Journal of Computer Science and Technology
Volume 4 ; Number 3 ; September 2026 ; Page 268-283
DOI : https://doi.org/10.59435/jocstec.v4i3.764
https://jurnal.padangtekno.com/index.php/jocstec
E-ISSN : 2985-4318
P-ISSN : 2985-5772

Comparative Analysis Of Ensemble Learning Algorithms For Student Mental Health Classification: Balancing Predictive Performance And Computational Efficiency

Lia Umbari Putri[1*], Laina Tussifah Santoso[2], Masri Wahyuni[3]

[1]Computer Engineering, Akademi Manajemen Informatika dan Komputer Polibisnis
[2,3]Computerized Accounting, Akademi Manajemen Informatika dan Komputer Polibisnis
[1*]liaumbariputri@email.com

ABSTRACT

Student mental health disorders have become an increasingly important issue due to their significant impact on academic performance and psychological well-being. Machine learning-based approaches offer a promising alternative for supporting early detection; however, only a limited number of studies have conducted comprehensive evaluations of various ensemble learning algorithms by simultaneously considering predictive performance, computational efficiency, and statistical significance. This study aims to compare ten classification algorithms, namely Random Forest, Extra Trees, Gradient Boosting, Hist Gradient Boosting, AdaBoost, XGBoost, LightGBM, CatBoost, Logistic Regression, and Stacking Ensemble, using 5-fold Stratified Cross-Validation. Model performance was evaluated using Accuracy, Precision, Recall, F1-Score, Receiver Operating Characteristic–Area Under the Curve (ROC-AUC), training time, inference time, confusion matrix, feature importance, as well as One-Way Analysis of Variance (ANOVA) and post-hoc Tukey's Honestly Significant Difference (Tukey HSD) tests. The experimental results show that CatBoost achieved the highest classification performance, with an Accuracy of 84.86%, an F1-Score of 87.19%, a Recall of 88.03%, and an ROC-AUC of 0.9220. XGBoost delivered comparable predictive performance with a substantially shorter training time (3.92 seconds), while LightGBM was the most computationally efficient algorithm, requiring an average training time of only 0.85 seconds. In contrast, the Stacking Ensemble required the longest training time (215.41 seconds) without providing a statistically meaningful performance improvement over the boosting-based models. The ANOVA results revealed significant differences in ROC-AUC among the evaluated models ($p < 0.001$), whereas the Tukey HSD test indicated that most boosting algorithms did not differ significantly from one another. These findings provide practical guidance for selecting classification algorithms by balancing predictive performance and computational efficiency to support early mental health screening systems for university students.

Keywords : ensemble learning, student mental health, classification, machine learning, Stratified Cross-Validation

INTRODUCTION

Mental health disorders represent a major challenge in higher education because they adversely affect students' academic performance, quality of life, and ability to continue their studies. Academic pressure, financial difficulties, heavy workloads, sleep patterns, and social circumstances are among the factors known to increase the risk of depression among university students [1], [2]. Early detection is essential for identifying at-risk students and enabling timely intervention before

their conditions become more severe. However, conventional psychological assessments require considerable time, resources, and professional involvement, making them inefficient for large student populations [1]. Therefore, a machine-learning-based approach capable of providing rapid, accurate, and automated classification is needed to support the early detection process.

Advances in machine learning have made significant contributions to various healthcare applications, including the prediction of mental health disorders. Baba and Bunji [1] demonstrated that machine learning models could predict mental health problems among university students using annual health survey data, achieving promising performance. Faisti et al. [3] compared Naïve Bayes and Random Forest algorithms for mental health classification, while Febriani et al. [4] applied Support Vector Machine and Random Forest models to analyze the psychological effects of social media exposure on adolescents. These findings demonstrate the ability of machine-learning-based approaches to identify complex patterns in psychological data. Nevertheless, their performance remains strongly influenced by the characteristics of the data and the algorithms employed.

In recent years, ensemble learning algorithms have gained increasing attention because they can improve generalizability by combining multiple learning models. Algorithms such as Gradient Boosting, XGBoost, LightGBM, and CatBoost have demonstrated competitive performance across various classification and prediction tasks [5]-[8]. XGBoost provides efficient optimization through a gradient tree boosting approach and has consequently become one of the most widely used algorithms in machine learning competitions [5]. LightGBM offers high computational efficiency through histogram-based learning [6], while CatBoost can effectively process categorical features without requiring complex encoding procedures [7]. In addition, a Stacking Ensemble combines the predictions of multiple base learners through a meta-model, potentially producing more stable and accurate predictions [9], [10]. Previous studies have also demonstrated the effectiveness of boosting algorithms in diverse domains, including electrical load forecasting, financial transaction fraud detection, and disease-risk prediction [8], [10], [11].

Despite these advances, previous studies have generally evaluated only one or two algorithms, thus providing an incomplete understanding of the relative strengths of different machine learning and ensemble learning algorithms for classifying depression among university students. Furthermore, most studies have reported only standard classification metrics such as accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic curve (ROC-AUC) without considering computational efficiency, inference time, feature importance, or statistical evaluation of performance differences across models. Consequently, relatively small differences in performance are often used directly to select the best-performing model without determining whether these differences are statistically significant [12]-[14].

To address these gaps, this study proposes a comprehensive evaluation framework for comparing machine learning and ensemble learning algorithms in the classification of depression among university students. Unlike most previous studies, which have primarily focused on comparisons of standard classification metrics, the proposed framework integrates predictive-performance evaluation, computational-efficiency assessment, model interpretation through feature-importance analysis, examination of misclassification patterns using confusion matrices, and statistical significance testing using one-way analysis of variance (ANOVA) followed by Tukey's honestly significant difference (Tukey HSD) post hoc test within a consistent evaluation protocol. The framework is applied to compare ten classification algorithms: Random Forest, Extra Trees, Gradient Boosting, Histogram-Based Gradient Boosting, AdaBoost, XGBoost, LightGBM, CatBoost, Logistic Regression, and a Stacking Ensemble. Five-fold stratified cross-validation is employed to ensure that all models are evaluated under identical experimental conditions. Through this approach, the study not only identifies the algorithm with the strongest predictive performance but also evaluates the trade-offs among classification accuracy, discriminative ability, computational efficiency, and statistical significance, thereby providing a robust basis for selecting a model to support the early detection of depression among university students.

RESEARCH METHODOLOGY

A. Research Stages

This study employs a quantitative experimental approach to compare the predictive performance and computational efficiency of several ensemble learning algorithms in classifying the mental health status of university students. The research design comprises seven key stages: dataset acquisition, examination of data characteristics, data preprocessing, model development, evaluation using five-fold stratified cross-validation, measurement of computational efficiency, and statistical analysis and interpretation of results. This structured workflow ensures that the experiment is conducted systematically and consistently, while also facilitating reproducibility.

The research workflow implemented in this study is presented in Figure 1. The research began with the acquisition of the *Student Depression Dataset*, which contains information regarding students' demographic characteristics, academic status, lifestyle, and psychological condition. This dataset comprises 27,901 observations and 18 attributes, consisting of 17 input features and one target variable. The target variable is binary: class 0 represents students without depression, while class 1 represents students with depression.

Figure 2 displays the class distribution prior to the application of the *Synthetic Minority Over-sampling Technique* (SMOTE). The dataset contains approximately 16,300 observations in the depression class and 11,600 observations in the non-depression class, resulting in a class imbalance ratio of approximately 1.41:1. This distribution indicates a moderate level of class imbalance, with the depression class comprising a larger proportion of observations than the non-depression class. Consequently, SMOTE was applied specifically to the training data in each fold to balance the class distribution while preventing data leakage into the held-out test data.

Figure 1. Research workflow for comparing ensemble learning methods in university student mental health classification

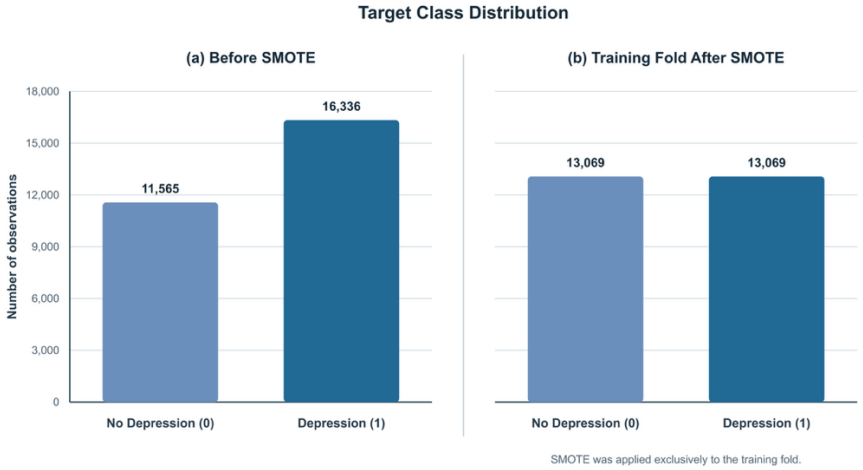

Figure 2. Target distribution of the dataset

Subsequently, we conducted Exploratory Data Analysis (EDA) including class distribution analysis and missing value identification to ascertain the inherent data characteristics prior to modeling. The ensuing data preprocessing phase encompassed data cleaning, categorical feature encoding, and feature scaling. To mitigate class imbalance without introducing data leakage, we applied the Synthetic Minority Over-sampling Technique (SMOTE) exclusively to the training subset within each cross-validation fold. Following preprocessing, we employed 5-Fold Stratified Cross-Validation to evaluate the models, ensuring each algorithm was systematically trained and tested across balanced data distributions.

During the model development phase, this study compared ten classification algorithms: Random Forest, Extra Trees, Gradient Boosting, Histogram-based Gradient Boosting, AdaBoost, XGBoost, LightGBM, CatBoost, Logistic Regression, and a Stacking Ensemble. Finally, we comprehensively assessed the classification performance of all models using Accuracy, Precision, Recall, F1-Score, and ROC-AUC metrics, supplemented by confusion matrix analysis.

Table 1. Algorithms evaluated in this study

No	Group	Model
1	Bagging	Random Forest
2	Bagging	Extra Trees
3	Boosting	AdaBoost
4	Boosting	Gradient Boosting
5	Boosting	Hist Gradient Boosting
6	Boosting	XGBoost
7	Boosting	LightGBM
8	Boosting	CatBoost
9	Heterogeneous Ensemble	Stacking Ensemble
10	Baseline	Logistic Regression

In addition to predictive performance, this study evaluates computational efficiency by measuring training and inference times. Furthermore, we conducted a statistical analysis using One-way ANOVA, followed by the Tukey HSD post-hoc test, to identify significant performance variations among the models. To enhance the interpretability of the findings, we performed a feature importance analysis on models that inherently support interpretability—namely Random Forest, XGBoost, and CatBoost.

In the final stage, we conducted a comprehensive comparative analysis and discussion, comparing predictive performance, computational efficiency, misclassification patterns, and prediction probability correlations across the models. Based on these combined evaluation results, we

determined which algorithm offers the optimal balance between predictive accuracy and computational efficiency for implementation in a student mental health screening system.

B. Model Evaluation Metrics

Classification performance was evaluated using accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC). Accuracy indicates the overall proportion of correctly predicted instances, whereas precision measures the proportion of correctly identified positive predictions. Recall reflects the model's capacity to identify actual positive samples, while the F1-score represents the harmonic mean of precision and recall. Finally, ROC-AUC assesses the model's ability to distinguish between positive and negative classes across various classification thresholds.

Accuracy is calculated using Equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision is calculated using Equation:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall is calculated using Equation:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

The F1-score is calculated using Equation:

$$F1 - Score = \frac{Precision\ x\ Recall}{Precision + Recall} \quad (4)$$

In the aforementioned equations, TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. Beyond the aggregate performance metrics, a confusion matrix is employed to visualize the classification error patterns of each model.

C. Computational Efficiency Evaluation

We analyzed computational efficiency using two primary indicators: training time and inference latency. Training time was measured in seconds per fold, encompassing the entire period from the initiation of the training process to model completion. Inference latency was calculated in milliseconds per sample, representing the time required for the model to generate predictions on the test set. We incorporated these metrics because models exhibiting high predictive performance are not necessarily efficient for real-world deployment in web-based screening systems. Consequently, our analysis extended beyond mere accuracy or F1-score rankings to critically evaluate the trade-off between predictive quality and computational overhead. The experimental results demonstrated that the variance in training times was substantially more pronounced than the performance differentials among the models, particularly when comparing LightGBM, XGBoost, CatBoost, and the Stacking Ensemble.

D. Confusion Matrix and Prediction Probability Analysis

We constructed a confusion matrix for each model using predictions generated across all folds to analyze the distribution of true positives, true negatives, false positives, and false negatives. In the context of mental health screening, we paid particular attention to false negatives, as these cases represent students who were actually at risk but went undetected by the model. Additionally, we calculated Pearson correlation coefficients between predicted probabilities to measure the degree of output similarity across models. A correlation value close to 1.0 indicates that two models produce highly aligned probability patterns. We utilized this analysis to evaluate predictive diversity among the base learners, providing valuable insight into why the Stacking Ensemble did not consistently yield performance improvements over the best-performing single model.

E. Feature Importance Analysis

We conducted a feature importance analysis on models that inherently support feature-contribution-based interpretability specifically Random Forest, XGBoost, and CatBoost. We extracted the ten most significant features to identify the variables exerting the greatest influence on classification outcomes. Given that each algorithm employs different mechanisms for calculating importance and representing features, we avoided directly comparing absolute importance scores across models. Instead, our analysis focused on the consistent appearance of top-ranked features across these algorithms.

F. Statistical Analysis

We conducted a statistical analysis of the ROC-AUC scores from the five cross-validation folds to evaluate whether there were significant performance differences among the models. Initial testing employed a One-way Analysis of Variance (ANOVA) at a significance level of $\alpha = 0.05$. Upon detecting a statistically significant difference, we subsequently applied Tukey's Honestly Significant Difference (HSD) post-hoc test to identify the specific model pairs exhibiting differences in mean ROC-AUC scores.

RESULTS AND DISCUSSION

A. Evaluation of Classification Performance

We evaluated ten machine learning algorithms: Random Forest, Extra Trees, Gradient Boosting, Histogram-based Gradient Boosting, AdaBoost, XGBoost, LightGBM, CatBoost, a Stacking Ensemble, and Logistic Regression, which served as a non-ensemble baseline. We assessed all models using 5-Fold Stratified Cross-Validation to preserve the target class proportions within each fold. Model performance was quantified using accuracy, precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC). We report the results for each metric as the mean and standard deviation to illustrate both overall predictive performance and inter-fold consistency.

Table 2. Model evaluation results using 5-fold stratified cross-validation

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
AdaBoost	0,8472 ± 0,0021	0,8731 ± 0,0028	0,8647 ± 0,0022	0,8689 ± 0,0017	**0,9221 ± 0,0012**
CatBoost	**0,8486 ± 0,0027**	0,8638 ± 0,0032	**0,8803 ± 0,0020**	**0,8719 ± 0,0021**	0,9220 ± 0,0012
Extra Trees	0,8142 ± 0,0052	0,8323 ± 0,0044	0,8549 ± 0,0050	0,8434 ± 0,0044	0,9038 ± 0,0024
Gradient Boosting	0,8464 ± 0,0036	0,8629 ± 0,0042	0,8770 ± 0,0037	0,8699 ± 0,0030	0,9206 ± 0,0016
Hist Gradient Boosting	0,8460 ± 0,0027	0,8624 ± 0,0035	0,8770 ± 0,0018	0,8696 ± 0,0021	0,9201 ± 0,0017
LightGBM	0,8454 ± 0,0016	0,8605 ± 0,0012	0,8782 ± 0,0035	0,8693 ± 0,0016	0,9185 ± 0,0024
Logistic Regression	0,8440 ± 0,0033	**0,8785 ± 0,0028**	0,8514 ± 0,0051	0,8647 ± 0,0031	0,9211 ± 0,0010
Random Forest	0,8300 ± 0,0029	0,8515 ± 0,0037	0,8596 ± 0,0085	0,8555 ± 0,0031	0,9105 ± 0,0012
Stacking Ensemble	0,8471 ± 0,0030	0,8706 ± 0,0036	0,8680 ± 0,0037	0,8693 ± 0,0026	0,9218 ± 0,0011
XGBoost	0,8483 ± 0,0021	0,8635 ± 0,0031	0,8800 ± 0,0031	0,8717 ± 0,0017	0,9212 ± 0,0016

The results in Table 2 show that CatBoost achieved the highest overall classification performance across three key metrics, with an accuracy of 0.8486 ± 0.0027, recall of 0.8803 ± 0.0020, and F1-score of 0.8719 ± 0.0021. XGBoost produced a highly comparable performance, with an accuracy of 0.8483 ± 0.0021, recall of 0.8800 ± 0.0031, and F1-score of 0.8717 ± 0.0017. The difference in F1-score between CatBoost and XGBoost was only 0.0002, indicating that CatBoost's numerical advantage was relatively marginal. AdaBoost achieved an accuracy of 0.8472 ± 0.0021 and an F1-score of 0.8689 ± 0.0017. Although its recall was lower, at 0.8647 ± 0.0022, AdaBoost

obtained a precision of 0.8731 ± 0.0028 and the highest ROC-AUC value, at 0.9221 ± 0.0012. This value was slightly higher than those of CatBoost, which achieved a ROC-AUC of 0.9220 ± 0.0012, and the Stacking Ensemble, which achieved 0.9218 ± 0.0011. These results indicate that the model with the highest F1-score was not necessarily the model with the highest ROC-AUC.

Gradient Boosting and Histogram-Based Gradient Boosting exhibited similar performance patterns. Gradient Boosting achieved an accuracy of 0.8464 ± 0.0036, recall of 0.8770 ± 0.0037, F1-score of 0.8699 ± 0.0030, and ROC-AUC of 0.9206 ± 0.0016. Histogram-Based Gradient Boosting obtained an accuracy of 0.8460 ± 0.0027, recall of 0.8770 ± 0.0018, F1-score of 0.8696 ± 0.0021, and ROC-AUC of 0.9201 ± 0.0017. Both models demonstrated relatively high and stable sensitivity, although neither surpassed CatBoost or XGBoost in terms of F1-score. LightGBM achieved an accuracy of 0.8454 ± 0.0016, recall of 0.8782 ± 0.0035, F1-score of 0.8693 ± 0.0016, and ROC-AUC of 0.9185 ± 0.0024. Although LightGBM's ROC-AUC was lower than that of the leading boosting models, its recall was the third highest after CatBoost and XGBoost. The relatively small standard deviations in its accuracy and F1-score also indicate consistent performance across folds. The Stacking Ensemble achieved an accuracy of 0.8471 ± 0.0030, precision of 0.8706 ± 0.0036, recall of 0.8680 ± 0.0037, F1-score of 0.8693 ± 0.0026, and ROC-AUC of 0.9218 ± 0.0011. Although this performance placed the Stacking Ensemble among the top-performing models, it did not outperform CatBoost or XGBoost as individual models. Specifically, the Stacking Ensemble obtained the same F1-score as LightGBM, at 0.8693, but remained below CatBoost at 0.8719 and XGBoost at 0.8717.

In contrast, the bagging-based models produced lower performance. Random Forest achieved an accuracy of 0.8300 ± 0.0029, recall of 0.8596 ± 0.0085, F1-score of 0.8555 ± 0.0031, and ROC-AUC of 0.9105 ± 0.0012. Extra Trees yielded the lowest results, with an accuracy of 0.8142 ± 0.0052, recall of 0.8549 ± 0.0050, F1-score of 0.8434 ± 0.0044, and ROC-AUC of 0.9038 ± 0.0024. These findings reveal a more pronounced performance gap between the bagging-based models and most boosting-based models. Logistic Regression achieved an accuracy of 0.8440 ± 0.0033, the highest precision of 0.8785 ± 0.0028, recall of 0.8514 ± 0.0051, F1-score of 0.8647 ± 0.0031, and ROC-AUC of 0.9211 ± 0.0010. Its high precision combined with lower recall suggests that the model was more selective in assigning positive predictions. However, this result should be interpreted with caution because the Logistic Regression training process previously generated a warning indicating that the maximum convergence criterion had not been reached.

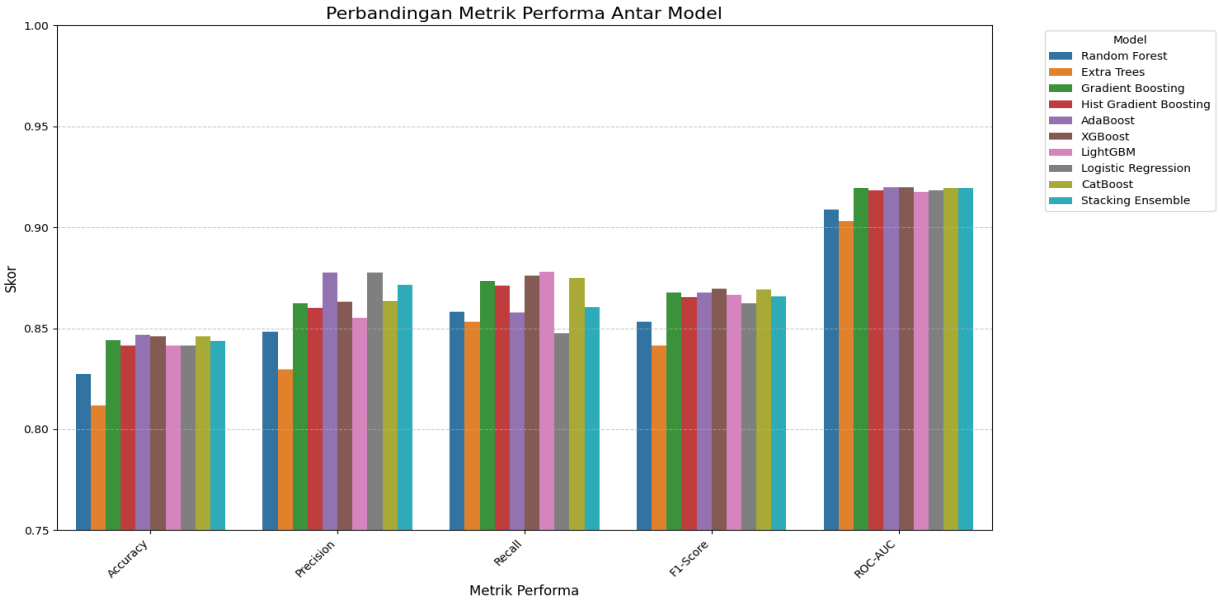

Figure 3. Comparison of accuracy, precision, recall, F1-score, and ROC-AUC across models

The visualization in Figure 3 shows that the performance differences among CatBoost, XGBoost, Gradient Boosting, Histogram-Based Gradient Boosting, AdaBoost, LightGBM, and the

Stacking Ensemble were relatively small. More pronounced differences were observed for Random Forest and Extra Trees, particularly in terms of accuracy, F1-score, and ROC-AUC. The graph also indicates variation in metric orientation across models. CatBoost and XGBoost performed better in terms of recall and F1-score, whereas AdaBoost and Logistic Regression achieved relatively high precision

B. ROC-AUC Curve Analysis

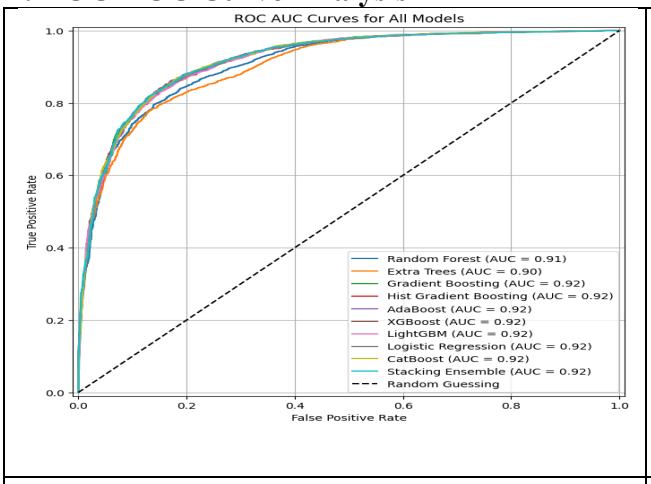

Figure 4. ROC-AUC curves for all models

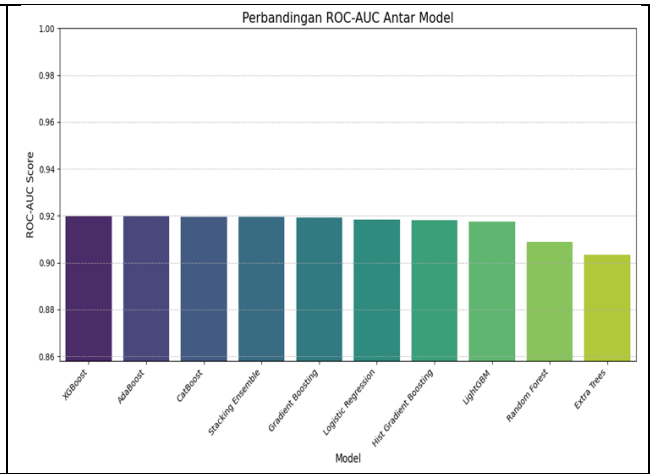

Figure 5. Comparison of ROC-AUC values across models

The ROC curves in Figure 4 show that most boosting models and the Stacking Ensemble had relatively similar discriminative ability. The curves largely overlapped across most of the false-positive-rate range, with AUC values of approximately 0.92. Random Forest and Extra Trees performed lower, with mean ROC-AUC values of 0.9105 and 0.9038, respectively, in the cross-fold evaluation. The differences in ROC-AUC among the best-performing models were small. AdaBoost obtained the highest value, at 0.9221, followed by CatBoost at 0.9220, the Stacking Ensemble at 0.9218, and XGBoost at 0.9212. These results indicate that increasing architectural complexity did not lead to substantial differences in discriminative ability among the top-performing models.

The ROC-AUC ranking plot in Figure 5 further shows that XGBoost, AdaBoost, CatBoost, the Stacking Ensemble, Gradient Boosting, and Logistic Regression were located within a narrow performance range. In contrast, Random Forest and Extra Trees showed a more pronounced decline. Therefore, ROC-AUC evaluation should be complemented by analyses of recall, F1-score, training time, and inference latency so that model selection is not based solely on probabilistic discriminative ability.

Overall, the findings indicate that gradient-boosting-based algorithms, such as CatBoost, XGBoost, Gradient Boosting, and LightGBM, achieved better performance than bagging-based methods and the stacking ensemble. This advantage is attributable to the ability of boosting algorithms to build models sequentially by correcting prediction errors from previous iterations, thereby improving generalization. CatBoost achieved the best performance because it incorporates ordered boosting and effective handling of categorical features, which can reduce prediction shift and the risk of overfitting [7]. Meanwhile, XGBoost produced comparable performance with higher computational efficiency through optimized gradient boosting and effective regularization techniques [2]. These findings are consistent with Zhang and Jánošík [10], who reported that CatBoost and XGBoost are highly competitive algorithms across various classification tasks, and with Zhao et al. [11], who demonstrated that ensemble learning approaches can improve predictive capability compared with single classification models. Accordingly, algorithm selection should consider not only accuracy but also computational efficiency and the characteristics of the dataset.

C. Confusion Matrix Evaluation

Figure 6 presents the confusion matrices for all evaluated models. Overall, all algorithms correctly classified the majority of samples, as evidenced by the dominance of True Positive (TP) and True Negative (TN) values over False Positive (FP) and False Negative (FN) values. Boosting-based models specifically XGBoost, CatBoost, Gradient Boosting, and LightGBM yielded relatively fewer false negative instances compared to other models, indicating a superior ability to identify students who were truly depressed. Conversely, Random Forest and Extra Trees produced higher numbers of misclassifications, particularly regarding False Positive and False Negative counts; this aligns with the lower accuracy, F1-score, and ROC-AUC values observed for these models.

Among all models, CatBoost produced the lowest number of false negatives (391 cases), followed by LightGBM (399 cases), Gradient Boosting and Histogram-Based Gradient Boosting (402 cases each), and XGBoost (405 cases). However, while Logistic Regression yielded the lowest number of false positives (385 cases), it produced a relatively higher number of false negatives (486 cases), indicating a more conservative tendency when making positive predictions. Meanwhile, the Stacking Ensemble did not significantly reduce misclassification errors compared to the best-performing boosting models, despite its substantially higher computational complexity. These findings reinforce the performance metric results by demonstrating that boosting models particularly XGBoost and CatBoost provide a better balance between detection sensitivity and classification accuracy without requiring a significant increase in model complexity.

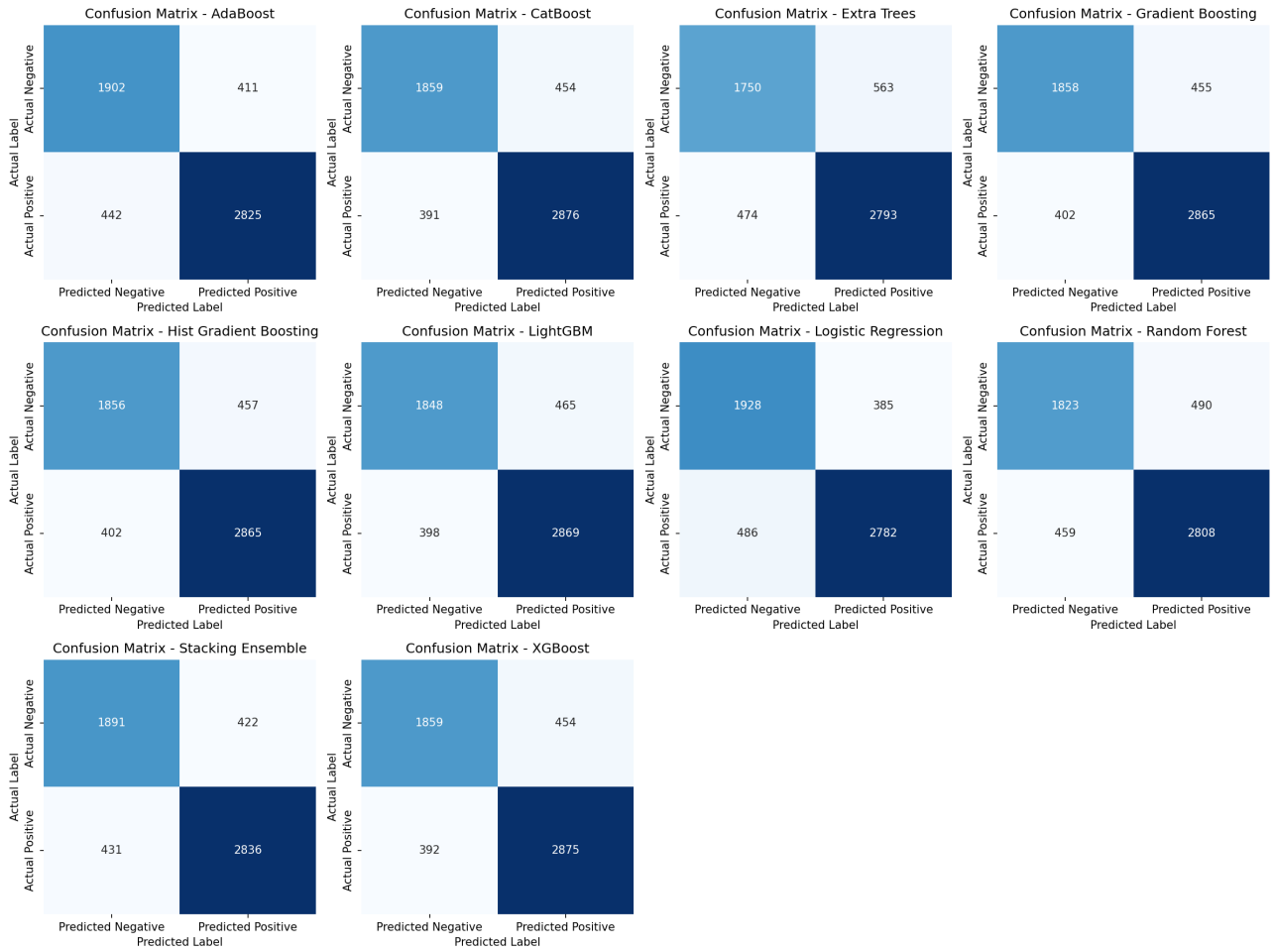

Figure 6. Confusion matrices of the evaluated models

D. Inter-Model Correlation of Prediction Probabilities

The heatmap in Figure 7 shows that the correlations of prediction probabilities across models were high, ranging from approximately 0.92 to 1.00. The correlation between Gradient Boosting and Histogram-Based Gradient Boosting reached 1.00, as did the correlations between Gradient Boosting

and both XGBoost and the Stacking Ensemble. XGBoost, LightGBM, CatBoost, Logistic Regression, and the Stacking Ensemble also showed correlations of approximately 0.98–0.99. The lowest correlation was observed between Extra Trees and the Stacking Ensemble, at 0.92; however, this value was still considered high.

Overall, these results indicate that most models produced highly similar prediction-probability patterns. This high degree of predictive similarity suggests limited diversity in the information available to the meta-learner in the Stacking Ensemble. This finding does not directly establish a causal relationship between prediction correlation and the performance of the Stacking Ensemble. However, the very high correlations indicate that combining models with nearly uniform prediction patterns provides only limited complementary information to the meta-learner.

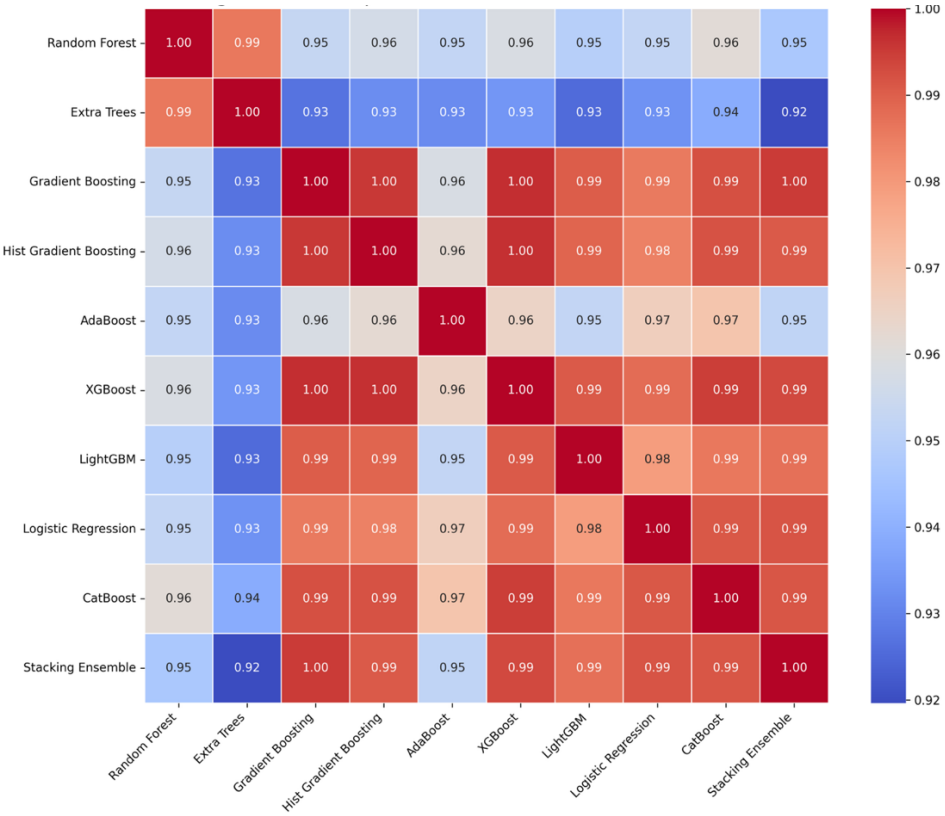

Figure 7. Heatmap of inter-model prediction-probability correlations

E. Computational Efficiency Analysis

Computational efficiency was analyzed using training time in seconds and inference latency in milliseconds per sample. The evaluation results revealed substantially greater variation in computational performance than in predictive metrics.

Table 3. Comparison of training time and inference latency across models

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC	Training Time (s)	Inference Time (ms/sample)	TN	FP	FN	TP
Random Forest	0,8273	0,8485	0,8583	0,8534	0,9089	10,3191	0,0804	1.812	501	463	2.805
Extra Trees	0,8117	0,8298	0,8534	0,8415	0,9032	12,7149	0,0832	1.741	572	479	2.789
Gradient Boosting	0,8443	0,8623	0,8736	0,8679	0,9193	33,4464	0,0147	1.857	456	413	2.855
Hist Gradient Boosting	0,8416	0,8601	0,8712	0,8656	0,9182	6,6219	0,0841	1.850	463	421	2.847
AdaBoost	0,8468	0,8776	0,858	0,8677	0,9197	9,7861	0,0633	1.922	391	464	2.804
XGBoost	0,8461	0,8631	0,8761	0,8696	0,9198	6,4849	0,0084	1.859	454	405	2.863

LightGBM	0,8414	0,8551	0,8779	0,8664	0,9174	0,986	0,0267	1.827	486	399	2.869
Logistic Regression	0,8416	0,8777	0,8476	0,8624	0,9184	73,8059	0,001	1.927	386	498	2.770
CatBoost	0,8459	0,8637	0,8748	0,8693	0,9196	12,3002	0,0084	1.862	451	409	2.859
Stacking Ensemble	0,8439	0,8714	0,8605	0,8659	0,9195	220,1153	0,068	1.898	415	456	2.812

LightGBM achieved the fastest training time, at 1.4566 ± 1.1412 seconds per fold. XGBoost ranked second, with a training time of 3.4616 ± 0.6276 seconds, followed by Random Forest at 4.1153 ± 0.7142 seconds, Extra Trees at 4.6550 ± 1.1665 seconds, and Histogram-Based Gradient Boosting at 6.0570 ± 0.3986 seconds. AdaBoost required 9.7345 ± 0.8718 seconds for training, whereas CatBoost required 11.9349 ± 0.8015 seconds. Gradient Boosting imposed a higher computational burden, with a training time of 29.4878 ± 0.7124 seconds. Logistic Regression required 70.4957 ± 0.8438 seconds, while the Stacking Ensemble had the longest training time, at 213.5678 ± 1.9167 seconds per fold.

Using LightGBM as the reference, the Stacking Ensemble required approximately 146.62 times longer training time. It was also approximately 61.70 times slower than XGBoost and 17.89 times slower than CatBoost. However, this substantially higher computational cost was not accompanied by an improvement in F1-score. The Stacking Ensemble achieved an F1-score of 0.8693, which was identical to that of LightGBM and lower than those of CatBoost and XGBoost.

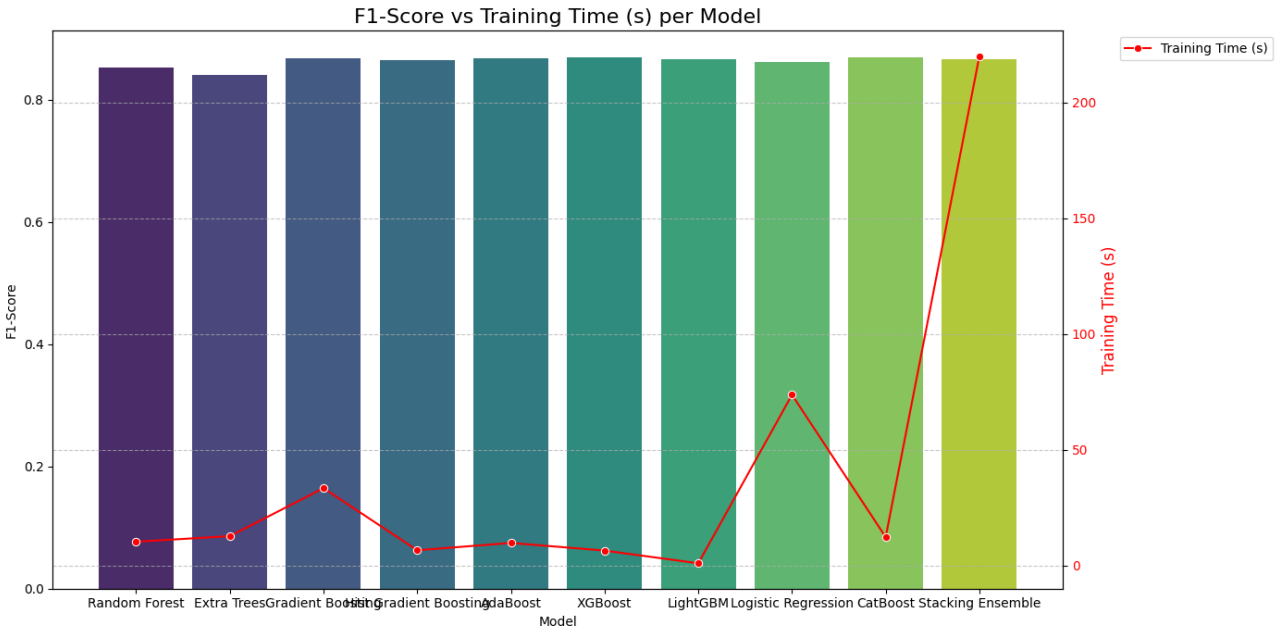

Figure 8. Trade-off between F1-score and training time across models

The visualization in Figure 8 shows that the F1-score bars for all boosting models and the Stacking Ensemble fall within a relatively narrow range. In contrast, the training-time line increases sharply for Logistic Regression and, most notably, for the Stacking Ensemble. This pattern indicates that higher training costs are not necessarily accompanied by proportional improvements in predictive performance.

In terms of inference, Logistic Regression recorded the lowest latency, at 0.0007 ± 0.0003 ms per sample. CatBoost ranked next, with 0.0053 ± 0.0026 ms per sample, followed by XGBoost at 0.0097 ± 0.0042 ms per sample and Gradient Boosting at 0.0140 ± 0.0007 ms per sample. LightGBM required 0.0315 ± 0.0102 ms per sample. Random Forest and Extra Trees recorded latencies of 0.0457 ± 0.0118 and 0.0464 ± 0.0102 ms per sample, respectively. AdaBoost and Histogram-Based Gradient Boosting had latencies of approximately 0.0691 and 0.0698 ms per sample, respectively, whereas the Stacking Ensemble was the slowest model, with an inference latency of 0.0775 ± 0.0141 ms per sample.

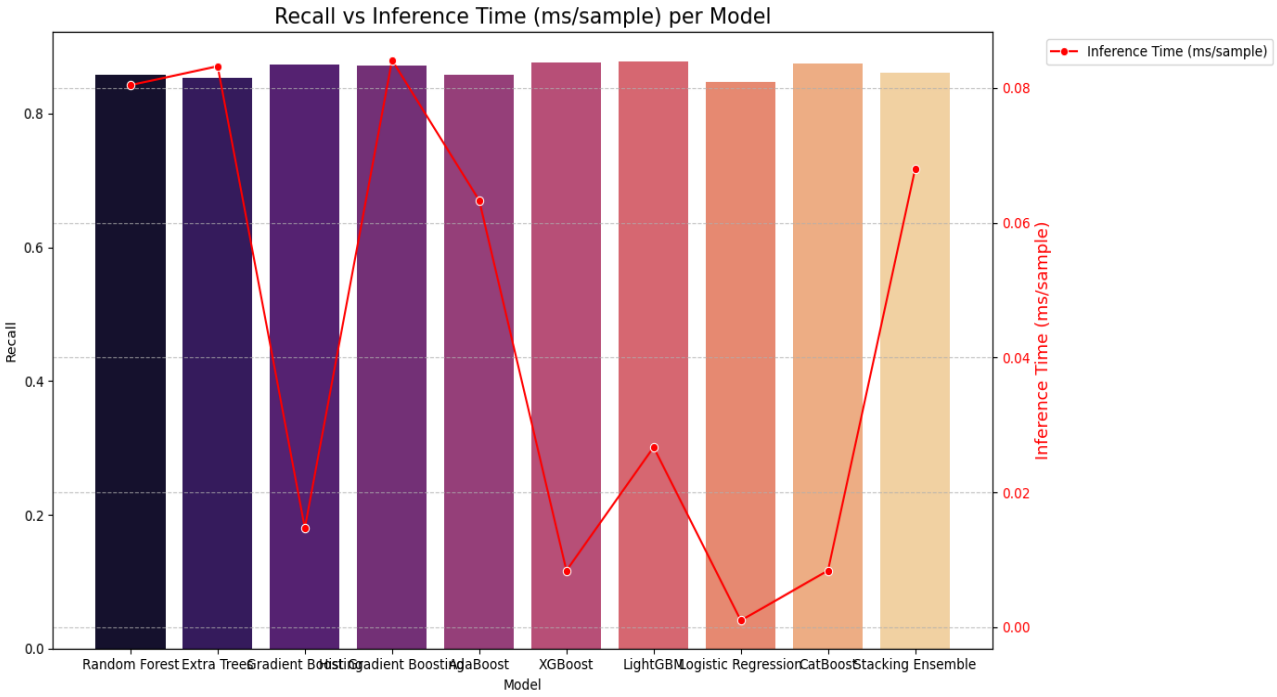

Figure 9. Trade-off between recall and inference latency across models

The figure shows that CatBoost and XGBoost occupy a favorable position because they achieved high recall while maintaining low inference latency. LightGBM obtained a recall of 0.8782 but had higher latency than CatBoost and XGBoost. The Stacking Ensemble exhibited the highest latency, while its recall reached only 0.8680.

F. Feature Importance Analysis

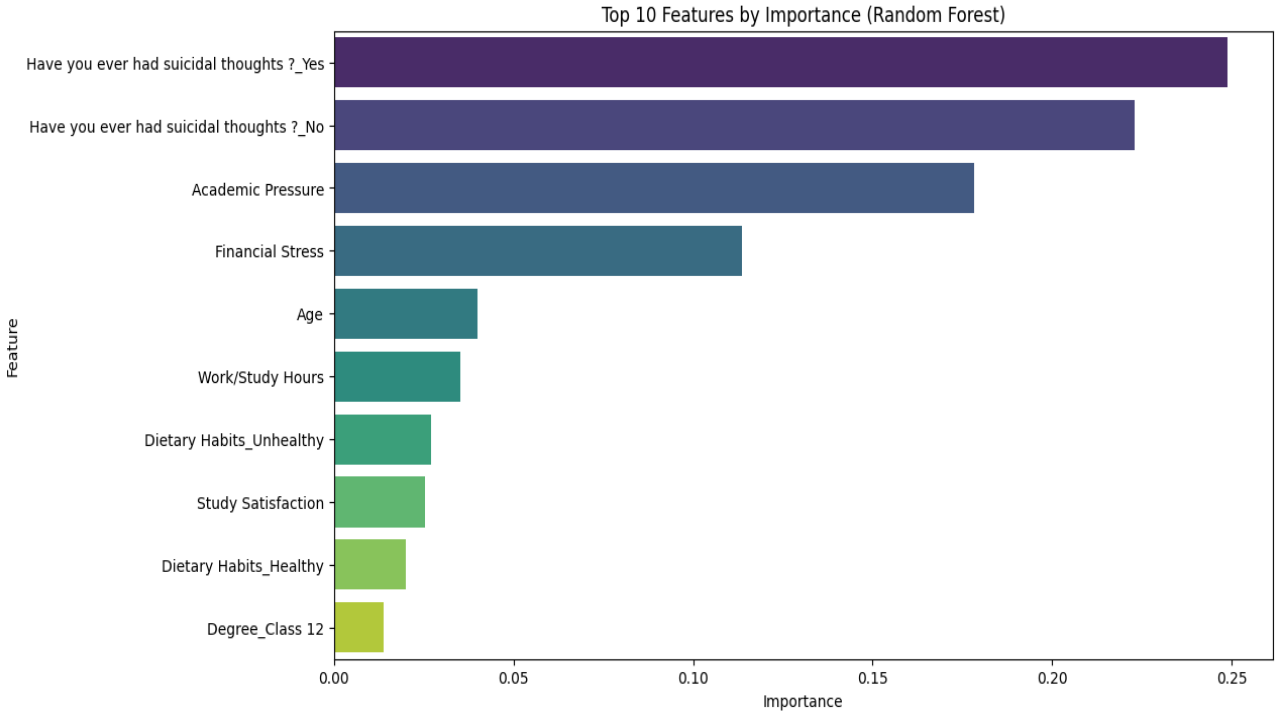

Figure 10. Ten most important features according to Random Forest

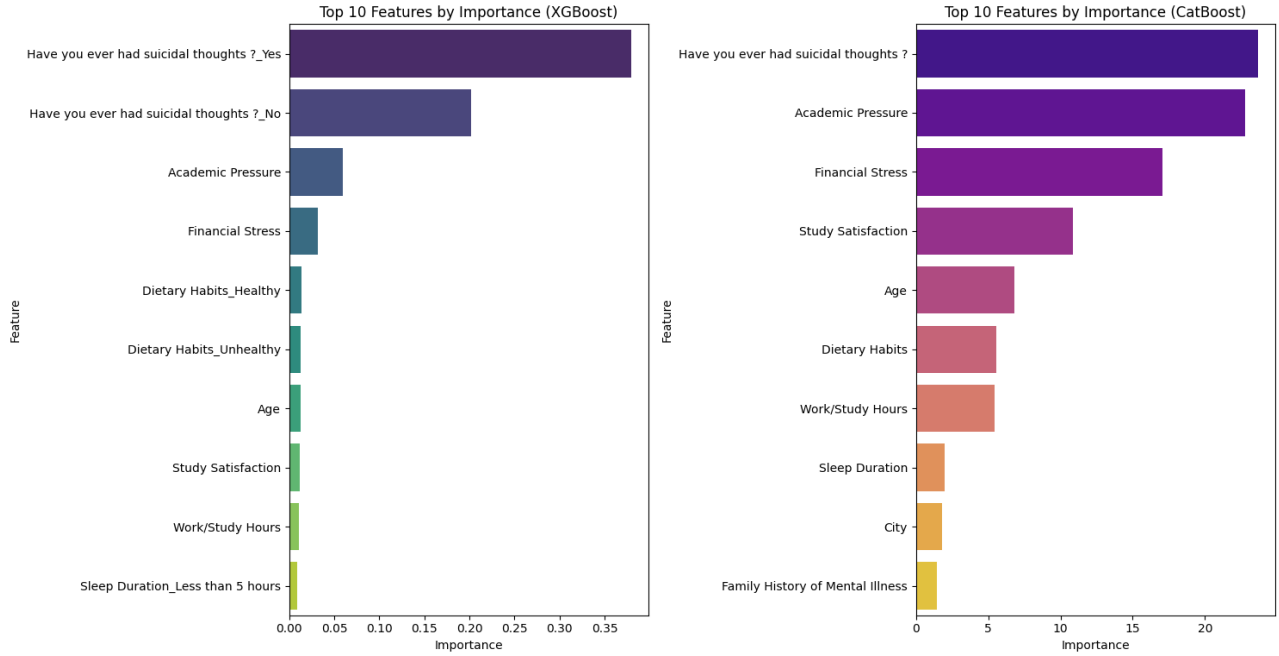

Figure 11. Ten most important features according to XGBoost and CatBoost

The feature importance analysis revealed consistency across several key variables. History of suicidal thoughts was the most influential feature in Random Forest, XGBoost, and CatBoost. Other variables that consistently appeared among the top-ranked features included academic pressure, financial pressure, study satisfaction, age, work or study hours, dietary habits, and sleep duration.

In XGBoost, the one-hot-encoded representation of history of suicidal thoughts contributed most dominantly, followed by academic pressure and financial pressure. In CatBoost, the features with the highest contributions were history of suicidal thoughts, academic pressure, financial pressure, study satisfaction, age, dietary habits, and work or study hours. Importance values were not compared directly across models because Random Forest and XGBoost used encoded feature representations, whereas CatBoost retained categorical features in a different representation. Therefore, these visualizations were used to identify the consistency of important-factor rankings rather than to compare the absolute magnitude of importance values across models.

G. Statistical Testing Results

The one-way analysis of variance test on ROC-AUC scores across models yielded an F-statistic of 71.6719 with a p-value < 0.001. This result indicates a significant difference in mean ROC-AUC for at least one model group.

Table 3. Results of Tukey HSD post hoc tests on ROC-AUC

Model Comparison	Mean ROC-AUC Difference	p-adjusted	Decision
AdaBoost vs CatBoost	-0.0002	1.0000	Not significantly different
AdaBoost vs XGBoost	-0.0010	0.9937	Not significantly different
AdaBoost vs Stacking Ensemble	-0.0003	1.0000	Not significantly different
CatBoost vs XGBoost	-0.0008	0.9983	Not significantly different
CatBoost vs Stacking Ensemble	-0.0002	1.0000	Not significantly different
Gradient Boosting vs XGBoost	0.0006	0.9999	Not significantly different
Gradient Boosting vs Stacking Ensemble	0.0012	0.9741	Not significantly different
Hist Gradient Boosting vs XGBoost	0.0010	0.9907	Not significantly different
LightGBM vs XGBoost	0.0027	0.2432	Not significantly different
LightGBM vs Stacking Ensemble	0.0033	0.0672	Not significantly different
Logistic Regression vs XGBoost	0.0000	1.0000	Not significantly different
Extra Trees vs XGBoost	0.0174	<0.001	Significantly different
Extra Trees vs CatBoost	0.0182	<0.001	Significantly different

Model Comparison	Mean ROC-AUC Difference	p-adjusted	Decision
Extra Trees vs AdaBoost	0.0183	<0.001	Significantly different
Random Forest vs XGBoost	0.0106	<0.001	Significantly different
Random Forest vs CatBoost	0.0114	<0.001	Significantly different
Random Forest vs AdaBoost	0.0116	<0.001	Significantly different

Based on the Tukey HSD post hoc test results in Table 3, most comparisons among the high-performing models showed no significant differences in ROC-AUC. CatBoost did not differ significantly from XGBoost (adjusted p-value = 0.9983) or the Stacking Ensemble (adjusted p-value = 1.0000). Similarly, AdaBoost did not differ significantly from CatBoost, XGBoost, or the Stacking Ensemble, while Gradient Boosting and Histogram-Based Gradient Boosting did not differ significantly from XGBoost. LightGBM also showed no significant difference from XGBoost (adjusted p-value = 0.2432) or the Stacking Ensemble (adjusted p-value = 0.0672).

In contrast, Extra Trees showed significant differences from AdaBoost, CatBoost, and XGBoost, while Random Forest differed significantly from all three models, with all adjusted p-values < 0.001. These results indicate that numerical differences in ROC-AUC among the top-performing models do not necessarily represent statistically meaningful differences. Although the Stacking Ensemble required the longest training time, its ROC-AUC did not differ significantly from those of CatBoost, XGBoost, or AdaBoost. Because the scores were obtained from the same cross-validation folds, the ANOVA and Tukey HSD results are positioned as supporting parametric analyses; Friedman testing and pairwise comparisons may be used in future evaluations to account for the repeated-measures structure across folds.

H. Summary of Findings

Overall, CatBoost achieved the highest accuracy (84.86%), recall (88.03%), and F1-score (87.19%), whereas AdaBoost obtained the highest ROC-AUC value (0.9221). However, the Tukey HSD test showed that the ROC-AUC differences among CatBoost, AdaBoost, XGBoost, Gradient Boosting, and the Stacking Ensemble were not statistically significant ($p > 0.05$), indicating that these boosting-based algorithms had relatively comparable discriminative ability. XGBoost delivered classification performance that was very close to that of CatBoost while requiring a shorter training time (3.92 seconds), whereas LightGBM was the most efficient algorithm, with an average training time of only 0.85 seconds and no meaningful decline in performance. In contrast, the Stacking Ensemble required the highest training and inference times but did not improve performance compared with the individual boosting models.

These findings are consistent with Zhang and Jánošík [10], who reported that CatBoost and XGBoost are highly competitive boosting algorithms for tabular data. The findings also align with Suryadi [11], who showed that XGBoost can provide a favorable balance between accuracy and computational efficiency. In addition, the efficiency of LightGBM observed in this study supports the findings of Ke et al. [6], who stated that histogram-based gradient boosting can accelerate the training process without substantially compromising predictive performance. Conversely, this study shows that the Stacking Ensemble does not always improve performance over individual boosting models. This finding indicates that the success of stacking is strongly influenced by the degree of diversity among base learners. When the base models produce highly similar predictions, aggregation through a meta-learner may not provide sufficient complementary information to improve classification performance.

Practically, these findings emphasize that classification model selection should not rely on a single predictive metric alone. Instead, it should also consider classification sensitivity, the balance between precision and recall, probabilistic discriminative ability, computational efficiency, and statistical significance across models. Accordingly, algorithm selection can be tailored to implementation needs, whether the objective is to achieve maximum predictive performance or to support a student mental health screening system that requires rapid responses and efficient use of computational resources.

CONCLUSION

This study successfully conducted a comparative analysis of nine ensemble learning algorithms and one non-ensemble baseline model for classifying university students' mental health status using the Student Depression Dataset. The evaluation results show that boosting-based algorithms generally achieved better predictive performance than bagging-based methods, with all boosting models obtaining ROC-AUC values above 0.91. Based on the classification metrics, CatBoost delivered the highest predictive performance, with an accuracy of 84.86%, recall of 88.03%, and F1-score of 87.19%. However, the Tukey HSD test showed that the performance differences among CatBoost, XGBoost, AdaBoost, Gradient Boosting, and the Stacking Ensemble were not statistically significant in terms of ROC-AUC.

From the perspective of computational efficiency, XGBoost demonstrated the best balance between predictive performance and computational cost, achieving a recall of 88.00%, F1-score of 87.17%, ROC-AUC of 0.9212, an average training time of 3.92 seconds, and an inference latency of 0.0084 ms per sample. Meanwhile, LightGBM was the fastest algorithm in terms of training time, requiring only 0.85 seconds, although its predictive performance was slightly lower than those of CatBoost and XGBoost. In contrast, the Stacking Ensemble required approximately 215 seconds for training without providing a meaningful performance improvement over the individual boosting models. This finding indicates that increasing model complexity does not necessarily improve classification capability, particularly when the base learners produce highly similar predictions, thereby limiting the additional information that can be exploited by the meta-learner.

Overall, this study shows that algorithm selection for classifying university students' mental health status should consider the balance among predictive performance, computational efficiency, and implementation requirements. If the primary objective is to obtain the highest classification performance, CatBoost is the most suitable choice. Conversely, if the system is intended for implementation in a web-based mental health screening application that requires fast response times and low computational cost, XGBoost offers the best balance between accuracy and computational efficiency.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to the Directorate General of Research and Development, Ministry of Higher Education, Science, and Technology of the Republic of Indonesia, for providing financial support through the Beginner Lecturer Research Program (PDP) BIMA. The authors also thank the Akademi Manajemen Informatika dan Komputer Polibisnis for its institutional support and research facilities during the implementation of this study. Appreciation is also extended to all members of the research team for their contributions to data collection, experimental implementation, and preparation of the research outputs.

REFERENCES

[1] A. Baba and K. Bunji, "Prediction of mental health problems using annual student health survey: Machine learning approach," JMIR Formative Research, vol. 7, e42420, 2023, doi:10.2196/42420.

[2] V. Elysia, I. Rokhiyah, M. Y. Setiani, A. Priatna, and T. Chandrawati, "Examining mental health needs of open and distance learning students: A case study of Universitas Terbuka, Indonesia," Jurnal Penelitian dan Evaluasi Pendidikan, vol. 18, no. 2, pp. 89–98, 2024, doi:10.21831/jpip.v18i2.94621.

[3] M. J. Faisti, R. H. Kusumodestoni, and G. W. N. Wibowo, "Mental health classification using Naïve Bayes and Random Forest algorithms," Journal of Applied Informatics and Computing, vol. 9, no. 4, pp. 315–322, 2025, doi:10.30871/jaic.v9i4.8448.

[4] W. Febriani, M. Mambang, S. E. Prastya, B. Sabella, and F. D. Marleny, "Analysis of the impact of violent content on social media on adolescent cyberpsychology using Support

Vector Machine and Random Forest," Journal of Applied Informatics and Computing, vol. 10, no. 1, pp. 45–52, 2026, doi:10.30871/jaic.v10i1.8901.

[5] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794, doi:10.1145/2939672.2939785.

[6] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in Advances in Neural Information Processing Systems, vol. 30, 2017.

[7] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in Advances in Neural Information Processing Systems, vol. 31, 2018.

[8] L. Zhang and D. Jánošík, "Enhanced short-term load forecasting with hybrid machine learning models: CatBoost and XGBoost approaches," Expert Systems with Applications, vol. 241, 122686, 2024, doi:10.1016/j.eswa.2023.122686.

[9] M. H. Izzaturrahim, I. B. P. Prayoga, A. N. Hilmy, N. P. Utama, and A. Purwarianti, "Implementation of Stacking Ensemble Learning method for fraud detection in financial transactions," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 11, no. 3, pp. 521–530, 2024, doi:10.25126/jtiik.202411388601.

[10] S. Zhao, D. Zhou, H. Wang, and L. Yu, "Ensemble-learning techniques for predicting student performance on video-based learning," International Journal of Information and Education Technology, vol. 12, no. 8, pp. 741–745, 2022, doi:10.18178/ijiet.2022.12.8.1678.

[11] S. Suryadi, "XGBoost algorithm for cervical cancer risk prediction: Multi-dimensional feature analysis," Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 9, no. 3, pp. 619–625, 2025, doi:10.29207/resti.v9i3.6587.

[12] Trelles, A., Ruiz, T. F., & Rojo, A. P., *Systematic Review and Meta-Analysis of Explainable Machine Learning Models for Clinical Depression Detection*, Behavioral Sciences, vol. 15, no. 11, 2025. DOI: 10.3390/bs15111476.

[13] Schaab, B. L., et al., How do machine learning models perform in the detection of depression, anxiety, and stress among undergraduate students? A systematic review, Cadernos de Saúde Pública, vol. 40, no. 11, 2024. DOI: 10.1590/0102-311XEN029323.

[14] Cai, Y., Lin, D., & Lu, Q., Comparison of Different Machine Learning Algorithms in the Mental Health Assessment of College Students, Journal of ICT Standardization, 2024. DOI: 10.13052/jicts2245-800X.1243.