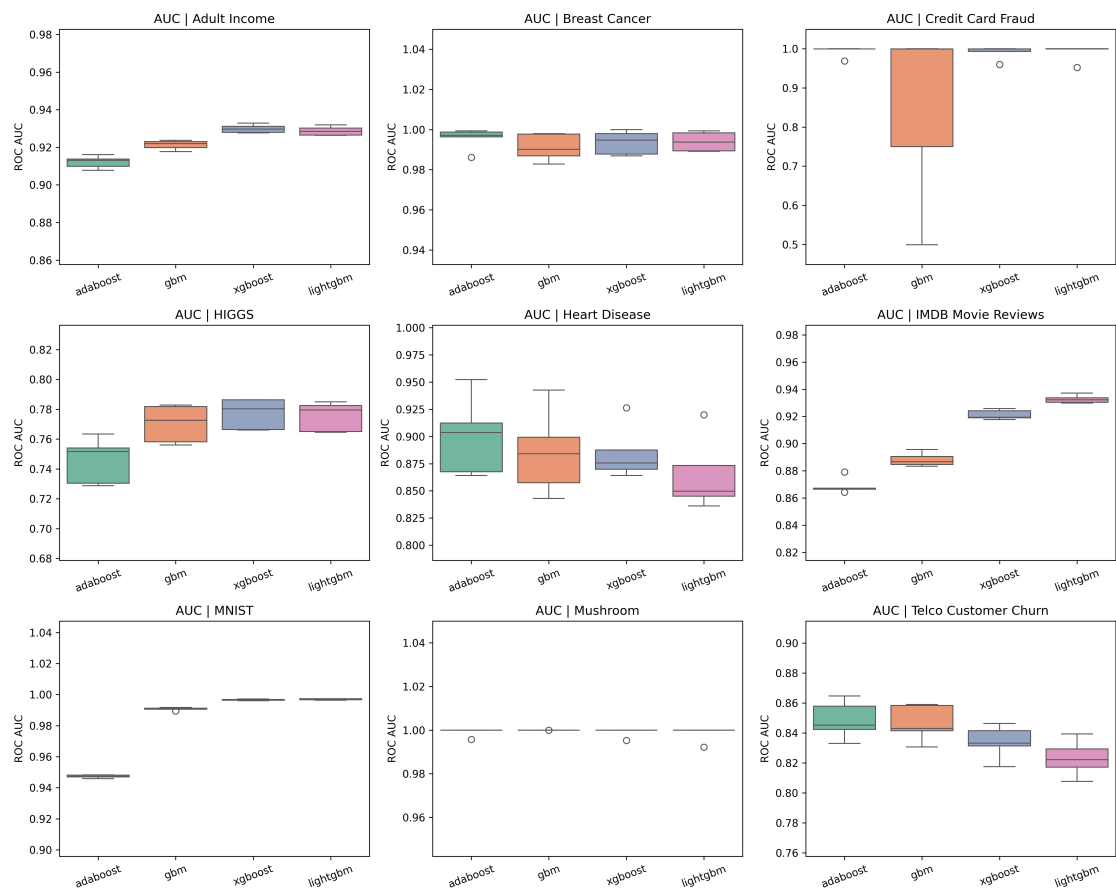

Figure 1: AUC Comparison Across Boosting Algorithms

From the results, we observe that modern boosting algorithms generally outperform their classical counterparts across most datasets, while there is no single algorithm that consistently dominates all others. Notably, XGBoost and LightGBM exhibit significant improvements in AUC scores on larger datasets such as HIGGS and Credit Card Fraud, likely due to their optimized handling of large-scale data through parallelization and histogram-based split finding. On smaller datasets like Heart Disease and Breast Cancer, the performance gap narrows, suggesting that the advantages of modern algorithms may be more pronounced in high-dimensional or large-sample scenarios. Results in terms of the other metrics (accuracy and F1-score) follow similar trends, due to space limitations, we only present AUC results here, please refer to Appendix A for complete results.

In terms of computational efficiency, LightGBM consistently demonstrates the fastest training times across datasets, attributed to its histogram-based approach and leaf-wise tree growth strategy. XGBoost also shows considerable speed improvements over GBM, particularly on larger datasets. AdaBoost, while competitive on smaller datasets, tends to lag in training speed as data size increases. The training time comparison is illustrated in Figure 2.

4.4 Comparative Analysis on Regression Tasks

5 Conclusion and Future Direction

[1]

References

[1] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
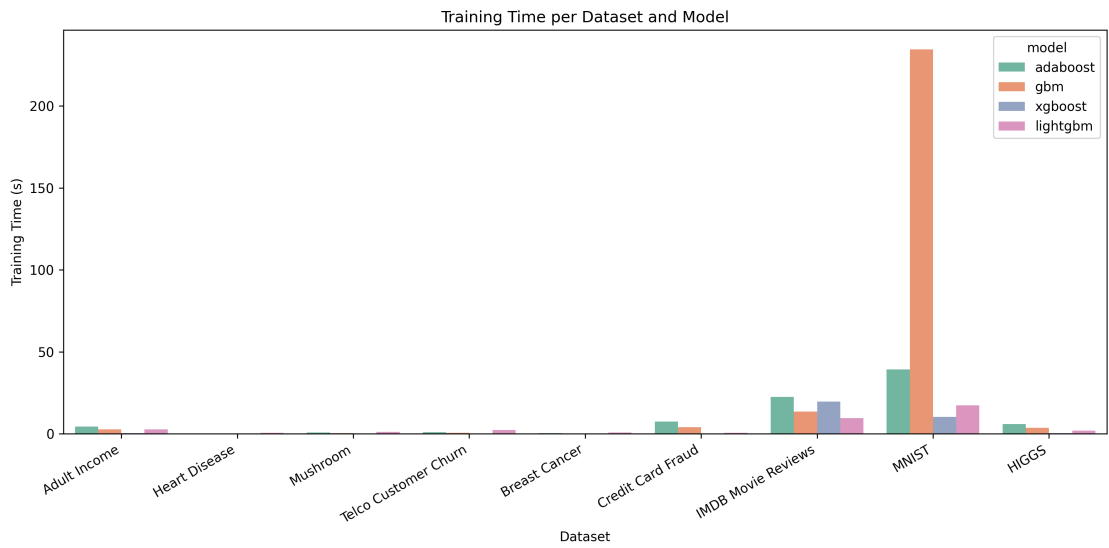
Figure 2: Training Time Comparison Across Boosting Algorithms [To Be Updated]

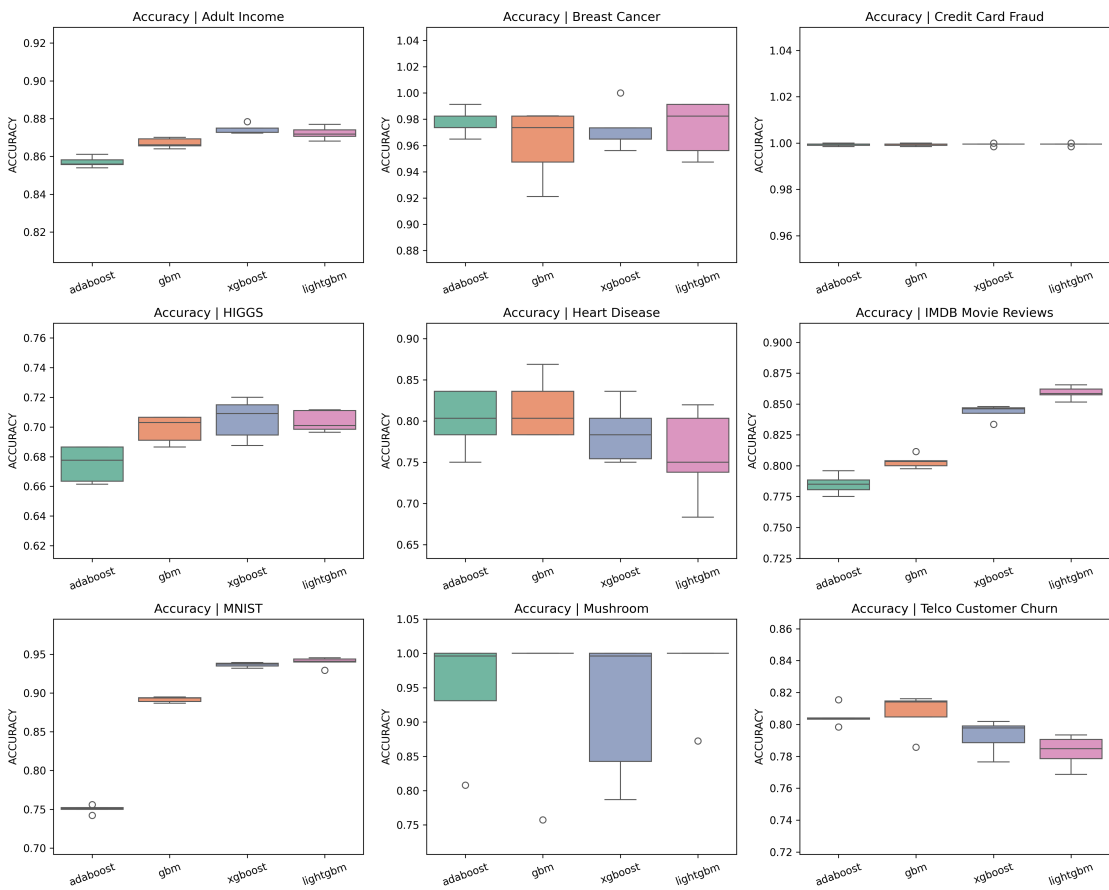# A   Additional Experimental Results
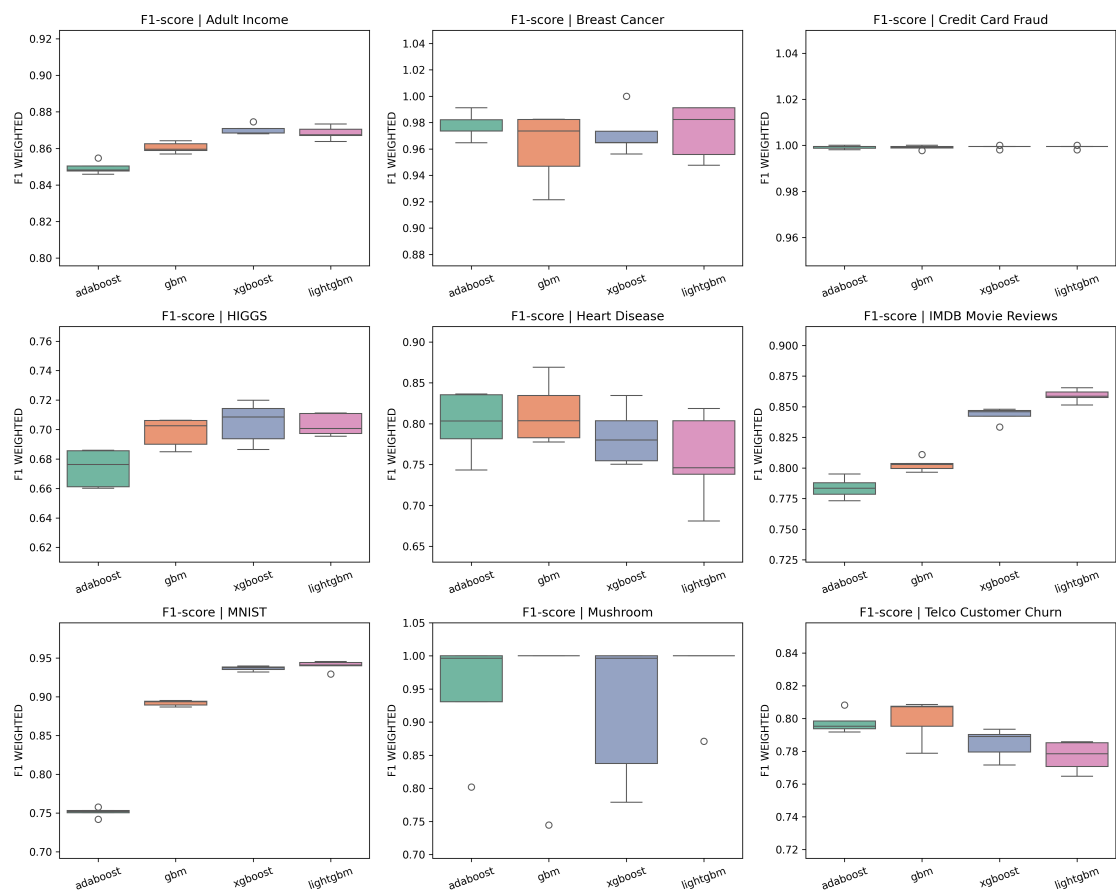

Figure 3: Accuracy Comparison Across Boosting Algorithms

Figure 4: F1-score Comparison Across Boosting Algorithms