African Scientific Research and Innovation Council

Available online at https://asric.africa/engineering-sciences
ASRIC Journal on Engineering Sciences Vol.4(1)(2023) 266-276

Student Performance Prediction Using Machine Learning: A Comprehensive Analysis

Ashraf Farouk Heikal & Zeyad Aly Khalil

Lecturers at Al-Obour Higher Institute for Management and Informatics for Management and Informatics ashraf.heikal@oi.edu.eg zeiadaly@oi.edu.eg

Abstract

The anticipation of student performance stands as a pivotal element in educational systems, with organizations aspiring to enrich the learning experience and elevate student outcomes. Its prominence in the education domain arises from its capacity to refine educational results and furnish invaluable insights for educators, administrators, and policymakers alike. In this paper, we use the Student Performance Dataset (SPD) to evaluate the effectiveness of different machine learning methods across diverse application scenarios. More precisely, we explore the performance of eighteen machine learning models that include classification models, namely Artificial Neural Networks (ANNs), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Naive Bayes (NB), Decision Trees (DT), Random Forests (RF), Boosting and Bagging models. These methods are systematically applied to analyze binary prediction tasks within the context of student performance. Various machine learning algorithms, are employed to analyze and predict student performance metrics, such as grades, exam scores, and graduation outcomes. Evaluation of predictive models is a critical aspect, and the paper examines various performance metrics such as Accuracy, Precision, Recall, F1-measure, and the Area Under the Receiver Operating Characteristic curve (AUC). The experimental results demonstrate that the Categorical Boosting model (CatBoost) outperformed the rest of the models used in the study as the best-performing model in general, as it consistently achieved high scores in accuracy, recall, F1 score, and AUC. The results also showed that the results of the Decision Tree (DT) model were lower than ensemble methods, indicating potential limitations in handling complex relationships. In addition, the performance of Bagging techniques generally improved performance compared to their base models, demonstrating the effectiveness of aggregating multiple models and Boosting Techniques models consistently performed well, indicating the power of sequential learning and model combination.

Keywords: Machine learning Models, Supervised Learning, Classification Algorithms, Student performance prediction

Introduction

Education is a fundamental pillar of modern society, serving as the foundation for personal and societal growth, development, and progress. As educational institutions continue to grow in size and complexity, there is an increasing need to monitor and enhance student performance and success. The performance of students within educational institutions is a critical indicator of the effectiveness of these institutions, as each student possesses unique strengths, weaknesses, and individualized learning methods. Academic performance can be influenced by a wide range of internal and external factors [1]. Predicting student performance is a crucial step in the pursuit of improving educational systems, ensuring the best possible outcomes for all students.

The motivation behind this study lies in showcasing the potential of machine learning in analyzing and predicting student performance. With the proliferation of data within educational institutions, including information related to the demographic composition of students, attendance, grades, and more, machine learning techniques can be harnessed to create predictive models that offer valuable insights. These models can assist educators, administrators, and policymakers in making informed decisions, ultimately leading to more effective educational interventions and resource allocation [2].

In recent times, numerous studies have demonstrated the effectiveness of machine learning techniques in predicting student performance. Machine learning technology offers a wealth of methods and tools that can be leveraged for this purpose, ensuring more accurate and reliable such as a k-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), Random Forest (RF), Logistic Regression (LR), and Artificial Neural Networks (ANNs) [3].

The primary objective of this paper is to provide a comprehensive analysis of the use of machine learning in predicting student performance. We aim to explore the various factors that impact student performance, including both academic and non-academic variables, and demonstrate how machine learning models can be employed to predict outcomes with a high degree of accuracy. By achieving this objective, we intend to:

(1) Identify the key features and factors influencing student performance.
(2) Evaluate the performance of various machine learning algorithms in predicting student outcomes.
(3) Provide insights into the practical application of machine learning models in educational institutions.
(4) Offer recommendations for improving student support and resource allocation based on predictive models.

This research focuses on the analysis of student performance using ensemble techniques, including (Boosting, and Bagging). Ensemble techniques are machine learning techniques that combine multiple models to improve predictive performance and reduce overfitting. These techniques are especially useful when working with complex data sets and models. The study examines a variety of features and their impact on student outcomes, including demographic information, attendance, past academic performance, and extracurricular activities. The study is applied to a dataset from a real educational institution to ensure the practical application of the results.

The remainder of this paper is organized as follows: Related work is presented in Section 2. The method used in this research is described in Section 3. The results of the experiments and the discussion are presented in Section 4. Finally, conclusions are given in Section 5.

Literature Review

A literature review for student performance prediction using machine learning techniques typically involves an analysis of existing research, methods, and findings in the field. There are many papers published that have dealt with the topic of predicting student performance using machine learning techniques and most of these works usually treat the problem as a classification or regression task, and use machine learning methods for training and prediction. We will list the most important published research on predicting student performance in next paragraph.

Harikumar Pallathadka et al. [4] utilized Nave Bayes, ID3, C4.5, and SVM techniques to predict student performance using UCI machinery student performance data set [5]. Algorithms are evaluated based on characteristics such as accuracy and error rate. SVM is the most accurate technique for classifying a data set of student performance.

Ihsan A. Abu Amra et al. [6] utilized two classification algorithms KNN and Naïve Bayes on educational data set of secondary schools, collected from the ministry of education in Gaza Strip for 2015 year. The experimental results in this paper show that Naïve Bayes is better than KNN by receiving the highest accuracy value of 93.6%.

Emmy Hossain et al. [7] proposed a system named Student Performance Analysis System (SPAS) to keep track of students' result in the Faculty of Computer Science and Information Technology (FCSIT). The proposed system utilized J-48, Simple CART, BFTree, Random Tree, J48 Garft techniques to predict the students' performance in course "TMC1013 System Analysis and Design", The experimental results in this paper show that BF-Tree is the best.

Leila Ismail et al. [8] evaluate and compare the performance of the most used machine learning classification models DT, NB, ANN, SVM, and RF, for students' performance prediction. The experimental results in this paper reveal that for a dataset having fewer observations, SVM linear, SVM polynomial, and

NB outperforms the other models under study, whereas for a dataset having a large number of observations, DT and RF outperforms the other models under study.

Ajibola Oyedeji et al. [9] analyzed the past results of students including their individual attributes including age, demographic distribution, family background and attitude to study and tests this data using machine learning tools. Three models which are; Linear regression for supervised learning, linear regression with deep learning and neural network were tested using the test and train data with the Linear regression for supervised learning having the best mean average error (MAE).

Annisa Uswatun Khasanah et al. [10] conducted Feature Selection to select high influence attributes with student performance in Department of Industrial Engineering Universitas Islam Indonesia. Then, two popular classification algorithm, Bayesian Network and Decision Tree, were implemented and compared to know the best prediction result. The outcome showed that student's attendance and GPA in the first semester were in the top rank from all Feature Selection methods, and Bayesian Network is outperforming Decision Tree since it has higher accuracy rate.

Nitin Ramrao Yadav et al. [11] reviewed many papers aimed at predicting student performance in education sector. The most widely used Machine Learning algorithms to enhance student performance at entry level and during academic year are Artificial Neural Network (ANN), Support Vector Machine (SVM), Naïve Bayes (NB), Linear Regression (LR) and Decision Tree (DT). Praveena Chakrapani et al. [12] presented a comprehensive and systematic literature review of the numerous researches done in predicting students' performance through Machine Learning techniques during the period 2015 to 2022 and assess the quality of the accuracy of predictions in a clear and crisp manner. In this review, papers published

Yawen Chen et al. [13] presented a summary of a series of studies that used machine learning techniques to predict student performance including the machine learning techniques, dataset used, limitations, and specific educational tasks addressed in the studies was provided. Then applied seven machine learning methods (Random Forest, Decision Tree, K-Nearest Neighbors (KNN), Naive Bayes, Support Vector Machine (SVM), Logistic Regression (LR), and Artificial Neural Networks (ANNs)) to study multiple types of performance prediction (binary and multiclassification prediction) on three different types of task-oriented educational data to investigate the performance of machine learning methods in different application scenarios. The experimental results concluded that Random Forest achieved superiority in all selected data sets.

Method

The aim of the envisioned system technology is to enhance the accuracy of student performance predictions by employing ensemble techniques. Figure (1) illustrates the schematic layout of the proposed system, which comprises:

(1) Data collection and preprocessing
(2) Feature selection,
(3) Data partitioning
(4) Model selection
(5) Model training and testing
(6) Model evaluation.
(7) Resulting

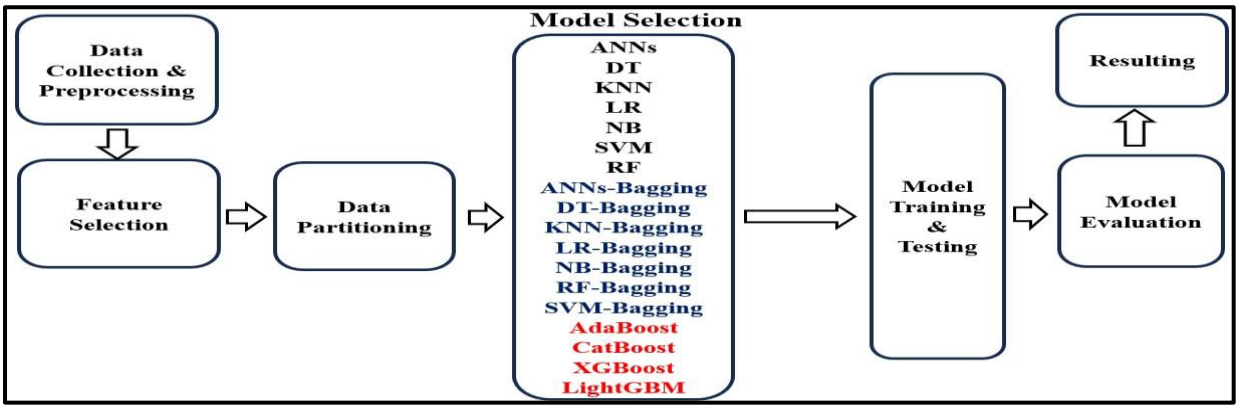

Figure 1: The schematic layout of the proposed system

Data Collection and Preprocessing

In this study We leveraged student performance data to categorize and anticipate student grades. To be more precise, we employed the Student Performance Dataset (SPD), which can be accessed from the widely recognized UCI Repository. The SPD dataset is available in the website (https://www.kaggle.com/datasets/larsen0966/student-performance-data-set).

This dataset pertains to the academic performance of secondary school students in two Portuguese schools, as described in reference [14]. The dataset encompasses a range of data attributes, including student grades, demographic information, social factors, and school-related characteristics. These details were gathered through the use of school reports and questionnaires. Our specific area of interest was the subject of Mathematics (mat), and the variable we aimed to predict was the third-quarter grade. In total, the dataset consists of 649 samples, each representing a student, and encompasses 33 variables, which are detailed in table 1.

We categorized students' scores into two distinct groups. The first category, marked “Fail,” includes grades less than 10. The second category, marked “Pass,” includes grades greater than or equal to 10. During the data pre-processing phase, the initial step involves checking for missing values. When dealing with numerical missing values, mean imputation is utilized as it is a straightforward and effective method, especially since these missing values typically constitute only a small portion of the data. Categorical missing values, on the other hand, are filled using the mode value to prevent unnecessary loss of information.

Table (1): Model variables of SPD.

No.	Attribute	Type	Description
F1	school	binary	student's school ('GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
F2	sex	binary	student's sex ('F' - female or 'M' - male)
F3	age	numeric	student's age (from 15 to 22)
F4	address	binary	student's home address type (binary: 'U' - urban or 'R' - rural)
F5	famsize	binary	family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
F6	Pstatus	binary	parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
F7	Medu	arithmetic	mother’s education (arithmetic: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 – secondary education or 4 - higher education)
F8	Fedu	arithmetic	father's education (arithmetic: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 – secondary education or 4 - higher education)
F9	Mjob	nominal	mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
F10	Fjob	nominal	father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
F11	reason	nominal	reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
F12	guardian	nominal	student's guardian (nominal: 'mother', 'father' or 'other')

F13	traveltime	numeric	home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
F14	studytime	numeric	weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
F15	failures	numeric	number of past class failures (numeric: n if 1<=n<3, else 4)
F16	schoolsup	binary	extra educational support (binary: yes or no)
F17	famsup	binary	family educational support (binary: yes or no)
F18	paid	binary	extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
F19	activities	binary	extra-curricular activities (binary: yes or no)
F20	nursery	binary	attended nursery school (binary: yes or no)
F21	higher	binary	wants to take higher education (binary: yes or no)
F22	internet	binary	Internet access at home (binary: yes or no)
F23	romantic	binary	with a romantic relationship (binary: yes or no)
F24	famrel	numeric	quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
F25	freetime	numeric	free time after school (numeric: from 1 - very low to 5 - very high)
F26	goout	numeric	going out with friends (numeric: from 1 - very low to 5 - very high)
F27	Dalc	numeric	workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
F28	Walc	numeric	weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
F29	health	numeric	current health status (numeric: from 1 - very bad to 5 - very good)
F30	absences	numeric	number of school absences (numeric: from 0 to 93)
F31	G1	numeric	first period grade (numeric: from 0 to 20)
F32	G2	numeric	second period grade (numeric: from 0 to 20)
F33	G3	numeric	final grade (numeric: from 0 to 20, **output target**)

In the second step, categorical and continuous data are processed differently. For categorical data, we first employ Ordinal Encoding to transform it into an array of integers [15]. However, these integers cannot be directly input into the model because they might be misinterpreted as ordered values by the machine learning model. In reality, they only represent different categories without any inherent hierarchy. Hence, we employ One-Hot Encoding to properly handle this data [16].

On the other hand, when dealing with continuous data, we opt for the split-box method, which discretizes the continuous data to reduce noise and mitigate the risk of model overfitting. For categorical labels, such as those found in binary classification problems where labels are typically "Yes" or "No," we use Label Encoding to process these labels accordingly [17].

Feature Selection

Feature selection is a process to select a subset of the original features for model training it's usually used as a pre-processing step before doing the actual learning. Feature selection is an important step in machine learning and data analysis, where you choose a subset of the most relevant features from your dataset to improve model performance, reduce overfitting, and speed up the training process. Various techniques and models can be used for feature selection. There are several models used for feature selection, each with its own advantages and disadvantages [18]. Some of the most commonly used models are:

(1) Filter methods: These methods use statistical tests to evaluate the correlation between each feature and the target variable. Such as chi-squared, mutual information, and correlation coefficient.

(2) Wrapper methods: These methods use a specific machine learning algorithm to evaluate the performance of different subsets of features. Such as Recursive Feature Elimination (RFE), Forward Floating Selection (FFS), and Backward Floating Selection (BFS).

(3) Embedded methods: These methods are also called Intrinsic methods. These methods combine feature selection with model training. Such as Lasso regression, Random Forest, Gradient Boosting Models, and decision trees.

In our research, we employed a filtering technique (correlation coefficient), to quantify the relationship between features and the target variable. Figure 2 illustrates the degree of correlation between the features and the target, whether it is positive or negative. Table 2 delineates the features and its ranking. To enhance model accuracy and prevent overfitting, we reduced the number of features by deleting features with a positive correlation score less than 0.1 and features with a negative correlation score more than -0.1.

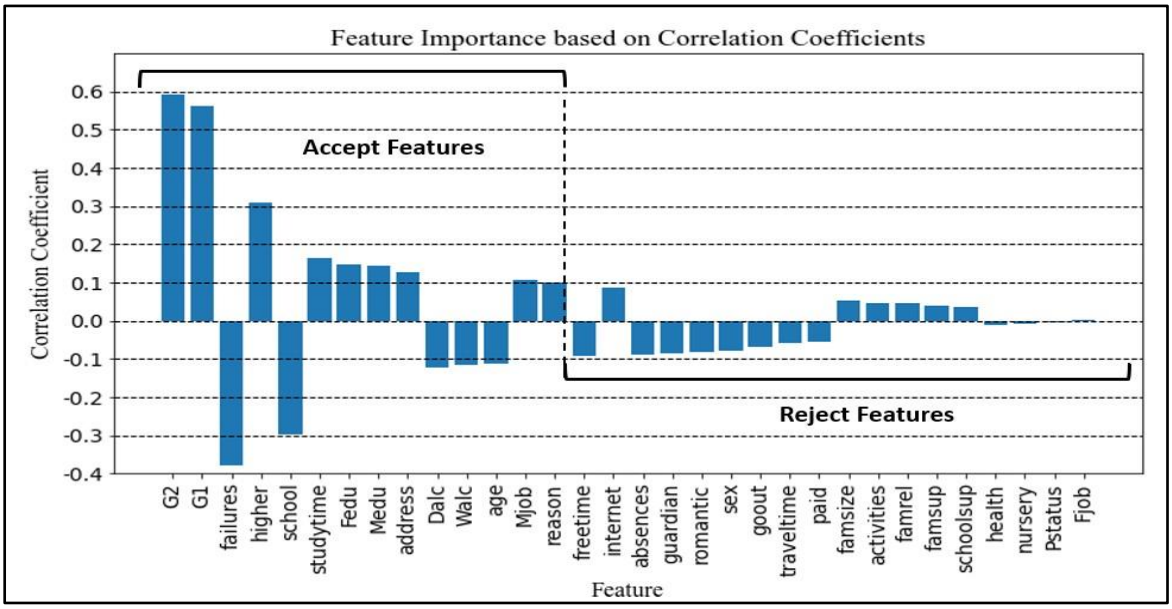

Figure (2): Feature importance based on correlation coefficients

Data Partitioning

In this step, the heart disease dataset is divided into an 80% training set and a 20% as the testing set. the training set is utilized for training the models, and the testing set is utilized to evaluate the models. Also, ninefold cross-validation is utilized in the training set.

Model Selection

In this section, we have selected eighteen machine learning models, encompassing KNN, DT, RF, SVM, NB, ANN, and ensemble techniques, including Boosting and Bagging. The Bagging models consist of ANNs-Bagging, DT-Bagging, KNN-Bagging, LR-Bagging, NB-Bagging, RFBagging, and SVM-Bagging. The Boosting models encompass Adaptive Boosting (AdaBoost), Categorical Boosting (CatBoost), Extreme Gradient Boosting (XGBoost), and Light Gradient Boosting (LightGBM). The essential principles and critical parameters for each algorithm are elaborated upon in the following descriptions.

(1) K-Nearest Neighbors (KNN):

K-Nearest Neighbors (KNN) is a straightforward and versatile machine learning algorithm used for classification and regression tasks. In the training phase, KNN stores labeled instances, each characterized by features and corresponding outcomes. When predicting the label or value for a new data point, the algorithm identifies the k-nearest neighbors based on distance metrics like Euclidean or Manhattan distance. For classification, the majority class among these neighbors is assigned to the new point, while regression predictions are derived from aggregating the target values. KNN is known for its simplicity, making it a useful baseline model in various applications [19].

(2) Decision Tree (DT):

The Decision Tree algorithm is a popular and widely used supervised machine learning algorithm. It is primarily used for classification and regression tasks. Decision Trees are a flowchart-like structure where each internal node represents a feature or attribute, each branch represents a decision rule, and each leaf node represents the outcome or the class label. The main objective of the Decision Tree algorithm is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the input features. It partitions the data into subsets based on different attributes and recursively builds a tree-like structure until it reaches the leaf nodes [20].

(3) Random Forests (RF):

Random Forest is a popular and powerful algorithm used in machine learning for both classification and regression tasks. It is an ensemble learning method that combines multiple decision trees to make predictions. Random Forest gets its name from the fact that it creates a "forest" of decision trees, where each tree is built using a random subset of the training data and features. The main idea behind Random Forest is to reduce overfitting by introducing randomness in the model. It achieves this by combining multiple decision trees, each trained on a different subset of the data and features. The final prediction is made by aggregating the predictions of all the individual trees, either through voting (for classification) or averaging (for regression) [21].

(4) Naive Bayes (NB):

Naive Bayes is a simple yet powerful algorithm commonly used in machine learning for classification tasks. It is based on the Bayes' theorem and assumes that the features are conditionally independent of each other, given the class label. The algorithm is called "naive" because it makes a strong assumption of feature independence, which may not always hold true in real-world scenarios. Despite this assumption, Naive Bayes has been proven to perform well in various applications, especially in text classification and spam filtering. Naive Bayes algorithm calculates the probability of a sample belonging to a particular class by using the joint probability of the features given the class label. It then assigns the class label with the highest probability to the sample [22].

(5) Support Vector Machine (SVM):

Support Vector Machine (SVM) is a powerful and versatile supervised learning algorithm used for both classification and regression tasks. It is used for both binary and multi-class classification tasks. SVM aims to find a hyperplane in a high-dimensional space that best separates data points into different classes while maximizing the margin between the classes. The "support vectors" are the data points closest to the decision boundary, and the algorithm's effectiveness is often attributed to its ability to handle complex, nonlinear relationships through the use of kernel functions. SVMs have found success in various applications, including image classification, text classification, and bioinformatics [23].

(6) Logistic Regression (LR):

Logistic regression is a popular statistical model used for binary classification tasks in machine learning. It is a supervised learning algorithm that predicts the probability of an instance belonging to a particular class. In logistic regression, the dependent variable is binary, meaning it can take only two possible outcomes. The independent variables, also known as features, can be continuous or categorical. The goal of logistic regression is to find the best-fitting line (or hyperplane) that separates the two classes. The logistic regression model uses the logistic function (also called the sigmoid function) to map the predicted values to probabilities between 0 and 1. Logistic regression has several advantages, including simplicity, interpretability, and efficiency. It can handle both linear and non-linear relationships between the independent variables and the log-odds of the dependent variable [24].

(7) Artificial Neural Networks (ANNs):

Artificial Neural Networks (ANN) is a popular algorithm used in machine learning. ANNs can be used for various tasks, such as classification, regression, and pattern recognition. It is inspired by the structure and functioning of the human brain. ANN consists of interconnected nodes, called artificial neurons, which are organized in layers. The basic unit of an ANN is the artificial neuron, also known as a perceptron. Each perceptron takes multiple inputs, applies weights to these inputs, and then passes the weighted sum through an activation function to produce an output. The activation function introduces non-linearity into the network, enabling it to learn complex patterns and relationships. ANNs are trained using a process called backpropagation. During training, the network adjusts the weights and biases of its neurons based on the error between its predicted output and the expected output. This iterative process continues until the network achieves a satisfactory level of accuracy [25].

(8) Ensemble Techniques:

Ensemble techniques involve combining multiple machine learning models to enhance prediction accuracy and robustness. Rather than relying on a single model, these techniques harness the

collective insights of various models, contributing to improved performance. They are instrumental in mitigating overfitting, enhancing generalization, and refining prediction accuracy, finding applications in diverse domains like finance, healthcare, and natural language processing. There are two primary categories of ensemble techniques (Bagging and Boosting). In bagging, numerous models are independently trained on distinct subsets of the training data. The predictions from each model are then amalgamated through averaging or voting to yield the final prediction [26]. In boosting, multiple models are trained sequentially, with each subsequent model aiming to rectify errors made by its predecessors. The ultimate prediction is derived by aggregating the predictions of all models [27].

Model Evaluation

Experimental Results

The experimental results were implemented using Python and executed on an Intel (R) Core i7 CPU with 16 GB of memory. To assess and compare the performance of the models, five evaluation metrics were chosen: Accuracy, Precision, Recall, F1-measure, and the Area Under the ROC Curve (AUC). The performance results of the eighteen models on SPD are shown in Tables 2.

Table (2): The comparative performance of eighteen models on SPD.

No	Model	Accuracy	Precision	Recall	F1 Score	AUC
1	**ANNs**	90.8%	94.8%	94.8%	94.8%	93.1%
2	**DT**	83.1%	86.7%	93.3%	89.9%	78.9%
3	**KNN**	89.2%	90.4%	98.3%	94.2%	92.7%
4	**LR**	90.0%	95.5%	93.0%	94.3%	96.3%
5	**NB**	90.0%	**97.2%**	91.3%	94.2%	94.1%
6	**SVM**	88.5%	92.4%	94.8%	93.6%	95.4%
7	**ANNs-Bagging**	91.5%	95.6%	94.8%	95.2%	94.0%
8	**DT-Bagging**	93.1%	95.7%	96.5%	96.1%	95.9%
9	**KNN-Bagging**	89.2%	90.4%	98.3%	94.2%	92.8%
10	**LR-Bagging**	90.8%	95.6%	93.9%	94.7%	96.0%
11	**NB-Bagging**	90.0%	**97.2%**	91.3%	94.2%	94.1%
12	**RF-Bagging**	93.1%	94.9%	97.4%	96.1%	96.5%
13	**SVM-Bagging**	90.8%	94.8%	94.8%	94.8%	95.0%
14	**RF**	93.1%	94.9%	97.4%	96.1%	96.6%
15	**AdaBoost**	91.5%	96.4%	93.9%	95.2%	96.3%
16	**CatBoost**	**95.4%**	95.8%	**99.1%**	**97.4%**	**97.0%**
17	**XGBoost**	93.8%	94.2%	**99.1%**	96.6%	95.7%
18	**LightGBM**	91.5%	96.4%	93.9%	95.2%	95.9%

4.2. Comprehensive Analysis:

(1) Accuracy:

- Highest Accuracy: Categorical Boosting (CatBoost) with an accuracy of 95.4%.
- Other High Performers: Extreme Gradient Boosting (XGBoost) and Random Forest (RF) also demonstrate high accuracy (93.8% and 93.1%, respectively).
- Lowest Accuracy: Decision Tree (DT) with an accuracy of 83.1%.

(2) Precision:

- Highest Precision: Naive Bayes (NB) Bagging and Logistic Regression (LR) Bagging both achieve a precision of 97.2%.
- Other High Performers: CatBoost and XGBoost show high precision (95.8% and 94.2%, respectively).

- Lowest Precision: Decision Tree (DT) with a precision of 86.7%.

(3) Recall:
- Highest Recall: CatBoost with a recall of 99.1%.
- Other High Performers: XGBoost, K-Nearest Neighbors (KNN) Bagging, and Random Forest (RF) Bagging demonstrate high recall (99.1%, 98.3%, and 97.4%, respectively).
- Lowest Recall: Naive Bayes (NB) with a recall of 91.3%.

(4) F1 Score:
- Highest F1 Score: Categorical Boosting (CatBoost) with an F1 score of 97.4%.
- Other High Performers: Extreme Gradient Boosting (XGBoost) and Random Forest (RF) also show high F1 scores (96.6% and 96.1%, respectively).
- Lowest F1 Score: Decision Tree (DT) with an F1 score of 89.9%.

(5) Area Under the Curve (AUC):
- Highest AUC: Categorical Boosting (CatBoost) with an AUC of 97%.
- Other High Performers: Logistic Regression (LR), XGBoost, and Random Forest (RF) also exhibit high AUC values (96.3%, 95.7%, and 96.6%, respectively).
- Lowest AUC: Decision Tree (DT) with an AUC of 78.9%.

(6) While Decision Trees can be interpretable, the provided Decision Tree model has relatively lower accuracy (83.1%) and AUC (78.9%) compared to other models. This may suggest that the decision tree is not capturing the underlying patterns in the data as effectively.

(7) Bagging Techniques (RF-Bagging, DT-Bagging, ANN-Bagging, etc.): Generally improved performance compared to their base models, demonstrating the effectiveness of aggregating multiple models.

(8) Boosting Techniques (AdaBoost, CatBoost, XGBoost): Consistently performed well, indicating the power of sequential learning and model combination.

Conclusion

In this research, our focus centered on the prediction of student performance. To refine our predictive capabilities, we employed the correlation coefficient technique, reducing the initial set of 32 features to a more manageable 14. Subsequently, we applied eighteen machine learning models, encompassing ensemble methods such as boosting and bagging, to classify students into distinct groups based on their performance: Pass or Fail.

A comprehensive comparison was conducted between ensemble methods (boosting and bagging) and six classifiers (ANNs, KNN, SVM, NB, DT, and RF) based on the selected features. The results showed that the categorical boosting model (CatBoost) outperformed the rest of the models used in the study as the best-performing model in general, as it consistently achieved high scores in accuracy, precision, recall, F1 score, and AUC. The results also showed that the results of the Decision Tree model were lower than ensemble methods, indicating potential limitations in handling complex relationships.

In the future, we would like to apply more advanced techniques like deep learning in the student performance prediction tasks to further improve the prediction of the model. Furthermore, Reinforcement learning could be explored to design adaptive learning environments. These environments would dynamically adjust content and difficulty based on students' progress, ensuring an optimal learning experience. Future trends may involve the development of highly personalized learning models. These models could adapt to individual student needs, taking into account learning styles, preferences, and pace.

References

1. Chamidy, Totok, Muhammad Ainul Yaqin, and Suhartono Suhartono. "The Influence of Internal and External Factors on Learning Achievement." In 4th Annual International Conference on Language, Literature and Media (AICOLLIM 2022), pp. 562-573. Atlantis Press, 2023.
2. Oyedeji, Ajibola Oluwafemi, Abdulrazaq M. Salami, Olaolu Folorunsho, and Olatilewa R. Abolade. "Analysis and prediction of student academic performance using machine learning." JITCE (Journal of Information Technology and Computer Engineering) 4, no. 01 (2020): 10-15.
3. Rastrollo-Guerrero, Juan L., Juan A. Gómez-Pulido, and Arturo Durán-Domínguez. "Analyzing and predicting students' performance by means of machine learning: A review." Applied sciences 10, no. 3 (2020): 1042.
4. Pallathadka, Harikumar, Alex Wenda, Edwin Ramirez-Asís, Maximiliano Asís-López, Judith Flores-Albornoz, and Khongdet Phasinam. "Classification and prediction of student performance data using various machine learning algorithms." Materials today: proceedings 80 (2023): 3782-3785.
5. https://archive.ics.uci.edu/ml/datasets/Student+Performance
6. Amra, Ihsan A. Abu, and Ashraf YA Maghari. "Students performance prediction using KNN and Naïve Bayesian." In 2017 8th international conference on information technology (ICIT), pp. 909-913. IEEE, 2017.
7. Sa, Chew Li, Emmy Dahliana Hossain, and Mohammad bin Hossin. "Student performance analysis system (SPAS)." In The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M), pp. 1-6. IEEE, 2014.
8. Ismail, Leila, Huned Materwala, and Alain Hennebelle. "Comparative Analysis of Machine Learning Models for Students' Performance Prediction." In Advances in Digital Science: ICADS 2021, pp. 149-160. Springer International Publishing, 2021.
9. Oyedeji, Ajibola Oluwafemi, Abdulrazaq M. Salami, Olaolu Folorunsho, and Olatilewa R. Abolade. "Analysis and prediction of student academic performance using machine learning." JITCE (Journal of Information Technology and Computer Engineering) 4, no. 01 (2020): 10-15.
10. Khasanah, Annisa Uswatun. "A comparative study to predict student's performance using educational data mining techniques." In IOP Conference Series: Materials Science and Engineering, vol. 215, no. 1, p. 012036. IOP Publishing, 2017.
11. Yadav, Nitin Ramrao, and Sonal Sachin Deshmukh. "Prediction of Student Performance Using Machine Learning Techniques: A Review." In International Conference on Applications of Machine Intelligence and Data Analytics (ICAMIDA 2022), pp. 735-741. Atlantis Press, 2023.
12. Chakrapani, Praveena, and D. Chitradevi. "Academic performance prediction using machine learning: A comprehensive & systematic review." In 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC), pp. 335-340. IEEE, 2022.
13. Chen, Yawen, and Linbo Zhai. "A comparative study on student performance prediction using machine learning." Education and Information Technologies (2023): 1-19.
14. Cortez, Paulo, and Alice Maria Gonçalves Silva. "Using data mining to predict secondary school student performance." (2008).
15. Potdar, Kedar, Taher S. Pardawala, and Chinmay D. Pai. "A comparative study of categorical variable encoding techniques for neural network classifiers." International journal of computer applications 175, no. 4 (2017): 7-9.
16. Bagui, Sikha, Debarghya Nandi, Subhash Bagui, and Robert Jamie White. "Machine learning and deep learning for phishing email classification using one-hot encoding." Journal of Computer Science 17 (2021): 610-623.
17. Bisong, Ekaba. Introduction to Scikit-learn. In Building machine learning and deep learning models on Google cloud platform (pp. 215–229). Apress, Berkeley, CA.
18. Chandrashekar, Girish, and Ferat Sahin. "A survey on feature selection methods." Computers & Electrical Engineering 40, no. 1 (2014): 16-28.
19. Altman, Naomi S. "An introduction to kernel and nearest-neighbor nonparametric regression." The American Statistician 46, no. 3 (1992): 175-185.
20. Quinlan, J. Ross. "Induction of decision trees." Machine learning 1 (1986): 81-106.
21. Breiman, Leo. "Random forests." Machine learning 45 (2001): 5-32.
22. Manning, Christopher D., Prabhakar Raghavan, and Hinriche Schütze. "Xml retrieval." Introduction to Information Retrieval (2008).
23. Schölkopf, Bernhard, and Alexander J. Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.
24. Brownlee, Jason. "Logistic regression for machine learning." Machine Learning Mastery 1 (2016).

25. Da Silva, I. Nunes, D. Hernane Spatti, R. Andrade Flauzino, LH Bartocci Liboni, and S. F. dos Reis Alves. "Artificial Neural Networks, 2017. doi: 10.1007."
26. Hastie, Trevor, Robert Tibshirani, Jerome H. Friedman, and Jerome H. Friedman. The elements of statistical learning: data mining, inference, and prediction. Vol. 2. New York: springer, 2009.
27. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785-794. 2016.