

UTILIZING AN ARTIFICIAL INTELLIGENCE-BASED APPROACH TO EVALUATE STUDENT ACADEMIC PERFORMANCE THROUGH THE APPLICATION OF AN ENSEMBLE ALGORITHM

Manoher C. Alipour¹

¹Bachelor of Science in Information Technology Program, Isabela State University, Angadanan, 3307, Isabela, Philippines.

¹<https://orcid.org/0009-0006-9594-8233>

E-mail: Manoher.c.alipour@isu.edu.ph

ARTICLE INFO

Article History

Received: February 15, 2026

Reviewed: March 15, 2026

Accepted: April 01, 2026

Published: May 29, 2026

Keywords:

Student Performance,
Educational Data Mining,
Machine Learning,
Ensemble Model,
Prediction.

ABSTRACT

This study addresses the use of machine learning to identify a pattern for predicting student academic performance in the Philippines. The data used in this study comes from the UCI Student Performance dataset. The researcher used Multinomial logistic regression, Long Short-Term Memory, and eXtreme Gradient Boosting to create an ensemble model in which all three algorithms work together to solve the problem, in this case, finding a pattern for student performance, and uses 10-fold cross-validation for robust model training. The dataset is divided into four categories for Student Performance, Portuguese language dataset, Math dataset, joint dataset, and merge Dataset. The results demonstrate that the Ensemble model achieves high predictive performance for student-combined-join (accuracy of 86.5), student-merged (accuracy of 85.4), student-mat (accuracy of 86.5), and student-por (accuracy of 84.8). The results show that the created model could further enhance students' performance by using predictive models that help students, schools, and parents identify shortcomings in students and curriculum outcomes. The model exhibits robust performance across all evaluated systems, suggesting that thoughtful model selection and combination yield accurate classifications across various metrics. This indicates not only the effectiveness of individual models but also the importance of their synergistic interaction in improving overall classification accuracy.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Artificial intelligence (AI) and Machine Learning (ML) creates unforeseen destiny for humanity, it's one of the best inventions of humankind that changes everything and has the potential to break or make the human future by impacting every aspect of life, from the smallest to the biggest domain humans have engaged with, such as livelihood, career, business, learning, and above all, human innovations. The involvement of AI across a wide variety of human life could change how humans interact with their domain. However, creating or using one algorithm does not solve multiple problems. For this purpose, we need to use several algorithms to solve a problem, since each targets different aspects of the datasets provided for analysis. Educational data mining (EDM) is a method that uses AI to understand patterns in human learning across different topics, helping them learn and understand better by analyzing data generated during learning. The learning process depends on various criteria and variables. In addition, environments play a significant role in students' performance, both in remote and face-to-face settings, despite the heterogeneity of educational data (demographic, numerical, and categorical features). EDM has the potential to identify learning difficulties based on data that prevents students from learning correctly and efficiently, by analyzing the data provided. This analysis helps students, academic institutions, families, and decision-makers make smarter decisions and create better policies and applications, such as gamification and interactive multimedia for subjects and areas that need improvement. EDM allows predicting students' performance behaviors and identifying those at risk of failure or dropout, monitoring students' performance based on different activities such as quizzes, exams[1],[2].

In the real world, a single algorithm cannot solve all problems and achieve high-quality results; therefore, combining different algorithms can help better understand and recognize patterns in the provided data. Ensemble learning combines multiple learning methods to achieve better predictive accuracy. Other methods used by various researchers to analyze students' performance behaviors include Multinomial logistic regression [3], machine learning, random forest classification, clustering, Meta-Analysis, Light GBM, GA-based algorithms, and combinations of algorithms to generate better, more accurate results [2], [4-6].

The researcher in this study, in the context of prior studies on students' performance prediction, considers the importance of enhancing students' output using AI. This study presents a combination of multiple algorithms: Extreme Gradient boosting (XGBoost) that corrects the errors of weak learners and often performs well on structured data, Multinomial logistic regression (LR) that works well with binary classification and probability estimation, and long short-term memory (LSTM) which understanding sequential or time-dependent patterns, such as language or temporal signals in one unified ensemble to work together to find a pattern in the data provided which taken form UCI Student Performance in two subject language Mathematics (mat) and Portuguese language (por) [7-12].

The rest of this study is as follows: Section 2, is Related Works, which discussed other researchers approaches to solve the problems and formulas that used for each algorithms moreover Section 3, Methodology, which describes in detail how the ensemble approach works, from inputting data to final predictions, including data preprocessing that prepares data for different models, base learners, algorithm implementations, and communications between them, and the process that took place to find average weighting. Furthermore, Section 4 Results and discussions present evaluations of different algorithms, cross-validation, performance metrics, and results from distinct algorithms and an ensemble pipeline using various metrics, and compare the Ensemble's performance with that of individual algorithms. And finally, section 5 conclusion, which summarizes all the information.

II. RELATED WORKS

The current study focused on Extreme Gradient Boosting (XGBoost), Multinomial logistic regression (LR), and Long Short-Term Memory (LSTM), combined to solve a problem and identify patterns in the dataset to accurately predict the outcome of the provided data. The strengths of each named algorithm combine to create an ensemble pipeline, which inspired the researcher to employ them. The researchers of this study [13]. Made a hybrid modeling framework that combines and integrates LSTM networks with LightGBM and CatBoost to work together for stock price prediction. After processing the data, they evaluated the output, which resulted in a 10-15% overall improvement in predictive accuracy and reduced error, suggesting that the ensemble strategy approach indeed improves results and enables the approach to find patterns in the dataset. The researchers of this study [14] An implicit particle swarm optimization algorithm that identifies different parameters for XGBoost [15] Training is a key component of the hybrid algorithm. By combining the two algorithms, the problem of low accuracy and efficiency is addressed. The researchers of this study [16] Used the naive Bayes classifiers to deal with noise in categorical data, which often happens in many datasets. Their study suggested, in their methodology, that as the uncertainty of a feature increases in the input data, it directly contributes to lower predictive performance and lower accuracy. This is achieved by integrating bidirectional-LSTM networks with the Naive Bayes classification algorithm.

II.1 ALGORITHMS

II.1.1 Gradient Boosting

This algorithm is a powerful predictive model used across environments that primarily require ML approaches. The formula for this model is an additive weak-learner model [17]. Gradient Boosting: a machine learning algorithm, a powerful technique used for predictive modeling. Gradient boosting is widely used for both classification and regression tasks and often provides state-of-the-art results on structured data.

II.1.2 XGBoost

Objective function: regularization helps mitigate the risk of overfitting. This approach incorporates regularization terms directly into the objective function, which serves as the guiding metric for model optimization, in addition to the objective function ($\mathcal{L}$) which includes a loss function term $\mathcal{L}(\mathbf{y}_i, \hat{\mathbf{y}}_i)$ and a regularization term $\Omega(\mathbf{f}_t)$.

$$\mathcal{L} = \sum_i \mathcal{L}(\mathbf{y}_i, \hat{\mathbf{y}}_i) + \sum_i \Omega(\mathbf{f}_t) \quad (1)$$

The regularization term: where T is the number of leaves in the given tree and w are the output scores for the leaves with respect to the tree. This loss function can be merged into the split criterion of decision trees, yielding a pruning strategy. The value of γ controls the minimum loss reduction gain needed to split an internal node. Furthermore, higher values of γ result in simpler trees. Shrinkage is an additional regularization parameter in XGBoost, the purpose of shrinkage is to reduce the step size in the additive expansion. Finally, the complexity of the trees can also be limited using other strategies, such as limiting the tree depth. A secondary benefit of reduced tree complexity is that models are trained faster and require less storage space.

$$\Omega(\mathbf{f}_t) = \gamma T + \frac{1}{2} \lambda \|\mathbf{w}\|^2 \quad (2)$$

II.1.3 Multinomial Logistic Regression

Multinomial logistic regression: generalizes binary logistic regression to cases where the dependent variable y can take more than two discrete outcomes. This is the standard probability model used in neural networks for classification. where: $\mathbf{W}_k$ = weight vector for class k , $\mathbf{b}_k$ = bias for class k , $\mathbf{x}$ = input features, The numerator computes an unnormalized score for class k and the denominator normalizes across all K classes, so the probabilities sum to 1.

$$P(\mathbf{y} = \mathbf{k} \mid \mathbf{x}) = \frac{\exp(\mathbf{W}_k^T \mathbf{x} + \mathbf{b}_k)}{\sum_{j=1}^K \exp(\mathbf{W}_j^T \mathbf{x} + \mathbf{b}_j)} \quad (3)$$

II.2 PERFORMANCE METRICS

In this section, the researcher discusses the metrics used for this study, and the researcher rigorously analyzes the data and results using different metrics. Accuracy: the total number of instances correctly classified. It treats all classes equally; however, it may provide a misleading representation in the context of imbalanced datasets, as it does not account for the performance of individual classes. Where N is the total number of predictions, $\hat{y}_i$ is the predicted value for the i -th instance, y_i is the actual true value for the i -th instance, $1(\hat{y}_i = y_i)$ The indicator function is a binary variable that takes the value 1 when the prediction is accurate and 0 when it is not. The $\sum_{i=1}^N$ counts the total number of correct predictions.

$$Accuracy = \frac{1}{N} \sum_{i=1}^N 1(\hat{y}_i = y_i) \quad (4)$$

Precision is a metric that evaluates the precision of a classification model by calculating it for each class, then averaging these values unweighted. This approach ensures that every class is treated equally, regardless of how many instances of each class are present in the dataset. Precision: Calculation formula for the precision of a specific class, and k is the ratio of true positives (TP_k) to the total number of instances predicted as positive for that class ($TP_k + FP_k$), includes both true positives and false positives (FP_k).

$$Precision_k = \left(\frac{TP_k}{TP_k + FP_k} \right) \quad (5)$$

Macro-Averaged Precision: The formula calculates the average of the individual precision scores across all K classes. Macro recall measures the ability of the classifier to correctly identify instances of each class, averaged equally across classes. It is especially useful when missing minority classes are costly.

$$Precision_{macro} = \frac{1}{K} \sum_{k=1}^K Precision_k \quad (6)$$

Recall: per Class where TP_k : True Positives for class k , the number of instances correctly identified as belonging to that class. FN_k : False Negatives for class k , the number of instances that actually belonged to class k but were missed by the model. This metric measures the classifier's ability to find all relevant cases of a specific class (its completeness or sensitivity).

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \quad (7)$$

Macro-Averaged Recall: Where K The total number of classes. $\sum_{k=1}^K Recall_k$: The sum of the recall scores for each individual class. Macro-averaging calculates the metric independently for each class and then takes the average (mean) of these scores, treating all classes equally, regardless of the number of instances in each class.

$$Recall_{macro} = \frac{1}{K} \sum_{k=1}^K Recall_k \quad (8)$$

Macro F1 is the unweighted mean of per-class F1 scores. It balances precision and recall per class and is robust to class imbalance by preventing dominant classes from overwhelming the metric. The F1 score for a specific class, denoted as $F1_k$. It is the harmonic mean of the precision and recall for that class, balancing the two metrics into a single value. Precision ($Precision_k$) Measures the accuracy of the optimistic predictions for class k (how many selected items are relevant). ($Recall_k$) identify all actual positive instances of class k . The F1 score is helpful for imbalanced datasets because it provides a better measure of the model's performance on the positive class than simple accuracy.

$$F1_k = \frac{2 \cdot Precision_k \cdot Recall_k}{Precision_k + Recall_k} \quad (9)$$

Macro-averaged F1 score. This is used in multi-class classification problems to get a single performance metric across all classes. The formula works by calculating the F1 score ($F1_k$) for each class independently and then taking the arithmetic mean (unweighted average) of those individual scores. Where K represents the total number of classes.

$$F1_{macro} = \frac{1}{K} \sum_{k=1}^K F1_k \quad (10)$$

Regardless of the number of instances of each class present in the given dataset, macro-averaging treats all classes equally. Receiver Operating Characteristic (ROC) curves computed under a One-versus-Rest (OvR). evaluates samples how well the model ranks positive above negative ones for each class using a one-vs-rest strategy, then averages the results. It is threshold-independent and reflects the model's discriminative power across all classes. The AUC is calculated for an individual class k , denoted as (AUC_k). It is an integral of the True Positive Rate (TPR) over the False Positive Rate ($x = FPR$). This integral formulation is a known method for computing the AUC from the ROC curve definition. This formula provides an aggregate measure of the classifier's performance across all classes, treating each class's performance equally.

$$AUC_k = \int_0^1 TPR(x) dx \quad (11)$$

III. METHODOLOGY

In this section, the researcher presents a detailed explanation of how all the algorithms used in this study work together to complete the task. For this study, the researchers used Google Colab and a Python environment. The hardware used for this training included a standard virtual machine in the Colab environment, with no GPU. The dataset was downloaded from the foundational UCI Machine Learning Repository at the following website address, which is freely accessible (<https://archive.ics.uci.edu/ml/datasets/Student+Performance>, accessed on 10 January 2024).

The dataset includes 33 attributes related to students, which aims to predict the grade of G3. Figure 1 shows the phases of the different processes in the model and their interconnectivity, including the input, process, and evaluation phases, in which they work together to solve the given problem, and the model approach uses machine learning techniques that enhance performance by fit in the strengths of different algorithms into a single design, therefore This model allows different algorithms to work together, leveraging their strengths to solve different dimension of problems.

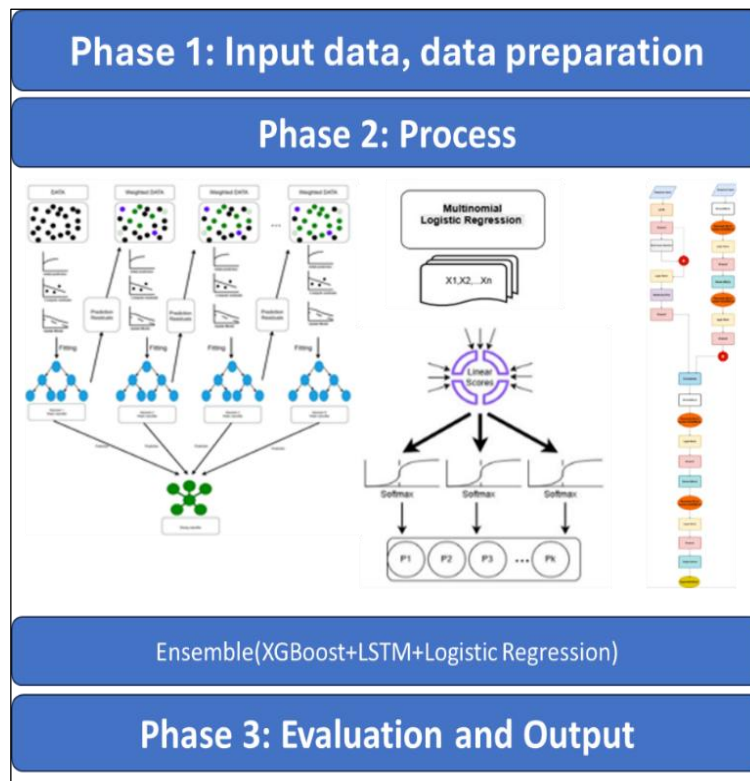


Figure 1: All three phases of the ensemble model.

Source: Author, (2026).

III.1 PHASE ONE

At the top of the diagram is the phase one input, in this phase, the dataset enters the model, in this phase data pre-process, normalized and clean followed by distribution and routed concurrently to four algorithms used in the second phase, All algorithms receive the same input, but process it according to their own algorithmic logic, including extreme Gradient Boosting, Multinomial Logistic Regression, a supervised ML that used for classifying different dataset typically binary output, and LSTM network ideal for forecasting tasks, selected by the researcher for this study, which comprises the ensemble pipeline, Furthermore, these four models differ in methodology. Still, they are intended to collaborate within a unified domain that leverages their strengths to solve the problems presented to them.

III.2 PHASE TWO

Phase two represent the algorithms that used by researcher in this study, after phase one ready the data, data simultaneously send to Extreme Gradient Boosting, Multinomial Logistic Regression, and LSTM, all algorithms received all data for training and revelations, in Extreme Gradient Boosting uses the histogram based technique, for speeding up the training and minimize the memory use its bins all numeric values into buckets, next step the algorithm enters the interactive boosting loop, in which in each interactive its compute the gradients and Hessians of the loss function, then using a decision tree that trained to predict the negative gradients and each leaf of tree receives a weight that calculated from the aggregated gradients and Hessians in that leaf, furthermore the process repeat many rounds until finally find best threshold for convergence threshold, then generate s class predictions by sending the test sample to all boosted trees, then apply the SoftMax for final transformation. Multinomial Logistic Regression uses the SoftMax to compute probabilities, rather than training each class to classify the multiclass problem.

During training, the algorithms start with random weight values and update along the way to find the best weight for the data presented to the algorithm, then measure the predicted probability distribution to match the actual class labels, and then update the weights to minimize the multinomial log-loss, in addition the optimizer computes gradients that put rules how each weight should change therefore it could adjust the weights until convergence, then for selecting highest probability that predicted applies the SoftMax functions to use the learned weights to compute the linear combination. LSTM algorithm layers configured to allow the model to get the forward and backward dependencies related to temporal dimension which returning the whole sequence of hidden states and attention layer computes contextualized representations across timesteps after residual connection adds the attention output back to the original LSTM output to stabilize training Layer normalization and Global Average Pooling compress the attended sequence into a single vector, followed by dropout for regularization then to ensure the attention mechanism pads inputs, the model applies a Multi-Head Self-Attention (MHA) layer that uses the previously generated mask,

Then output is added back to the LSTM output via a residual connection, followed by a Normalization Layer, which helps stabilize gradients and improve data flow, additionally dropout regularization is applied to lessen overfitting and the transformed sequence reduce into a single vector through Global Average Pooling. In parallel, the branch processes accept an input tensor with the same dimensionality as the tabular features and further pass it through two dense layers with ReLU activation, L2 regularization, dropout, and layer normalization. This merged vector flows into the model's classification, which consists of dense layers with Gaussian Error Linear Unit (GELU) activation, Layer Normalization, and Dropout, giving the network strong expressive capacity while controlling overfitting. The final layer is a Dense layer with three output units, corresponding to the three target classes, and uses a SoftMax activation to produce a normalized probability distribution over the classes.

III.3 PHASE THREE

Phase Three is the result of the evaluation of output using Accuracy, F1 Macro, Precision Macro, Precision Weighted, Recall Macro, and ROC AUC. To get the weights created by different processes, the model uses cross-validation with balanced accuracy as the scoring metric that let the model to assess the learnability of each algorithm by computing each model's weights. A proper weighting distribution achieves the required model process by collecting all weights, enforcing a minimum weight to ensure all models contribute to the overall weight. for the purpose of normalizing them so that their sum is one. The process multiplies each model's probability predictions by the learned weight and sums across all models to form the final model probability matrix. Class predictions are obtained by taking the class with the highest probability for each sample.

IV. RESULTS AND DISCUSSIONS

In this study, the researcher uses various machine learning and deep learning models to predict outcomes from multiple datasets. The models used in this study include XGBoost, Multinomial Logistic Regression, LSTM, and an Ensemble model combining all of them. The outputs of all algorithms yield good results, indicating that the algorithms correctly predict the outcomes of the majority of tests, as shown in Tables 1-4. LSTM scored well on the metrics for Student Performance, Portuguese language, Math Dataset, joint dataset, and merge Dataset, which are good metrics and remained stable over time, indicating the algorithm aims to capture actual cases. For XGBoost, the best overall test case correctness is across all metrics, showing strong performance across all classes and datasets. This tells that XGBoost is not just predicting the majority class, it is handling minority classes reasonably well. This means XGBoost is already a strong performer for this dataset. The Logistic Regression shows above-average metrics in many measurements, which indicates it has a reasonable baseline for all datasets. The final ensemble model, which combines all algorithms' predictions using learned weights, accurately identifies samples in the dataset. Results indicate that the prediction outcome and the actual labels are correctly predicted for all Datasets. Table 5 compares the proposed model with other researchers' models. Our approach is very close to the highest-performing model in the table, suggesting that the ensemble of XGBoost+LSTM+LR has potential for predicting students' performance. Figure 2 presents a comparative evaluation of an ensemble model using ROC and OvR; each subplot corresponds to a dataset of different size: student-mat (n = 395), student-por (n = 649), student-merged (n = 382), and student-combined-join (n = 1044). *Student-Mat*: with AUCs of 0.959 (Class 0), 0.930 (Class 1), and 0.967 (Class 2), with a micro-average AUC of 0.962.

- Student-Merged: with AUCs of 0.966 (Class 0), 0.932 (Class 1), and 0.959 (Class 2), and a micro-average AUC of 0.961.
- Student-Combined-Join: which has the largest sample size, with AUCs of 0.955 (Class 0), 0.930 (Class 1), and 0.973 (Class 2), with a micro-average AUC of 0.962.
- Student-Por: dataset shows the strongest overall separability, with class-wise AUCs of 0.976 (Class 0), 0.965 (Class 1), and 1.000 (Class 2), and a micro-average AUC of 0.977. The perfect AUC for Class 2 implies near-complete separation of that class from the others in terms of predicted scores.

Table 1: Summary results of the metric student performance Portuguese language course.

Models	Accuracy	F1 Macro	Precision Macro	Precision Weighted	Recall Macro	Roc auc ovr
XGBoost	0.9192	0.9329	0.9308	0.9239	0.9381	0.9814
LSTM	0.7576	0.7646	0.7501	0.7733	0.8060	0.9304
Logistic Regression	0.8384	0.8454	0.8290	0.8439	0.8731	0.9536
Ensemble Model	0.8485	0.8573	0.8421	0.8548	0.8832	0.9805

Source: Author, (2026).

Table 2: Summary results of the metric student performance math dataset.

Models	Accuracy	F1 Macro	Precision Macro	Precision Weighted	Recall Macro	Roc auc ovr
XGBoost	0.8650	0.8340	0.8337	0.8659	0.8346	0.9502
LSTM	0.8282	0.8026	0.7805	0.8427	0.8358	0.9411
Logistic Regression	0.7791	0.7452	0.7276	0.7898	0.7694	0.9302
Ensemble Model	0.8650	0.8346	0.8285	0.8675	0.8415	0.9521

Source: Author, (2026).

Table 3: Summary results of the metric student performance joint dataset.

Models	Accuracy	F1 Macro	Precision Macro	Precision Weighted	Recall Macro	Roc auc ovr
XGBoost	0.8646	0.7632	0.7677	0.8647	0.7591	0.9564
LSTM	0.7396	0.6773	0.6451	0.7958	0.7613	0.8882
Logistic Regression	0.8125	0.7540	0.7253	0.8521	0.8337	0.9316
Ensemble Model	0.8542	0.8016	0.7680	0.8757	0.8636	0.9521

Source: Author, (2026).

Table 4: Summary results of the metric student performance merge dataset.

Models	Accuracy	F1 Macro	Precision Macro	Precision Weighted	Recall Macro	Roc auc ovr
XGBoost	0.8621	0.8541	0.8518	0.8630	0.8567	0.9422
LSTM	0.8046	0.7961	0.7766	0.8173	0.8258	0.9388
Logistic Regression	0.8582	0.8513	0.8399	0.8620	0.8652	0.9501
Ensemble Model	0.8659	0.8578	0.8535	0.8668	0.8624	0.9524

Source: Author, (2026).

Table 5: Proposed method vs. other studies.

Study	Dataset Size	Model	Accuracy (%)
Proposed Method	student-combined-join =1044	XGBoost+LSTM+LR	86.5
	student-merged = 382		85.4
	student-mat = 395		86.5
	student-por =649		84.8
Kesgin et al., 2025 [12]	395	XGBoost + SMOTE + Adversarial Debiasing	87.5
Chakraborty et al., 2022[18]	1200	Logistic Regression	82.3
Zhang et al., 2023[19]	850	Random Forest + Fair-SMOTE	85.6

Source: Author, (2026).

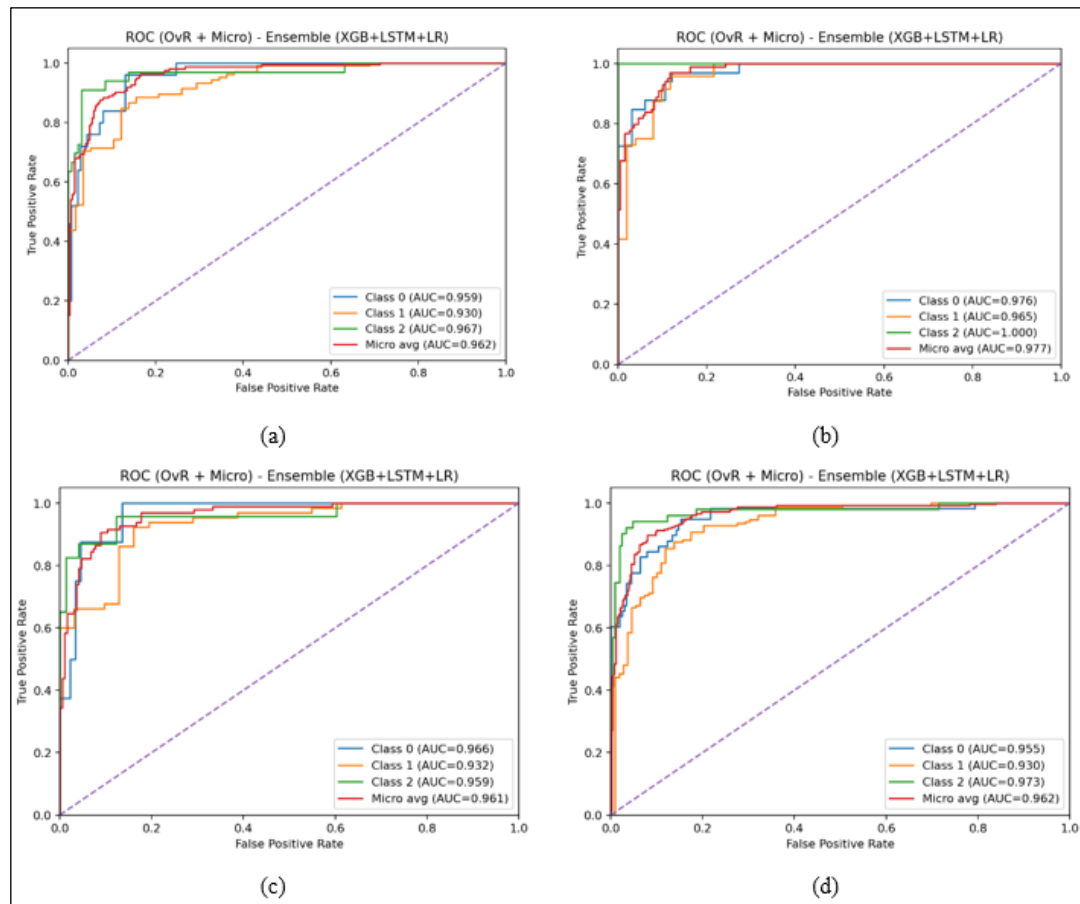


Figure 2: (a)student-mat, (b)student-por, (c)student-merged, (d)student-combined-join.

Source: Author, (2026).

V. CONCLUSIONS

The study apply the ML algorithms into different length of datasets to analyze data for Students' performance which depends on many factors, the algorithms tends to find a pattern in the dataset that helps them with the identification of students' performance output, which tends to improve in direct correlation with how much they engage with teaching methods conducted by the teacher, such as a more engaging learning environment, which is vital for their academic success. The results showed that student-combined-join with 1044 records had an accuracy of 86.5, student-merged with 382 records produced an accuracy of 84.5, student-math dataset with 395 records generated an accuracy of 86.5, and finally student-Portuguese with 649 records had an accuracy of 84.8. The model demonstrates good performance across all models, suggesting that using the correct combination of models results in correct classification across all metrics. The comparative ROC (OvR + micro-average) analysis demonstrates that the proposed ensemble model (XGBoost + LSTM + Logistic Regression) achieves consistently strong discriminative performance across all evaluated dataset variants (student-mat, student-por, student-merged, and student-combined-join). Reinforcing the topic that newly acquired knowledge is an essential aspect of effective learning, ensuring that students not only understand the material but can also apply it confidently in practical situations.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Manocher C. Alipour.
Methodology: Manocher C. Alipour.
Investigation: Manocher C. Alipour.
Discussion of results: Manocher C. Alipour.
Writing – Original Draft: Manocher C. Alipour.
Writing – Review and Editing: Manocher C. Alipour.
Resources: Manocher C. Alipour.
Supervision: Manocher C. Alipour.
Approval of the final text: Manocher C. Alipour.

VII. ACKNOWLEDGMENTS

The authors extend their sincere gratitude to Isabela State University, Angadanan Campus, for their invaluable support and guidance throughout the conduct of this research. Their encouragement and resources were instrumental in the successful completion of this study.

VIII. REFERENCES

- [1] E. Kalita et al., "Predicting student academic performance using Bi-LSTM: a deep learning framework with SHAP-based interpretability and statistical validation," *Front. Educ.*, vol. 10, Jun. 2025, doi: 10.3389/educ.2025.1581247.
- [2] Q. Ni, Y. Zhu, L. Zhang, X. Lu, and L. Zhang, "Leverage Learning Behaviour Data for Students' Learning Performance Prediction and Influence Factor Analysis," *IEEE Trans. Artif. Intell.*, vol. 5, no. 5, pp. 2422–2433, 2024, doi: 10.1109/TAI.2023.3320118.
- [3] J. Heaton, "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618," *Genet. Program. Evolvable Mach.*, vol. 19, Oct. 2017, doi: 10.1007/s10710-017-9314-z.
- [4] W. Zhao, Z. Cao, and H. Liao, "The Effects of Augmented Reality Learning Experiences on Students' Academic Performance: A Meta-Analysis Review," in *2022 8th International Conference on Information Management (ICIM)*, 2022, pp. 25–31, doi: 10.1109/ICIM56520.2022.00012.
- [5] T. O. Soyoye, T. Chen, R. Hill, and K. McCabe, "Predicting Academic Performance of University Students Using Machine Learning: A Case Study in the UK," in *2023 IEEE International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 2023, pp. 431–434, doi: 10.1109/WI-IAT59888.2023.00071.
- [6] G. Li, J. Cui, H. Fu, and Y. Sun, "Light GBM and GA Based Algorithm for Predicting and Improving Students' Performance in Higher Education in the Context of Big Data," in *2024 7th International Conference on Education, Network and Information Technology (ICENIT)*, 2024, pp. 1–5, doi: 10.1109/ICENIT61951.2024.00009.
- [7] P. Cortez, "Student Performance." 2008.
- [8] S. M. Al-Selwi et al., "RNN-LSTM: From applications to modeling techniques and beyond—Systematic review," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 5, p. 102068, 2024, doi: <https://doi.org/10.1016/j.jksuci.2024.102068>.
- [9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [10] J. Ke, H. Zheng, H. Yang, and X. Michael, "Short-term forecasting of passenger demand under on-demand ride services : A spatio-temporal deep learning approach," *Transp. Res. Part C*, vol. 85, no. June, pp. 591–608, 2017, doi: 10.1016/j.trc.2017.10.016.
- [11] M. R. d. H. Maia, A. Plastino, and A. A. Freitas, "An Ensemble of Naive Bayes Classifiers for Uncertain Categorical Data," in *2021 IEEE International Conference on Data Mining (ICDM)*, 2021, pp. 1222–1227, doi: 10.1109/ICDM51629.2021.00148.
- [12] K. Kesgin, S. Kiraz, S. Kosunalp, and B. Stoycheva, "Beyond Performance: Explaining and Ensuring Fairness in Student Academic Performance Prediction with Machine Learning," *Applied Sciences*, vol. 15, no. 15, p. 8409, 2025, doi: 10.3390/app15158409.
- [13] C. Yu et al., "Gradient Boosting Decision Tree with LSTM for Investment Prediction," in *2025 5th Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*, 2025, pp. 57–62, doi: 10.1109/ACCTCS66275.2025.00017.
- [14] C. Sheng and H. Yu, "An optimized prediction algorithm based on XGBoost," in *2022 International Conference on Networking and Network Applications (NaNA)*, 2022, pp. 1–6, doi: 10.1109/NaNA56854.2022.00082.
- [15] 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.
- [16] R. Lin, "Comment Texts Sentiment Analysis Based on Improved Bi-LSTM and Naive Bayes," in *2022 International Conference on Data Analytics, Computing and Artificial Intelligence (ICDAI)*, 2022, pp. 407–412, doi: 10.1109/ICDAI57211.2022.00087.
- [17] J. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *Ann. Stat.*, vol. 29, Nov. 2000, doi: 10.1214/aos/1013203451.
- [18] J. Chakraborty, S. Majumder, and H. Tu, "Fair-SSL: Building fair ML Software with less data," in *Proceedings of the 2nd international workshop on equitable data and technology*, 2022, pp. 1–8.
- [19] F. Zhang, W. Xing, and C. Li, "Predicting students' algebra i performance using reinforcement learning with multi-group fairness," in *LAK23: 13th International Learning Analytics and Knowledge Conference*, 2023, pp. 657–662.