Contents lists available at JECA

Journal of Education and Computer Applications

journal homepage: https://jeca.aks.or.id/index.php/jeca/index

Predict Students' Dropout and Academic Success with XGBoost

Achmad Ridwan[a], Arif Mudi Priyatno[b], Lidya Ningsih[c]
[a]*Universitas Muhammadiyah Kudus, Jawa Tengah 59316, Indonesia*
[b]*Universitas Pahlawan Tuanku Tambusai, Riau, Indonesia*
[c]*Universitas Telkom, Bandung, Jawa Barat, Indonesia*

Article Info

Keywords:

Student Dropout,
Academic Prediction,
XGBoost,
Education Data Analysis,
Strategic Interventions

ABSTRACT

The attrition rate of students in higher education is a worldwide issue that profoundly affects both individuals and institutions. Students who fail to complete their studies often encounter economic and social difficulties, while educational institutions suffer a deterioration in reputation and operational efficacy. This paper proposes the creation of a prediction model utilizing the XGBoost algorithm to assess students' academic progress and dropout risk. The model incorporates several elements, such as academic, demographic, and socio-economic, to yield comprehensive insights into students' educational trends. This research utilizes the Predict Students' Dropout and Academic Success dataset, comprising 4,424 data points and 36 attributes. The data underwent normalization via StandardScaler and was divided into five scenarios for training and testing, ranging from a 50:50 to a 90:10 split. The evaluation of the model was conducted utilizing accuracy, precision, recall, and F1-Score criteria. The findings indicate that the model attains peak performance in the 80:20 scenario, exhibiting 88% precision and an 81% F1-Score, signifying an ideal equilibrium between predictive accuracy and risk identification capability. This study demonstrates that XGBoost can serve as a dependable predictive instrument to aid decision-making in the education sector. These findings establish a foundation for formulating targeted interventions aimed at enhancing student retention. Subsequent study may investigate the use of real-time data and sophisticated models to enhance predictive accuracy.

Corresponding Author:

Achmad Ridwan
University of Muhammadiyah Kudus
Jl. Ganesha Raya No. I, Purwosari, Kec. Kota Kudus, Kabupaten Kudus, Jawa Tengah 59316, Indonesia
Email: achmadridwan@umkudus.ac.id

1. INTRODUCTION

The dropout rate in higher education has emerged as a global issue, significantly affecting both individuals and institutions. Students who fail to complete their education frequently encounter economic and social difficulties, such as diminished employment prospects and decreased income. High dropout rates can

undermine the reputation and operational effectiveness of educational institutions [1]. A study indicates that the average dropout rate among undergraduate students in 23 OECD nations is 33%, underscoring the necessity to tackle this problem [2].

Dropout rates are influenced by numerous academic, social, and economic factors [3]. Studies indicate that pupils from low socioeconomic origins are more susceptible to dropout, hence intensifying social inequality. Furthermore, students encountering academic challenges or lacking social support are at an increased risk of failing to complete their education [4]. Comprehending and tackling these issues is essential for enhancing student retention and guaranteeing their academic achievement [5]. To mitigate elevated dropout rates among students, educational institutions must use a proactive strategy that utilizes data analysis to identify at-risk individuals. A data-driven methodology enables institutions to examine patterns and trends that may remain obscured by conventional techniques, facilitating timely actions that can enhance student retention [6]–[10].

The application of machine learning methodologies in educational data analysis has created new possibilities for forecasting academic achievement and dropout risk [11]. Research findings indicate that machine learning and deep learning techniques have been extensively utilized to forecast student attrition. The research [12] examined machine learning and deep learning techniques for forecasting student attrition. Research [13] anticipated dropout using balanced data. The findings demonstrate that success escalates with data balancing. The research [14] employs critical factors including attendance rates, academic performance, and students' social circumstances to forecast educational outcomes. The research [11] employs this dataset to assess the correlation between demographic variables and student achievement. Furthermore, Research [15] illustrates that integrating academic data with student behavior enhances predictive accuracy. Research [16], [17] indicate that machine learning models can amalgamate diverse educational data to yield profound insights into student learning behaviors. Nevertheless, despite the application of diverse strategies to utilize these datasets, certain obstacles persist. Many studies primarily concentrate on a singular facet, such as academic achievement or the likelihood of educational failure, without amalgamating both into a holistic forecasting framework. Moreover, many methodologies employed continue to encounter difficulties in forecast accuracy and outcome interpretation to facilitate educational decision-making. These problems underscore the necessity for the creation of more inventive models to deliver superior and more advantageous predictive solutions for educational institutions.

In this paper, we propose predicting students' dropout and academic success with XGBoost. this research is expected to provide a more accurate predictive model, offer deeper insights into student education patterns, and lay the groundwork for the development of more strategic interventions. The main contribution of this research is to provide new solutions that can enhance data-driven decision-making in the education sector.

2. RESEARCH METHOD

2.1. Data

The Predict Students' Dropout and Academic Success dataset from the UCI Machine Learning Repository comprises 4,423 data entries, each representing a student with diverse academic, demographic, and socio-economic characteristics. The goal labels in this dataset are classified into two primary categories: Academic Success and Dropout. Among the complete data, 3,003 students are classified under the Academic Success category, whereas 1,421 students are categorized as Dropouts. This distribution establishes a robust basis for training a classification model, despite a minor imbalance between the two categories[18].

This dataset comprises 36 features, offering extensive and detailed insights into the factors that may affect student educational outcomes. The features can be classified into three primary categories. Demographic characteristics encompass details like 'Marital Status', 'Nationality', 'Gender', 'Age at enrollment', 'Mother's qualification', and 'Father's qualification'. The academic features encompass 'Admission grade', 'Curricular units 1st sem (credited)', 'Curricular units 1st sem (enrolled)', 'Curricular units 1st sem (approved)', 'Curricular units 2nd sem (credited)', and 'Curricular units 2nd sem (grade)'. In the meantime, socio-economic characteristics encompass information like 'unemployment rate', 'inflation rate', and 'GDP'.

Every element in the dataset plays a distinct role in the analysis. For instance, attributes like 'Admission grade' and 'Curricular units' offer direct insights into students' academic performance, whereas attributes such as 'Mother's qualification' and 'Father's occupation' can help in understanding the impact of family background on students' educational outcomes. Furthermore, macroeconomic variables like the

'Unemployment rate' and 'Inflation rate' enable an evaluation of the influence of external factors on education.

2.2. Normalization

Normalization is a crucial step in data preprocessing, ensuring that all features in the dataset are on a comparable scale [19], [20]. This study employs normalization via the StandardScaler method, implemented through the scikit-learn library. The StandardScaler operates by adjusting each feature in the dataset to achieve a mean of $\mu = 0$ and a standard deviation of $\sigma = 1$. This approach aims to minimize the effects of varying scales among features that exhibit notably different value ranges, such as the 'Admission grade' compared to binary variables like 'Debtor'. The normalization is conducted using Equation 1 [21].

$$z = \frac{z-\mu}{\sigma} \quad (1)$$

The formula defines z as the normalized feature value, x as the original feature value, μ as the mean of the feature, and σ as the standard deviation of the feature. This transformation guarantees that features with significant scales do not overshadow the model learning process, enabling all features to contribute equally in data analysis.

The normalization process is conducted in multiple phases. The dataset is initially partitioned into two segments: the training data and the testing data. This division focuses on training the model with the majority of the available data, while also ensuring that independent data is utilized to assess the model's performance effectively. Subsequently, the parameters for normalization, specifically the mean (μ) and standard deviation (σ), are computed using the training data for each individual feature. This approach guarantees that the normalization transformation is solely affected by the data utilized during model training, thus preventing any information leakage from the test data.

Once the parameters are computed, the training data undergoes normalization by deducting the mean of each feature and dividing by the corresponding standard deviation. This process leads to the training data features exhibiting a distribution characterized by a mean of 0 and a standard deviation of 1. The test data is subsequently normalized utilizing the parameters (μ and σ) derived from the training data, thereby maintaining consistency in the transformation process between the training and test datasets. This method allows for the processing of all features in the dataset on a uniform scale, thereby enhancing both the efficiency and accuracy of the machine learning model. This normalization allows the model to identify patterns in the data without interference from features that have larger value ranges. This method further ensures the integrity of model evaluation, as the test data remains unaffected by any information obtained from the normalization process.

2.3. Data Splitting

Splitting data is a crucial phase in the development of a machine learning model [22]. This study involves dividing the dataset into two primary components: the training set and the testing set [23]. The proportions allocated for training range from 50% to 90%, while those for testing vary from 50% to 10%. This step is implemented to guarantee that the model can effectively learn from the bulk of the accessible data, while also supplying independent data to evaluate the performance of the trained model.

The model is constructed and refined using training data. At this stage, algorithms analyze the provided data to identify patterns that enable them to forecast target labels using the available features. This data includes a significant portion of the dataset, providing the model with sufficient information to comprehend the relationship between features and labels. Test data serves the purpose of assessing the model's performance following its training phase. This phase focuses on evaluating the model's capacity to generalize previously unseen data. A strong generalization demonstrates that the model excels not only with the training data but also possesses the ability to make precise predictions in real-world scenarios.

The dataset was spliting using the scikit-learn library, with the train_test_split function. This function randomly splits the dataset into two subsets while maintaining a balanced label distribution in both subsets. This technique is called stratified splitting, which ensures that the proportion between the categories of Academic Success and Dropout remains consistent in both the training and testing data.

The process of data splitting involves utilizing the initial dataset, which has been subjected to preprocessing stages like cleaning and normalization, as input for the train_test_split function. The data is systematically partitioned into two subsets, with 50% to 90% allocated for training and 50% to 10% designated for testing. The distribution of labels within the dataset is analyzed to confirm that the ratio between Academic Success and Dropout is preserved.

This systematic approach to data splitting allows for an objective evaluation of the model, facilitating an assessment of its effectiveness in predicting student educational outcomes. A balanced allocation of training and testing data is crucial in avoiding overfitting, a scenario where the model excels on the training set yet struggles with unseen data. An effective data splitting process guarantees that the evaluation outcomes accurately represent the model's capability to function in practical scenarios.

2.4. XGBoost

Extreme Gradient Boosting (XGBoost) is a machine learning algorithm based on decision trees, specifically engineered to enhance the boosting process while ensuring high computational efficiency and exceptional prediction accuracy [24]–[26]. XGBoost represents a sophisticated implementation of the advanced gradient boosting algorithm, designed to effectively manage large datasets, imbalanced features, and diverse input data types. The construction of the model in XGBoost occurs through an iterative process, where new decision trees are introduced to reduce the prediction error generated by the preceding trees. XGBoost employs an ensemble approach, integrating multiple weak learners to create a robust model. (highly capable individual). This model demonstrates exceptional capability in managing non-linear relationships among features and offers the versatility needed to tackle a range of data situations. Furthermore, XGBoost incorporates regularization within its objective function to mitigate overfitting. The primary equation of XGBoost is represented as Equation 2.

$$L(\phi) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (2)$$

Where $L(\phi)$ represents the comprehensive objective function. The loss function $(y_i, \hat{y}_i)$ quantifies the discrepancy between the predicted value $(\hat{y}_i)$ dan and the actual value (y_i). $\Omega(f_k)$ serves as the regularization term aimed at managing the complexity of the model. f_k represents the decision tree at the k-th iteration, while K denotes the total number of decision trees. At this point, the regularization function in XGBoost is expressed through equation 3.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \quad (3)$$

Where γT is the penalty based on the number of leaves (T) on the tree. $\lambda \|w\|^2$ is the penalty for the magnitude of the weights on the tree leaves (w). γ and λ are regularization parameters to control overfitting.

The steps of XGBoost begin with model initialization. The initialization process starts by determining the loss function $(l(y_i, \hat{y}_i))$ and hyperparameters such as the number of trees (K), maximum tree depth, and regularization parameters (γ and λ). Then the stage of building the First Tree. The first model is built by initializing the prediction $(\hat{y}_i)$ as the average target value. The first tree is designed to minimize the loss function based on the residuals between the actual value (y_i) and the initial prediction. Next is the stage of adding a New Tree. At each iteration, a new tree is added to correct the errors of the previous tree. This tree is created by minimizing the gradient of the loss function. Then comes the regularization stage, where each added tree is refined with regularization to control the model's complexity. This prevents the model from becoming too complex, which can lead to overfitting. Finally, the final prediction stage is calculated by summing the contributions from all the decision trees that have been built. Equation 4 calculates the contribution of all trees. Where x_i is the input feature for the i-th data.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \quad (4)$$

2.5. Evaluation

Evaluating a model is a crucial phase in the machine learning workflow, as it assesses the model's performance on previously unseen data, specifically the test data [27]. This study evaluates the model's performance through four primary metrics: accuracy, recall, precision, and F1-score. The metrics offer an in-depth analysis of the model's performance in addressing classification challenges, particularly with datasets that might exhibit an uneven label distribution.

The metric of accuracy is utilized to evaluate the ratio of correct predictions in relation to the complete dataset. The accuracy metric offers a comprehensive assessment of the model's effectiveness in categorizing all labels. The calculation of accuracy is represented by Equation 5 [28]. Recall is a metric that evaluates the model's capacity to identify all instances that genuinely belong to the positive class. The importance of recall is paramount in situations where the inability to identify the positive class can lead to significant repercussions. Recall serves as a measure of the model's ability to identify all instances of actual positive data. Equation 6 provides a method for calculating recall [23].

Precision serves as a metric that evaluates the accuracy of a model's positive predictions, focusing on the relevance of those predictions within the positive class. Accuracy is crucial in contexts where it is essential to reduce the occurrence of false positives. Precision offers a clear understanding of the reliability of the model's predictions for the positive class. Equation 7 provides a method for calculating precision [29]. The F1-Score serves as a metric that integrates precision and recall into a unified harmonic value. This metric proves to be highly beneficial in scenarios characterized by an imbalance between positive and negative classes. The F1-Score strikes a balance between precision and recall, serving as a suitable metric for assessing model performance on imbalanced datasets. Equation 8 provides a method for calculating the F1-score [23].

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$recall = \frac{TP}{TP+FN} \quad (6)$$

$$precision = \frac{TP}{TP+FP} \quad (7)$$

$$f1-score = 2\ x\ \frac{precision\ x\ recall}{precision+\ recall} \quad (8)$$

In this context, TP (True Positive) refers to the count of accurate predictions made for the positive class. True Negative (TN) refers to the count of accurate predictions made for the negative class. False Positive (FP) refers to the count of incorrect predictions where the negative class is mistakenly identified as positive. FN (False Negative) refers to the count of erroneous predictions where instances of the positive class are incorrectly classified as negative.

3. RESULTS AND DISCUSSION

The dataset for predicting student dropout and academic success comprises 4,424 entries, featuring 37 columns that encapsulate a range of academic, demographic, and socio-economic details for each student. This dataset aims to investigate the elements that affect students' educational achievements, specifically within the Graduate and Dropout groups. The dataset's target labels are classified into two primary categories: Graduate and Dropout. Graduates signify individuals who have effectively fulfilled their academic requirements. Dropout refers to individuals who fail to finish their educational programs. Figure 1 illustrates the distribution of these labels, revealing an imbalanced distribution between the two categories within the dataset. While variations in quantity exist, they are not substantial enough to influence the model training process.

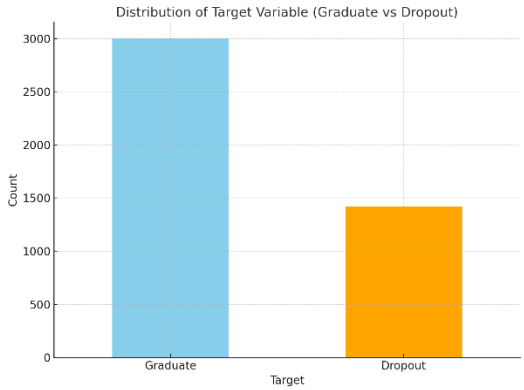

Figure 1. Distribution of Target class Graduate and dropout

The dataset comprises three primary categories of features: Demographic Features, Academic Features, and Socio-Economic Features. Demographic features encompass details like marital status, nationality, gender, age at enrollment, and the educational attainment of parents. Academic features encompass information including entry grades, the total number of courses undertaken, the count of approved courses, and the respective course grades. Socio-Economic Features encompass variables like the unemployment rate, inflation rate, and GDP (Gross Domestic Product). The data presented in Table 1 encompasses the three primary categories of features.

Table 1. Feature classification on dataset

Demographic Features	Academic Features	Socio-Economic Features
Marital Status	Admission grade	GDP
Nacionality	Curricular units 1st sem (credited	Inflation rate
Gender	Curricular units 1st sem (enrolled)	Unemployment rate
Age at enrollment	Curricular units 1st sem (approved)	
Mother's qualification	Curricular units 2nd sem (credited)	
Father's qualification	Curricular units 2nd sem (grade)	

This dataset is complete, with no missing values across all features, eliminating the need for any imputation or value replacement processes. This streamlines the following pre-processing stages, including normalization and dataset division. Normalization is achieved through the application of StandardScaler. The division of the dataset ranges from 50% to 90% allocated for training data, while testing data comprises 50% to 10%. The data presented in Table 2 illustrates the division between the training and testing datasets.

Table 2. Splitting of training data and test data

Data Splitting	Training Data	Test Data
50:50	2212	2212
60:40	2654	1770
70:30	3096	1328
80:20	3539	885
90:10	3981	443

The model was evaluated through five distinct training and testing data split scenarios: 50:50, 60:40, 70:30, 80:20, and 90:10. Each scenario underwent assessment through four primary metrics: accuracy, precision, recall, and F1-Score, offering a thorough understanding of the model's effectiveness in addressing classification challenges. The findings from the evaluation are presented in Table 3. The model demonstrates consistent accuracy across nearly all data split scenarios, achieving a peak value of 87% in the 60:40, 70:30, and 80:20 splits. The findings indicate that the model achieves optimal performance with a greater amount of training data, while also maintaining stability in its performance with smaller training data proportions (50:50 and 90:10).

Table 3. The performance of evaluation

Data Splitting	Accuracy	Precision	Recall	F1-Score
50:50	86	84	73	78
60:40	87	84	74	79
70:30	87	86	73	79
80:20	87	88	74	81
90:10	86	80	76	78

The utmost precision was attained in the 80:20 scenario, reaching a value of 88%, which signifies that the model demonstrates a high level of reliability in classifying pertinent positive predictions. The enhancement in accuracy in this context suggests that the reduced amount of testing data yields more targeted predictions for the positive category. The maximum recall achieved was 76%, observed in the 90:10 data split. This suggests that even with a greater amount of testing data, the model successfully identifies a higher number of true positive instances. Nonetheless, the somewhat reduced recall values in alternative scenarios can be balanced by increased precision. The F1-Score, indicating the equilibrium between precision and recall, demonstrated optimal performance in the 80:20 scenario, achieving a value of 81%. The integration of high precision with sufficient recall enhances the model's effectiveness in managing the utilized data distribution in this context.

The evaluation results indicate that the model performance remains consistent across different data split scenarios, with accuracy levels between 86% and 87%. The improvement in precision and F1-Score in the 80:20 scenario suggests that the model demonstrates enhanced performance with a sufficiently large training dataset, while also achieving solid evaluation results on the testing data. In the 90:10 split, while the recall rises to 76%, the precision drops to 80%, suggesting that the model may experience a slight decline in its accuracy when identifying positive data. This underscores the necessity of sustaining equilibrium between training and testing data to attain peak performance. The findings suggest that utilizing a greater share of training data, like an 80:20 ratio, enhances precision and F1-Score, positioning it as a favorable option for implementation. Educational institutions have the opportunity to implement this model for early intervention with at-risk students, ensuring high accuracy and reducing the occurrence of false positives and false negatives in predictions.

4. CONCLUSION

This investigation seeks to tackle the significant challenge of student attrition in higher education through the creation of a predictive model utilizing the XGBoost algorithm. The model effectively combines academic, demographic, and socio-economic factors to forecast academic success and the likelihood of dropout. This study utilizes the Predict Students' Dropout and Academic Success dataset, which comprises 4,424 data points and 36 features, to illustrate the capabilities of machine learning in examining intricate educational data. The findings indicate that the model demonstrates reliable performance across various data split scenarios, achieving an accuracy between 86% and 87%. The optimal results were obtained in the 80:20 data split scenario, showcasing a precision of 88% and an F1-Score of 81%. The equilibrium between precision and recall illustrates the model's capability to recognize at-risk students while minimizing the occurrence of false positives. The analysis indicates that factors like admission grade, course performance, and socio-economic conditions significantly influence the prediction outcomes. This study utilizes the strengths of XGBoost to offer a dependable and interpretable predictive tool for educational institutions, enabling them to proactively identify students who may be at risk of dropping out. The results highlight the critical role of utilizing data to inform decisions aimed at enhancing student retention and educational success. Future investigations may delve into the application of real-time data and advanced algorithms to enhance prediction precision.

REFERENCES

[1] D. K. Dake and C. Buabeng-Andoh, "Using Machine Learning Techniques to Predict Learner Drop-out Rate in Higher Educational Institutions," *Mob. Inf. Syst.*, vol. 2022, pp. 1–9, Nov. 2022, doi: 10.1155/2022/2670562.

[2] C. Bargmann, L. Thiele, and S. Kauffeld, "Motivation matters: predicting students' career decidedness and intention to drop out after the first year in higher education," *High. Educ.*, vol. 83, no. 4, pp. 845–861, Apr. 2022, doi: 10.1007/s10734-021-00707-6.

[3] C. Aina, E. Baici, G. Casalone, and F. Pastore, "The determinants of university dropout: A review of the socio-economic literature," *Socioecon. Plann. Sci.*, vol. 79, p. 101102, Feb. 2022, doi: 10.1016/j.seps.2021.101102.

[4] R. Z. Pek, S. T. Ozyer, T. Elhage, T. Ozyer, and R. Alhajj, "The Role of Machine Learning in Identifying Students At-Risk and Minimizing Failure," *IEEE Access*, vol. 11, pp. 1224–1243, 2023, doi: 10.1109/ACCESS.2022.3232984.

[5] J. C. Véliz Palomino and A. M. Ortega, "Dropout Intentions in Higher Education: Systematic Literature Review," *J. Effic. Responsib. Educ. Sci.*, vol. 16, no. 2, pp. 149–158, Jun. 2023, doi: 10.7160/eriesj.2023.160206.

[6] A. Namoun and A. Alshanqiti, "Predicting Student Performance Using Data Mining and Learning Analytics Techniques: A Systematic Literature Review," *Appl. Sci.*, vol. 11, no. 1, p. 237, Dec. 2020, doi: 10.3390/app11010237.

[7] B. Albreiki, N. Zaki, and H. Alashwal, "A Systematic Literature Review of Student' Performance Prediction Using Machine Learning Techniques," *Educ. Sci.*, vol. 11, no. 9, p. 552, Sep. 2021, doi: 10.3390/educsci11090552.

[8] A. Khan and S. K. Ghosh, "Student performance analysis and prediction in classroom learning: A review of educational data mining studies," *Educ. Inf. Technol.*, vol. 26, no. 1, pp. 205–240, Jan. 2021, doi: 10.1007/s10639-020-10230-3.

[9] J. L. Rastrollo-Guerrero, J. A. Gómez-Pulido, and A. Durán-Domínguez, "Analyzing and Predicting Students' Performance by Means of Machine Learning: A Review," *Appl. Sci.*, vol. 10, no. 3, p. 1042, Feb. 2020, doi: 10.3390/app10031042.

[10] J. López-Zambrano, J. Lara Torralbo, and C. Romero, "Early Prediction of Student Learning Performance Through Data Mining: A Systematic Review," *Psicothema*, vol. 3, no. 33, pp. 456–465, Aug. 2021, doi: 10.7334/psicothema2021.62.

[11] V. Realinho, J. Machado, L. Baptista, and M. V. Martins, "Predicting Student Dropout and Academic Success," *Data*, vol. 7,

no. 11, p. 146, Oct. 2022, doi: 10.3390/data7110146.
[12] D. Andrade-Girón *et al.*, "Predicting Student Dropout based on Machine Learning and Deep Learning: A Systematic Review," *ICST Trans. Scalable Inf. Syst.*, vol. 10, no. 5, pp. 1–11, Jul. 2023, doi: 10.4108/eetsis.3586.
[13] N. Mduma, "Data Balancing Techniques for Predicting Student Dropout Using Machine Learning," *Data*, vol. 8, no. 3, p. 49, Feb. 2023, doi: 10.3390/data8030049.
[14] K. Yan, "Student Performance Prediction Using XGBoost Method from A Macro Perspective," in *2021 2nd International Conference on Computing and Data Science (CDS)*, Jan. 2021, pp. 453–459. doi: 10.1109/CDS52072.2021.00084.
[15] J. Mun and M. Jo, "Applying machine learning-based models to prevent University student dropouts," *Korean Soc. Educ. Eval.*, vol. 36, no. 2, pp. 289–313, Jun. 2023, doi: 10.31158/JEEV.2023.36.2.289.
[16] H. Huo *et al.*, "Predicting Dropout for Nontraditional Undergraduate Students: A Machine Learning Approach," *J. Coll. Student Retent. Res. Theory Pract.*, vol. 24, no. 4, pp. 1054–1077, Feb. 2023, doi: 10.1177/1521025120963821.
[17] D. Duan, C. Dai, and R. Tu, "Research on the Prediction of Students' Academic Performance Based on XGBoost," in *2021 Tenth International Conference of Educational Innovation through Technology (EITT)*, Dec. 2021, pp. 316–319. doi: 10.1109/EITT53287.2021.00068.
[18] M. V. Martins, D. Tolledo, J. Machado, L. M. T. Baptista, and V. Realinho, "Early Prediction of student's Performance in Higher Education: A Case Study," in *Trends and Applications in Information Systems and Technologies*, 2021, pp. 166–175. doi: 10.1007/978-3-030-72657-7_16.
[19] D. Singh and B. Singh, "Feature wise normalization: An effective way of normalizing data," *Pattern Recognit.*, vol. 122, p. 108307, Feb. 2022, doi: 10.1016/j.patcog.2021.108307.
[20] A. M. Priyatno, "Spammer Detection Based on Account, Tweet, and Community Activity on Twitter," *J. Ilmu Komput. dan Inf.*, vol. 13, no. 2, pp. 97–107, Jul. 2020, doi: 10.21609/jiki.v13i2.871.
[21] L. B. V. de Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, "The choice of scaling technique matters for classification performance," *Appl. Soft Comput.*, vol. 133, p. 109924, Jan. 2023, doi: 10.1016/j.asoc.2022.109924.
[22] Y. Matsubara, M. Levorato, and F. Restuccia, "Split Computing and Early Exiting for Deep Learning Applications: Survey and Research Challenges," *ACM Comput. Surv.*, vol. 55, no. 5, pp. 1–30, May 2023, doi: 10.1145/3527155.
[23] A. M. Priyatno and F. I. Firmananda, "N-Gram Feature for Comparison of Machine Learning Methods on Sentiment in Financial News Headlines," *RIGGS J. Artif. Intell. Digit. Bus.*, vol. 1, no. 1, pp. 01–06, Jul. 2022, doi: 10.31004/riggs.v1i1.4.
[24] T. Kavzoglu and A. Teke, "Predictive Performances of Ensemble Machine Learning Algorithms in Landslide Susceptibility Mapping Using Random Forest, Extreme Gradient Boosting (XGBoost) and Natural Gradient Boosting (NGBoost)," *Arab. J. Sci. Eng.*, vol. 47, no. 6, pp. 7367–7385, Jun. 2022, doi: 10.1007/s13369-022-06560-8.
[25] T. Kavzoglu and A. Teke, "Advanced hyperparameter optimization for improved spatial prediction of shallow landslides using extreme gradient boosting (XGBoost)," *Bull. Eng. Geol. Environ.*, vol. 81, no. 5, p. 201, May 2022, doi: 10.1007/s10064-022-02708-w.
[26] A. Asselman, M. Khaldi, and S. Aammou, "Enhancing the prediction of student performance based on the machine learning XGBoost algorithm," *Interact. Learn. Environ.*, vol. 31, no. 6, pp. 3360–3379, Aug. 2023, doi: 10.1080/10494820.2021.1928235.
[27] M. M. Taye, "Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions," *Computers*, vol. 12, no. 5, p. 91, Apr. 2023, doi: 10.3390/computers12050091.
[28] A. M. Priyatno and L. Ningsih, "TF - IDF Weighting to Detect Spammer Accounts on Twitter based on Tweets and Retweet Representation of Tweets," *Sist. J. Sist. Inf.*, vol. 11, no. 3, pp. 614–622, 2022, [Online]. Available: http://sistemasi.ftik.unisi.ac.id/index.php/stmsi/issue/view/46
[29] M. R. A. Prasetya and A. M. Priyatno, "Dice Similarity and TF-IDF for New Student Admissions Chatbot," *RIGGS J. Artif. Intell. Digit. Bus.*, vol. 1, no. 1, pp. 13–18, Jul. 2022, doi: 10.31004/riggs.v1i1.5.