Predicting Academic Success and Failure Rate in University Students

1st Tasnuba Islam
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
tasnuba.islam@northsouth.edu

2nd Zarin Tasnim Pushpita
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
zarin.pushpita@northsouth.edu

3rd Rania Noor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
rania.noor@northsouth.edu

4th Maheenul Hoque Chowdhury
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
maheenul.chowdhury@northsouth.edu

***Abstract*—This thesis presents a comprehensive analysis of student dropout and academic success prediction using multiple machine learning techniques. We employed several algorithms, specifically k-Nearest Neighbors (KNN), Decision Tree, Logistic Regression, Support Vector Classifier (SVC), Random Forest, Gaussian Naive Bayes, AdaBoost, and XGBoost. Each model was evaluated based on its predictive performance to identify the most effective approach for the early identification of students at risk of dropping out and those likely to succeed academically. The results of this study have significant implications for educational institutions seeking to improve retention rates and support student achievement through targeted strategies based on predictive analytics [5]. After hyperparameter tuning, Random Forest and Support Vector Classifier (SVC) emerged as the best algorithms, both achieving accuracies of 92%.**

***Index Terms*—Student Dropout, Academic Success Prediction, Machine Learning, KNN, Decision Tree, Logistic Regression, SVC, Random Forest,Gaussian Naive Bayes,AdaBoost, XGBoost**

I. Introduction

Student dropout is a critical issue that poses significant challenges to educational institutions worldwide [7]. The consequences of student dropout extend beyond individual academic performance, impacting future career opportunities, economic stability, and societal development [13]. Thus, accurately predicting student dropout and academic success is essential for devising effective interventions that enhance student retention and overall success.

The primary objectives of this project are twofold: firstly, to evaluate the predictive capabilities of various machine learning models, and secondly, to identify the model that provides the highest accuracy in predicting student outcomes. The machine learning models examined in this study include k-Nearest Neighbors (k-NN), Decision Tree, Logistic Regression, Support Vector Classifier (SVC), Random Forest, AdaBoost, XGBoost, and Gaussian Naive Bayes Classifier. Each model is selected for its distinct strengths and potential to contribute to a comprehensive predictive framework.

To optimize model performance, hyperparameter tuning was meticulously conducted for several models. Hyperparameter tuning involves fine-tuning the parameters to get the best performance and efficiency. This critical step is essential for maximizing the predictive power of machine learning models, particularly in datasets with complex and varied attributes [8].

Among the models evaluated, the Random Forest as well as Support Vector Classifier (SVC) model emerged as the most accurate, achieving a remarkable accuracy of 92%. This underscores its effectiveness in predicting student dropout and academic success. Additionally, the Decision Tree, and Logistic Regression models demonstrated robust performance, each achieving over 91% accuracy. These results highlight the substantial potential of machine learning models to deliver reliable predictions, facilitating timely and targeted intervention strategies.

In the following sections of this paper structural description follows as: Section II describes the literature review, Section III explains the methodology implementation in data pre-processing, model training, and evaluation. Section IV portrays result analyzing. Section V containing the discussion related to the work and Section VI continues with the limitations of the projected work. Finally, Section VII concludes the entire paper.

II. LITERATURE REVIEW

Eric E. Osemwegie [11] in the paper 'Student Dropout prediction using machine learning' acquired their data through five academic sessions that were obtained from

the Department of Computer Science, University of Benin, Nigeria. However, only 906 out of the 947 data could be used after preprocessing and cleaning. Amidst six distinct classifiers including Naive Bayes (NB), Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), K-Nearest Neighbor (KNN), and Artificial Neural Networks (ANN) that were used in this study, Logistic Regression (LR) performed better than all the models examined in this study in terms of accuracy (98.9%), precision (100%), and F1-score. However, SVM Model performed the best after LR Model, yet it required a long time to learn and was not suitable for the prediction system.

Meseret Yihun Amare and Stanislava Šimonová [14] from University of Pardubice, discovered that the Logistic Regression model outperformed the other models in predicting the early dropout rate of students. They collected the dataset from Hawassa University Student Information Systems Portal (HUSIS) and the dataset consisted 13 features. The study predicted early dropouts among the students using several machine learning techniques. The assessment was done based on the f-score, accuracy, precision, and support metrics of different classifiers. Consequently, the algorithm with better performance was used to create the prediction model.

Cameron Gray and Dave Perkins [6] in their study represents Learning Analytics methods for identifying at-risk students at the 3rd week of the semester. The initial experiment utilized Sequential Forward Selection (SFS) and the Nearest Neighbor (1-NN) classifier. Furthermore, they introduced a new attendance statistic and a predictive model with 97% accuracy to the provide the students with supporting mechanisms.

Faraz Moghimi, Michael C. Metzger, John Sears [2] on their 'Predicting Student Dropouts via Machine Learning', evaluated their study by using different machine learning models such as XGBoost, Neural Networks, Bagging, Trees, SVM to predict student persistence a year ahead of time. However, in this study the dataset public research university in a US metropolitan area, the dataset consisted around 50000 observations and 160 features. Therefore, they also greatly emphasized on class imbalance classification and feature selection. After conducting hyperparameter tuning, XGBoost has shown to be the best model, exhibiting high accuracy and True Negative Rate (TNR). The results of feature importance analysis indicate that pre-enrollment static data points do not significantly affect the predictions.

Khalid Oqaidi and colleagues [10] classified the student features into six categories, such as academic features in the current program, previous academic features, socio-demographic features, institutional features, behavioral characteristics, and financial features. The authors assessed the model performance using various performance metrics including F1-Score, Area Under Curve (AUC), Accuracy, Precision and Recall. Moreover, they observed that there is no single algorithm that outperforms others in performance metrics, and the choice of algorithm and performance metrics depend on the specific context and goals of the study. They also identify the absence of a universally applicable model that can be generalized across different educational institutions.

Nabila Sghir and colleagues [3] explored about the use of Predictive Learning Analytics (PLA) in higher education, highlighting its potential to improve learning outcomes and student success. In terms of prediction accuracy Artificial Neural Networks, Random Forest, and Gradient Boosting ranked first, second, and third, respectively, among the prediction techniques applied. The performance of algorithms was evaluated using the confusion matrix and the measurements obtained from it, the Mean Squared Error (MSE), and R-Squared coefficient. They accentuated the need for privacy and security measures in Predictive Learning Analytics (PLA) and suggested the need of future research directions such as data augmentation and transfer learning techniques in this domain

III. METHODOLOGY

This section presents detailed approaches and techniques for analyzing and predicting academic success and failure rates among university students.

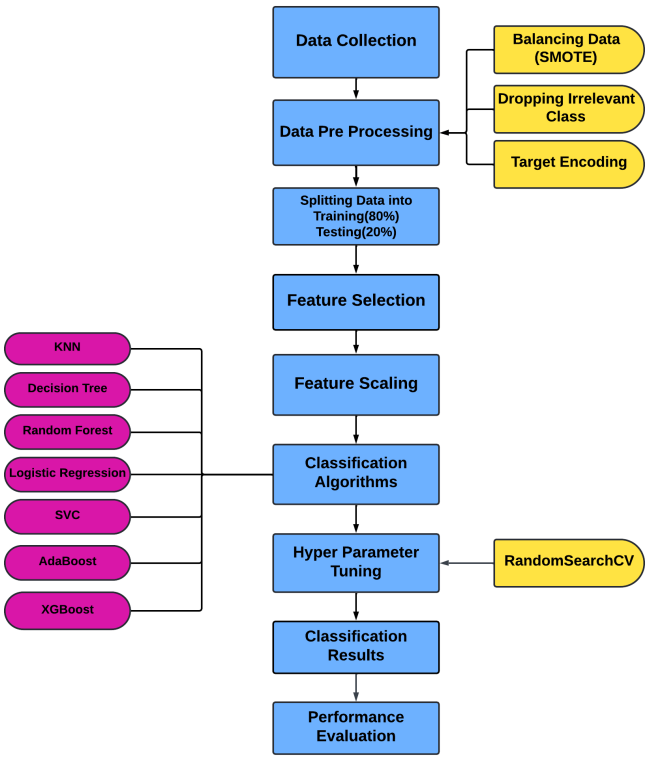


Fig. 1. Proposed Methodology

A. Dataset Acquisition and Description

We have obtained our dataset from the UCI Machine Learning Repository [12], ensuring a reliable and standardized source of data for our analysis. It has details about students from different programs like agronomy, design, education, nursing, journalism, management, social service, and technologies. The data includes academic history, demographics, socioeconomic status, and how students did in first two semesters. The final dataset is available as a CSV file encoded as UTF8 and consists of 4424 records with 37 attributes and contains no missing values. The target variable encompasses three classes: Dropout, Enrolled, and Graduated.

Table I shows the summary of the dataset.

Dataset	**No. of instances**	**No. of Features**	**Target**		
			Dropout	**Enrolled**	**Graduate**
Predict Student's Dropout and Academic Success	4424	37	1421	794	2209

TABLE I
STUDENT DROPOUT AND ACADEMIC SUCCESS DATASET

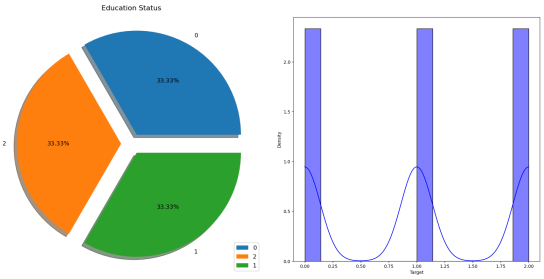
Table II provides an overview of the attributes present in the dataset, organized by category. Additionally, each attribute is accompanied by its respective type.

TABLE II
ATTRIBUTE CLASSES AND TYPES

Class of Attribute	**Attribute**	**Type**
Demographic details	Marital Status	Numeric
	Nationality	Numeric
	Displaced	Categorical
	Gender	Categorical
	Age at Enrollment	Numeric
	International	Categorical
Socioeconomic data	Mother's qualification	Numeric
	Father's qualification	Numeric
	Mother's occupation	Numeric
	Father's occupation	Numeric
	Educational special needs	Categorical
	Debtor	Categorical
	Tuition fees up to date	Categorical
	Scholarship holder	Categorical
Macroeconomic indicators	Unemployment rate	Numeric
	Inflation rate	Numeric
	GDP	Numeric
Academic data at enrollment	Application mode	Numeric
	Application order	Numeric
	Course	Numeric
	Daytime/evening attendance	Categorical
	Previous qualification	Numeric
	Previous qualification (grade)	Numeric
	Admission grade	Numeric
Academic data of 1st semester	Curricular units 1st sem (credited)	Numeric
	Curricular units 1st sem (enrolled)	Numeric
	Curricular units 1st sem (evaluations)	Numeric
	Curricular units 1st sem (approved)	Numeric
	Curricular units 1st sem (grades)	Numeric
	Curricular units 1st sem (without evaluations)	Numeric
Academic data of 2nd semester	Curricular units 2nd sem (credited)	Numeric
	Curricular units 2nd sem (enrolled)	Numeric
	Curricular units 2nd sem (evaluations)	Numeric
	Curricular units 2nd sem (approved)	Numeric
	Curricular units 2nd sem (grades)	Numeric
	Curricular units 2nd sem (without evaluations)	Numeric
Target	Target	Categorical

B. Data Preprocessing

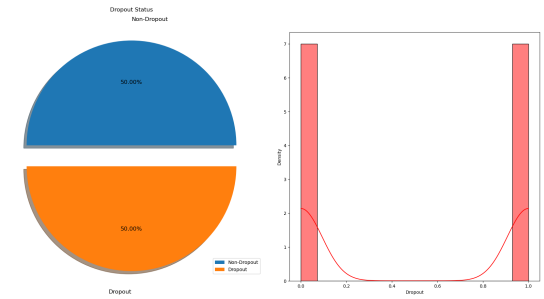
- Balancing Data (SMOTE):After testing the dataset on decision tree, we found that our dataset has some overfitting data. So, in order to balance that we applied 'Synthetic Minority Oversampling Technique' also known as SMOTE. This technique is designed to tackle imbalanced datasets by generating synthetic samples for the minority class that are similar to but not identical to existing minority class samples. It divides between current samples and their k nearest neighbors in the feature space to create new minority class samples and to accomplish this a point on the line segment connecting the two instances in the feature space is arbitrarily selected. Initially, our dataset had 4424 instances and after smote we balanced the dataset by generating 2203 more instance which gave us in total 6627 instances.



(a) Education status: Dropout, Enrolled, Graduated. (b) Target distribution after SMOTE.

Fig. 2. Education status and target distribution.

- Dropping Irrelevant Classes: The target column consists of three classes: Graduate, Dropout, and Enrolled. We are interested in predicting whether a student will dropout or not. Therefore, the number of "Enrolled" students is irrelevant, as all graduates and dropouts are also enrolled. Our focus is solely on determining whether a student graduated or dropped out. Consequently, we are removing the "Enrolled" values and proceeding with the "Graduate" and "Dropout" values for further analysis.



(a) Dropout status distribution: Non-Dropout and Dropout. (b) Class distribution post dropping enrolled class.

Fig. 3. Dropout status: Non-Dropout vs. Dropout. Post-dropping enrolled class distribution.

- Target Encoding: We utilized the LabelEncoder tool from scikit-learn [1] to convert the 'Target' column in the dataset into numerical labels . These labels were assigned as follows: 0 indicates Dropout, 2 represents Graduate, and 1 denotes Enrolled. This transformation allows for the adaptation of machine learning algorithms that necessitate numerical inputs, thereby streamlining model training and assessment processes

C. Data Splitting

In this step, we divided the preprocessed dataset into training and testing data. Specifically, we allocated 80% of the dataset for training our models, while the remaining 20% was reserved for testing their performance.

D. Feature Selection

To enhance the performance of our machine learning model, we conducted feature selection using Pearson's correlation. We first calculated the correlation matrix to determine the linear relationships between features and visualized it using a heatmap to easily identify highly correlated pairs. Features with a correlation coefficient greater than 0.9 were considered redundant. The highly correlated features identified were: "Father's occupation" and "Mother's occupation" (0.901), "Curricular units 2nd sem (credited)" and "Curricular units 1st sem (credited)" (0.947), "Curricular units 2nd sem (enrolled)" and "Curricular units 1st sem (enrolled)" (0.940), and "Curricular units 2nd sem (approved)" and "Curricular units 1st sem (approved)" (0.920). We chose to drop the following features: "Curricular units 2nd sem (approved)", "Curricular units 2nd sem (enrolled)", "Father's occupation", and "Curricular units 2nd sem (credited)". This preprocessing step ensured that our model was trained on a more independent and less redundant set of features, potentially improving its performance and generalization.

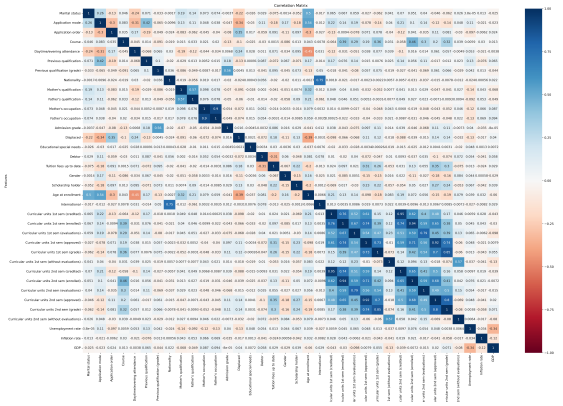


Fig. 4. Pearson's Correlation Matrix

E. Feature Scaling

We applied the StandardScaler to normalize the features in both the training and testing datasets. This process ensures that all features have a mean of 0 and a standard deviation of 1, which is crucial for enhancing the performance of

certain machine learning algorithms, especially those sensitive to varying feature scales.

F. Classification Algorithms

We trained eight models – K-Nearest Neighbor, Decision Tree, Logistic Regression, Support Vector Machine, Random Forest, Gaussian Naive Bayes, AdaBoost and XGBoost after dropping the 'Enrolled Class'. Then we splitted the data and examined the essential performance metrics for every classifier including Accuracy, Precision, F1 score, Recall and ROC AUC Score. Then we discovered that the best performance is provided by Random Forest, AdaBoost and compared to remaining others Decision tree and Logistic Regression showed better performance. After that, we carried out additional feature scaling to enhance K-Nearest Neighbor, Support Vector Machine and Logistic Regression performance. Following all of the experiments, we discovered that the models with the best performance are Decision Tree, Random Forest, Logistic Regression and Support Vector Machine. To further enhance these classifiers' performance, we used Random Search technique to tune their hyperparameters.

- K-Nearest Neighbor: It is used to categorize data points. It functions by finding 'K' closest points(neighbors) to a new data point and allocates it to the most prevalent class among those neighbors. To calculate proximity, it uses distance measure such as Euclidean distance, Manhattan distance etc. We chose the value of 'K' after plotting the data. We found that when K=7 the train accuracy is 90.38% and test accuracy is 89%. When K=28, the train accuracy is 86.81% and test accuracy is 87.55%. Hence, we can conclude, upon increasing the value of K, accuracy of the model decreases.

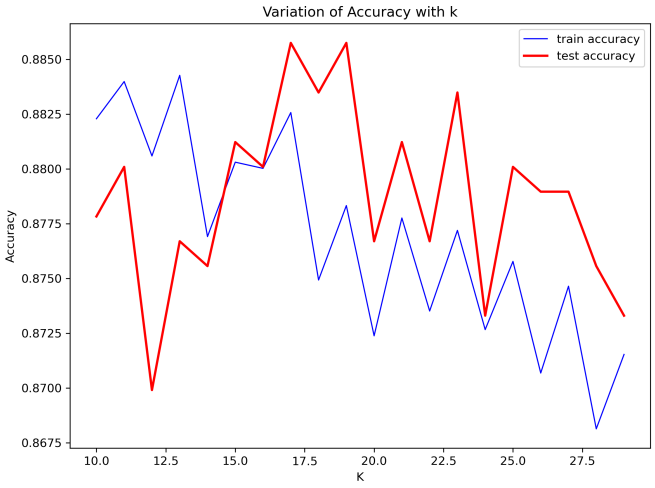


Fig. 5. Accuracy against K value

- Decision Tree: Decision tree is one of the most frequently used machine learning classifier. This approach splits the datasets into smaller subgroups by choosing the most appropriate attribute at the root. The main objective behind this division is to divide the dataset into discrete categories. Until the resultant subgroups are homogeneous, the splitting procedure is continued. A few well-known decision tree algorithms include ID3, Hunt's, C4.5, CART etc. In our approach, we used 'Gini' concept for constructing our decision tree. Gini is a measurement of impurity; zero gini means all instances belong to the same class. Gini index for a given node t, [9]

$$\text{Gini Index} = 1 - \sum_{i=0}^{c-1} p_i(t^2) \tag{1}$$

where $p_i(t)$ is the frequency of class i at node t and c is the total number of classes. A higher gini count indicates greater heterogeneity indicating a more mixed group of instances hence more impure form of distribution. In the beginning of our experiment, decision tree showed 100% train accuracy and 83% test accuracy. To improve the model performance further, we applied Random Search hyperparameter tuning that resulted in 90% test accuracy.

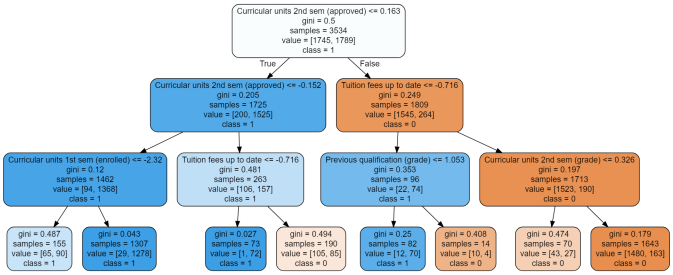


Fig. 6. Decision Tree

- Random Forest: Random Forest improves upon single decision trees using a technique called bootstrap aggregating. It creates multiple decision tree models by training on various versions of the training set. During classification, each model contributes to the final decision through a voting process. Popular for both classification and regression tasks, Random Forest combines multiple decision trees to make predictions. In our project, the Random Forest classifier initially achieved an accuracy of 91%. Following hyperparameter tuning, the accuracy increased to 92%.In both scenarios, the training accuracy was consistently 100%.
- Support Vector Machine(SVM): It is a classification algorithm that determines the best hyperplane to divide datapoints belonging to various classes in a high-dimensional space. The act of mapping complicated datasets to higher dimensions in a way that facilitates data point separation is the foundation of kernel functions. It simplifies the data boundaries for non-linear problems by adding higher dimensions to map complex data points. During our experiment, SVM model provided training accuracy of 100% and test accuracy of 47.5%. Upon feature scaling, model showed us a test accuracy of 91%. As it gave

such a tremendous performance boost, we further tuned its hyperparameters and got 94% training accuracy and 92% test accuracy.

- Logistic Regression: Logistic regression is a popular machine learning algorithm for binary classification problems. It estimates the likelihood that a given input belongs to a specific class. The basic principle underlying logistic regression is to employ a linear combination of input features. Rather than predicting the output directly, it used a sigmoid function to map the result to a probability value between 0 and 1. The sigmoid function is defined as:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-7x}} \tag{2}$$

where,

$$g(\theta^T x) = g(z) = \frac{1}{1 + e^{-z}} \tag{3}$$

h_θ is the hypothesis that determines the predicted output; y is predicted to be 1 if $h_\theta(x) \geq 0.5$ and y is predicted to be 0 if $h_\theta(x) < 0.5$. $g(z)$ maps an S-shaped curve as follows:

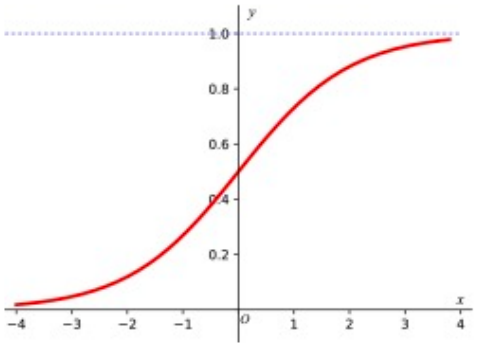


Fig. 7. Sigmoid Function Curve
[4]

Upon applying this algorithm, our model demonstrated 92% training accuracy and 91% test accuracy initially. Since it was one of the best performing models, we applied Random Search based hyperparameter tuning which resulted in same train and test accuracy as before.

- Gaussian Naive Bayes: Naive Bayes is a widely utilized machine learning technique for classification problems. It is based on Bayes' theorem, which assumes that features are conditionally independent given the class labels. Naive Bayes is particularly effective for handling large feature spaces and small training datasets. In this project, Naive Bayes was implemented using the Scikit-learn library. The model achieved a test accuracy of 87% and a training accuracy of 85%.
- AdaBoost: Adaptive Boosting (AdaBoost) is an ensemble learning method utilized for both classification and regression tasks. It sequentially combines multiple weak learners, typically decision trees with a single split, where each subsequent model focuses on correcting the errors made by its predecessor by assigning higher weights to misclassified instances. Through a weighted voting scheme, AdaBoost aggregates the predictions of these weak learners to construct a strong learner that surpasses the performance of its individual components. In our project, AdaBoost achieved a test accuracy of 90% and a training accuracy of 92%.
- XGBoost: XGBoost, or Extreme Gradient Boosting, is a highly efficient machine learning algorithm that constructs an ensemble of decision trees to enhance prediction accuracy. It iteratively corrects errors from previous trees by minimizing a loss function and employs regularization techniques to prevent overfitting. Known for its speed and performance, XGBoost utilizes parallel processing and tree pruning, making it well-suited for large datasets and complex tasks. Widely used in data science competitions and real-world applications, it excels in both classification and regression problems. In our project, XGBoost achieved a test accuracy of 56% and a training accuracy of 100%.

G. Hyperparameter Tuning

After predicting academic performance, we chose the classifier with better accuracy and fine-tuned its hyperparameters using RandomizedSearchCV. Our study showed that the Decision Tree, Random Forest, Logistic Regression, and SVM models performed well, so we tuned their hyperparameters. After tuning, the accuracy of the Decision Tree model increased from 83% to 91%. The accuracies of the Random Forest and SVM models improved from 91% to 92%. The accuracy of the Logistic Regression model stayed the same at 91%.

IV. Results

Performance evaluation was conducted based on accuracy, precision, recall, and F1 score metrics. These metrics are derived from the counts of true positives, false positives, true negatives, and false negatives.

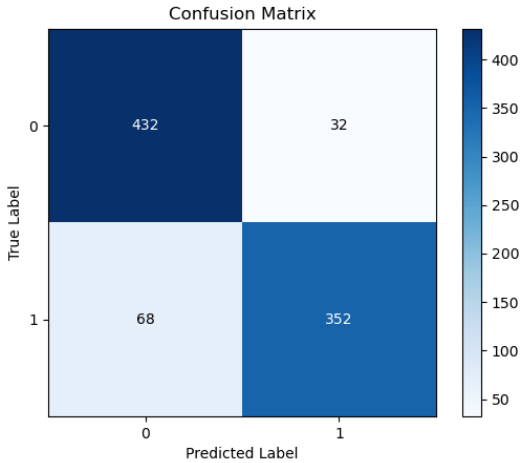


Fig. 8. KNN Confusion Matrix

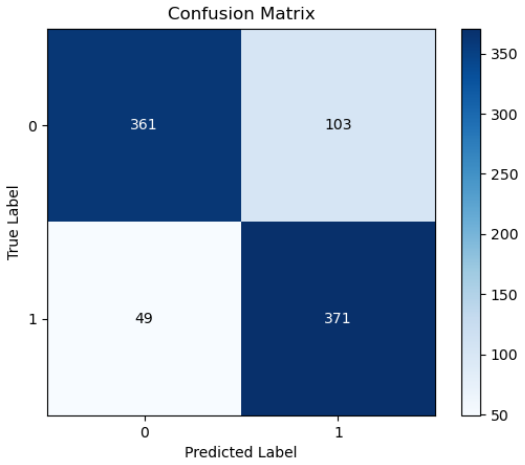


Fig. 9. Decision Tree Confusion Matrix

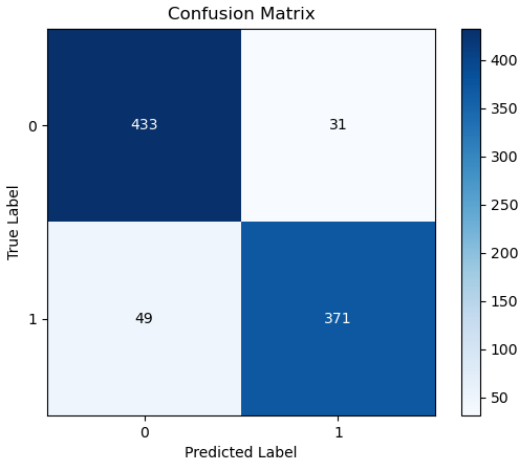


Fig. 10. Random Forest Confusion Matrix

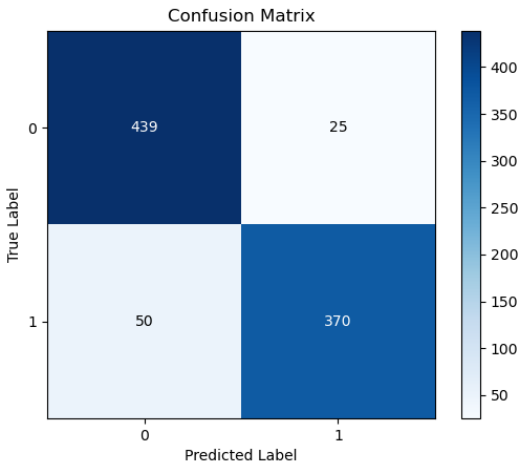


Fig. 11. SVM Confusion Matrix

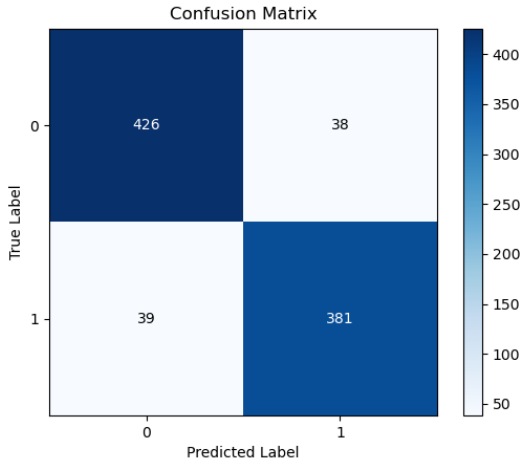


Fig. 12. Logistic Regression Confusion Matrix

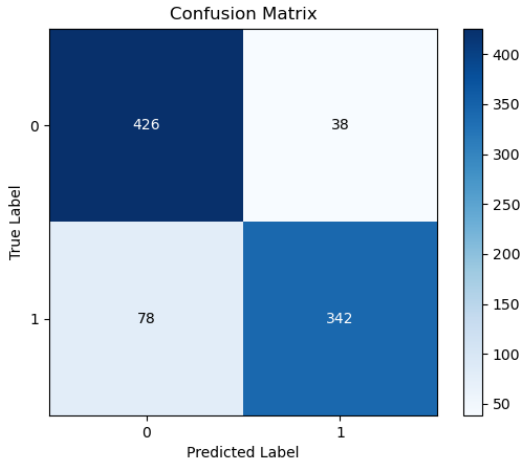


Fig. 13. Gaussian Naive Bayes Confusion Matrix

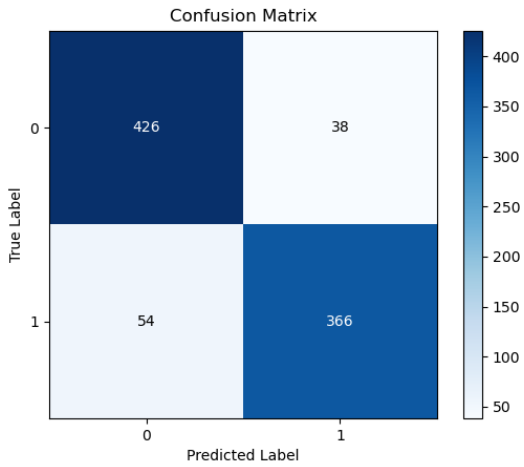


Fig. 14. AdaBoost Confusion Matrix

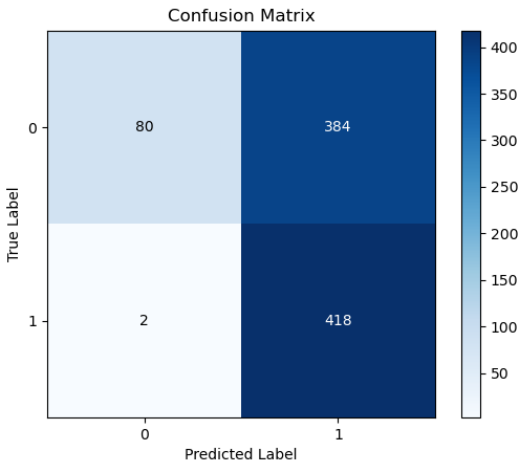


Fig. 15. XgBoost Confusion Matrix

Figures 8-15 depict the confusion matrices for the Student Performance dataset. The confusion matrix serves to define the performance of a classification algorithm and provides a visual summary of its performance. Accuracy, precision, recall, and F1-score scores were computed from these confusion matrices using the following equations:
[10]

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

[10]

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5}$$

[10]

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6}$$

[10]

$$F\text{-}Measure = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

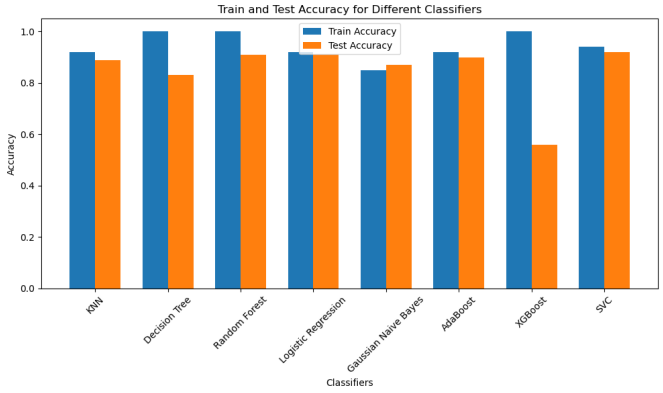


Fig. 16. Classifier Accuracy Evaluation

Figure 16 illustrates the comparison between the training and testing accuracy of the eight classification algorithms.

Classifiers	Accuracy	Precision	Recall	F1-Score
KNN	89%	91.67%	83.81%	88.46%
Decision Tree	91%	91%	90%	90%
Logistic Regression	91%	90.93%	90.71%	90.82%
SVM	92%	93.67%	88.09%	90.8%
Random Forest	92%	93.93%	88.57%	91.17%
Gaussian Naive Bayes	87%	90.64%	80.71%	85.39%
XGBoost	56%	75%	58%	49%
AdaBoost	90%	90.49%	88.33%	89.39%

TABLE III
CLASSIFICATION PERFORMANCE METRICS

Accuracy, Precision (Positive Predictive Value), Recall (Sensitivity), and F-Measure were also evaluated to compare the performance of the eight algorithms. The results, presented in Table III, indicate that the Random Forest and Support Vector Machine (SVM) algorithms outperformed all other algorithms in detecting and classifying academic success.

V. DISCUSSION

The objective of this project was to predict the academic success of university students, with a particular focus on identifying the dropout rates. To achieve this, eight distinct machine learning algorithms were employed: K-Nearest Neighbors (KNN), Decision Tree , Random Forest, Logistic Regression, Support Vector Machine (SVM), Gaussian Naive Bayes, AdaBoost, and XGBoost. Each of these algorithms was utilized to forecast students' academic outcomes based on various features. The goal was to determine which algorithm would provide the most accurate predictions. Among the algorithms tested, the Random Forest and Support Vector Machine classifiers yielded the best results. Both of these algorithms demonstrated superior performance, each achieving a prediction accuracy of 92%. This high level of accuracy indicates their effectiveness in identifying students who are at risk of dropping out, thereby potentially aiding in early intervention and support measures to improve student retention rates.

VI. LIMITATIONS

The limitations of our work are stated below:

- The dataset has an uneven distribution of classes (Dropout, Enrolled, Graduated), which can bias machine learning models towards the majority class.
- With 37 attributes and 4424 records, the dataset is at risk of overfitting, where models may learn noise instead of patterns.
- The data is from a single institution, limiting the generalizability of findings to other institutions.
- Missing influential factors like psychological metrics, peer influence, and extracurricular activities.

VII. CONCLUSION

This project highlights the analytical potential of predicting student dropout and academic success by assessing various

machine learning algorithms and identifying the best performers. Among the eight evaluated models, Random Forest and Support Vector Machine (SVM) achieved the highest accuracy at 92%. Other models, such as Decision Tree, Logistic Regression, and AdaBoost, also performed well. However, XGBoost was found to be inefficient, with the lowest accuracy at 56%.

REFERENCES

[1] Labelencoder. *scikit-learn*.

[2] Predicting student dropouts via machine learning. Jun 2021.

[3] and. Recent advances in predictive learning analytics: A decade systematic review (2012–2022). page 1–35, Jun 2022.

[4] Yunji Chen, Ling Li, Wei Li, Qi Guo, Zidong Du, and Zichen Xu. Chapter 2 - fundamentals of neural networks. In Yunji Chen, Ling Li, Wei Li, Qi Guo, Zidong Du, and Zichen Xu, editors, *AI Computing Systems*, pages 17–51. Morgan Kaufmann, 2024.

[5] Anat Cohen. Analysis of student activity in web-supported courses as a tool for predicting dropout. *Educational Technology Research and Development*, 65(5):1285–1304, October 2017. Publisher Copyright: © 2017, Association for Educational Communications and Technology.

[6] Cameron Gray and Dave Perkins. Utilizing early engagement and machine learning to predict student outcomes. *Computers Education*, 131, 12 2018.

[7] Vinayak Hegde and P. P. Prageeth. Higher education student dropout prediction and analysis through educational data mining. *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pages 694–699, 2018.

[8] Md Riyad Hossain, Douglas Timmer, and Hiram Moya. Machine learning model optimization with hyper-parameter tuning approach. 08 2021.

[9] Ari Melo Mariano, Arthur Bandeira de Magalhães Lelis Ferreira, Maíra Rocha Santos, Mara Lucia Castilho, and Anna Carla Freire Luna Campêlo Bastos. Decision trees for predicting dropout in engineering course students in brazil. *Procedia Computer Science*, 214:1113–1120, 2022. 9th International Conference on Information Technology and Quantitative Management.

[10] Khalid Oqaidi, Sarah Aouhassi, and Khalifa Mansouri. Towards a students' dropout prediction model in higher education institutions using machine learning algorithms. *International Journal of Emerging Technologies in Learning (iJET)*, 17:103–117, 09 2022.

[11] Eric Osemwegie and Frank Amadin. Student dropout prediction using machine learning. *FUDMA JOURNAL OF SCIENCES*, 7:347–353, 12 2023.

[12] Vieira Martins Mónica Machado Jorge Realinho, Valentim and Luís Baptista. Predict Students' Dropout and Academic Success. UCI Machine Learning Repository, 2021. DOI: https://doi.org/10.24432/C5MC89.

[13] Lakshmi Nath Roy, Uttam Majumder, and Dnr Paul. Dropout causes and the assessment of its consequences of primary students in the northern part of bangladesh. iosr journal of research method in education (iosrjrme), e-issn: 2320–1959. p- issn: 2320–1940. 9:54–61, 06 2019.

[14] Meseret Yihun and Stanislava Šimonová. Global challenges of students dropout: A prediction model development using machine learning algorithms on higher education datasets. *SHS Web of Conferences*, 129:09001, 01 2021.