

Paper Type: Original Article

Hybrid Machine Learning Models for Enhanced Student Performance Prediction

Hosam Eldin Fawzan Sayed ^{1,*}  and Mohamed M. Fouad ² 

¹ Department of Computer science, Faculty of Computers and Information, Arish University - North Sinai, Egypt; Hosam.Fawzan@ci.aru.edu.eg.

² Faculty of Computers and Information Technology, Future University, Cairo, Egypt; mmafoad@fue.edu.eg.

Received: 02 Apr 2025

Revised: 13 Apr 2025

Accepted: 18 Nov 2025

Published: 24 Nov 2025

Abstract

Predicting student academic performance is a critical task for educational institutions seeking to enhance learning outcomes and identify students at risk of underachievement. With the integration of artificial intelligence in education, machine learning (ML) techniques provide effective tools for modeling educational data and supporting data-driven decision-making. This study proposes a hybrid approach combining unsupervised and supervised learning to identify the most influential features affecting student performance and accurately classify academic outcomes. Initially, K-Means clustering was applied to the preprocessed dataset, and clustering quality was assessed using the Davies–Bouldin Index, yielding a score of 3.25. Subsequently, four classification algorithms—Support Vector Machine (SVM), Decision Tree, Naïve Bayes, and K-Nearest Neighbors (KNN)—were implemented and evaluated using standard performance metrics. Following hyperparameter optimization via Grid Search CV, SVM achieved the highest predictive performance, with an accuracy of 87.08%, precision of 87.21%, recall of 87.08%, and F1-score of 86.36%, whereas KNN exhibited the lowest performance, with 73.68% accuracy and 67.52% F1-score. These findings highlight the superiority of SVM for academic performance prediction and the relative limitations of KNN. Overall, the study demonstrates the effectiveness of optimized ML models in educational data mining and provides valuable insights for the development of intelligent student support systems. Code: <https://github.com/Salmao6u/CNN.git>

Keywords: Student Performance Prediction; Educational Data Mining; Machine Learning; K-Means Clustering; Support Vector Machine; Classification Algorithms; Academic Achievement.

1 | Introduction

Education plays an important role in shaping individuals' lives by constructing self-confidence and providing access to required resources. With technologies such as artificial intelligence rapidly developing, institutions of higher education have been incorporating such innovations at a fast rate into conventional learning environments [1]. Academic success is largely recognized to be a significant determinant of educational quality and learning outcome. It is influenced by various factors including demographic profiles, pedagogies, and learning environments. Predicting academic performance has become a field of growing importance, as academic performance is a crucial factor in institution rankings and enhances students' career prospects [2]. However, universities continue to grapple with tracking student performance, academic achievement, and



Corresponding Author: Hosam.Fawzan@ci.aru.edu.eg



<https://doi.org/10.61356/j.scin.2025.2616>



Licensee SciNexuses. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

creating early intervention strategies. Support programs, which most universities use today, begin at admission and grow in response to student progress. With the expansion of online education, e-learning is now a scalable and adaptive type of education. Such systems as Intelligent Tutoring Systems (ITS), Learning Management Systems (LMS), and Massive Open Online Courses (MOOCs) use educational data mining (EDM) methods in constructing adaptive learning environments, recommendation systems, and automatic grading tools. Whereas e-learning is more accessible and affordable, it also raises issues such as lack of adequate real-time interaction between students and instructors, high dropout rates, and difficulties in identifying student needs. Furthermore, the absence of standard evaluation tools makes performance measurement even more difficult. Nevertheless, the massive amount of data provided by e-learning environments holds valuable information that can be evaluated to predict student success and support customized learning paths [3]. A wide range of machine learning algorithms has proven useful in addressing specific learning issues, especially where conventional knowledge-engineering techniques cannot be applied due to insufficient domain knowledge [4].

Fundamentally, machine learning (ML) deals with discovering algorithms that learn from provided input data to develop generalized hypotheses capable of making accurate predictions on fresh instances [5]. At the same time, data mining is most crucial for extracting meaningful information from large bodies of data, thereby enabling well-informed decision-making processes. In the educational world, data mining has become increasingly pertinent through its potential to uncover patterns based on which pedagogical policies and planning at the institutional level can be decided [6]. Learning analytics is a companion field that addresses collecting and analyzing data generated by students to optimize instruction content and enhance learning [7]. A working example is the classification of students by behavioral and demographic profiles for suggesting certain adjustments to course organization and presentation.

The overall goal here is to identify key factors influencing performance and drop rates and examine relationships between these metrics using the theories of learning analytics and educational data mining (EDM) [8]. EDM involves revealing interesting patterns from learning data, making it easier for educators to predict outcomes, improve instruction practice, and improve measurement of student performance. By accomplishing this, students gain more tailored learning paths, and educational institutions can leverage lessons to enhance their overall performance and efficiency of learning systems [9]. In the last few years, EDM has been a principal driver of transformation of conventional education that enables adaptive data-driven learning systems to be aligned with specific needs of individual learners. The present study makes a valuable contribution to the field of Educational Data Mining (EDM) through extending student performance prediction with machine learning techniques. It addresses different problems that plague contemporary educational institutions and offers methodological innovations that enhance forecasting and analysis of academic performance. The research demonstrates the potential of integrating machine learning models into conventional teaching frameworks to augment the assessment and understanding of student performance.

The key aspect of this study is the application of K-Means clustering, assessed using the Davies–Bouldin Index, to reveal underlying patterns and identify critical factors influencing student success. Additionally, the predictive performance of several classification algorithms—Support Vector Machine (SVM), Decision Tree, Naïve Bayes, and K-Nearest Neighbors (KNN)—was compared. By emphasizing interpretable models such as SVM, Decision Tree, and KNN over more complex deep learning approaches, the study prioritizes real-world usability with minimal compromise in predictive accuracy. Best practices, including 5-fold cross-validation and rigorous hyperparameter tuning via Grid Search CV, were employed. Overall, the proposed framework stands out for its combined use of clustering and classification, comprehensive performance evaluation, and direct applicability to educational settings, highlighting both its practical relevance and scholarly contribution.

2 | Related Work

The newer development of Educational Data Mining (EDM) has significantly contributed towards data-based improvement of learning spaces. EDM couples' educational theory with data mining for the extraction of

patterns that guide decisions and facilitate educational performance [10]. ML algorithms enable educational institutions to build interactive applications, which respond dynamically to students' needs and individualize the process of learning. ML, a branch of artificial intelligence, enables programs to learn and predict from data without necessarily being explicitly programmed [11]. There exist prior works that have demonstrated the efficacy of employing ML models in predicting the performance of students. For instance, University of Córdoba employed a grammar-driven genetic programming algorithm called the G3PMI that yielded 74.29% accuracy. Other works on the Vishwakarma Engineering Research Journal and Somaiya College performed Naïve Bayes and ANNs evaluation with attained accuracy of 85% [12, 13].

Kotsiantis et al. discovered Naïve Bayes to be extremely useful with the average accuracy of 73%, and the Eindhoven University of Technology had recommended the use of the J48 decision tree for prediction of student dropout. University of Minho compared some of the models like SVM, decision trees, and neural networks to predict subject-based performance [14, 15]. Six commonly used ML algorithms in educational contexts were discovered in a systematic review [16] to comprise SVM, KNN, Naïve Bayes, and ANNs. Despite ANNs being typically accurate, they demand high technical expertise and computational capabilities. Hence, less interpretable but easier-to-deploy models like SVM, decision trees, KNN, and Naïve Bayes persist to dominate EDM research.

Machine learning has been increasingly applied to predict student academic performance. Studies have shown that careful feature selection, model choice, and data preprocessing significantly affect prediction accuracy, with SVM and Random Forest often performing well [17-19]. A systematic review further highlights trends, challenges, and best practices in this domain, emphasizing the potential of ML to support data-driven educational interventions [20]. The work [26] presents matrix factorization collaborative filtering with some performance enhancement to overcome cold start problem. In addition, it presents a comparative study among memory-based and model-based approaches.

On this foundation, the present study applies these models to forecast student academic achievement. By systematically tuning their hyperparameters with Grid Search CV and comparing their classification performance, the study presents an applicable and comparative evaluation of several algorithms for use in educational data mining.

3 | Materials and Methods

Machine learning (ML) has been widely used in recent years for the development of intelligent systems that can perform computational and complex tasks in various domains such as natural language processing, medical diagnosis, and pattern recognition [21]. Cluster analysis, a method of unsupervised learning, is one major technique among ML utilized for the clustering of similar data points without the knowledge of their labels [22]. K-Means clustering [23] was applied in this study to the preprocessed student performance dataset as a first exploratory method to discover underlying patterns and group students into natural clusters. Davies–Bouldin Index (DBI) was also utilized to check the quality of the formed clusters, with the smaller the DBI value the higher intra-cluster compactness and inter-cluster separation. Although the clustering results were not used specifically within the classification process, this analysis provided additional insight into the feature distribution and structure, which helped towards understanding the underlying patterns influencing student academic performance.

3.1 | Methodology

The proposed architecture for this research comprises four main tasks: data preprocessing, unsupervised clustering analysis, supervised machine learning classification, and model evaluation. The order of these tasks is to discover patterns in the data and accurately classify students based on academic performance. As shown in Figure 1, the process begins with preprocessing the dataset by encoding categorical variables and scaling numerical attributes. K-Means clustering is then applied as an exploratory step to discover latent structures in the data, and cluster quality is assessed according to the Davies–Bouldin Index (DBI). Next, four classifiers

Decision Tree, Support Vector Machine (SVM), Naïve Bayes [24], and K-Nearest Neighbors (KNN) [25] are trained and compared. Hyperparameter tuning [26, 27] for the individual models is performed using Grid Search CV to fine-tune model performance. Finally, the model performances are compared using traditional evaluation metrics like accuracy, precision, recall, and F1-score, to determine the best predictor of student academic performance.

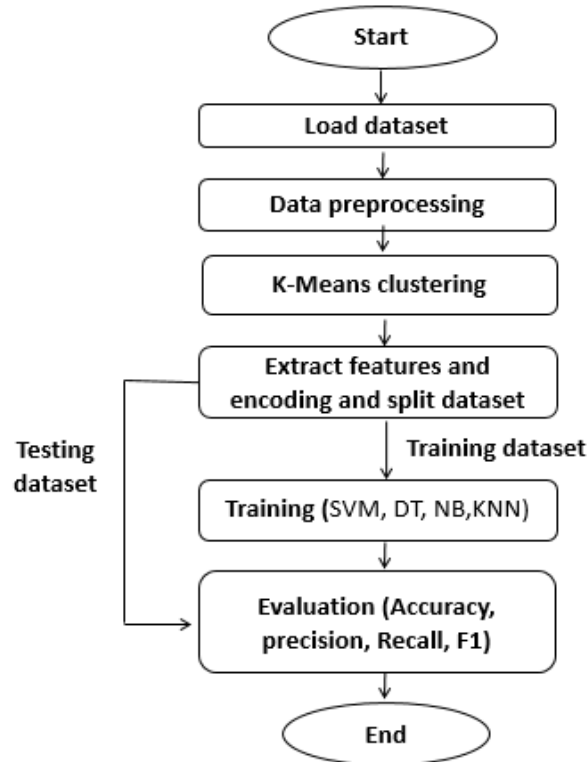


Figure 1. Flowchart of the Proposed Student Performance Prediction Model.

For this study, the data employed was an existing online repository (UCI Student Performance dataset). Preprocessing consisted of labels encoding categorical attributes and scaling of numerical attributes for normalization. K-Means clustering was applied to identify inherent patterns in the preprocessed data and Davies–Bouldin Index (DBI) to estimate the clustering quality. Feature selection was undertaken on relevant domain-specific attributes of the dataset.

Hyperparameter tuning was used to tune each of the four classifiers—Support Vector Machine (SVM), Decision Tree, Naïve Bayes, and K-Nearest Neighbors—through Grid Search CV to achieve optimal classification performance. For SVM, this entails kernel type adjustment and regularization parameter. Training and testing for accuracy, precision, recall, and F1-score were applied across all models for identifying the most effective predictor of student academic attainment. Hyperparameter-tuned SVM model generated best overall performance from the evaluated classifiers.

3.2 | Dataset

The dataset utilized in this study was acquired from the UCI Machine Learning Repository, to be specific, the "Student Performance Dataset". The dataset consisted of information relating to secondary students of two Portuguese schools, presented in two CSV files separately based on Mathematics and Portuguese language subjects. In the current study, both files were merged into a single dataset that contained 1,044 records of students with over 30 attributes covering demographics, academic work, and conduct. Each record has characteristics such as gender, age, address type, parents' educational level and profession, study hours, number of failures in prior classes, number of absences, and grades on finals (G1, G2, G3). Final grade G3

was used for classification with the intention to achieve a target variable of type binary, reflecting whether the student passed or failed.

During preprocessing, categorical features were label-encoded and numerical features were scaled to ensure equal scaling. Although there were barely any missing or repeated values in the data, such errors were identified and removed to ensure data integrity and model consistency.

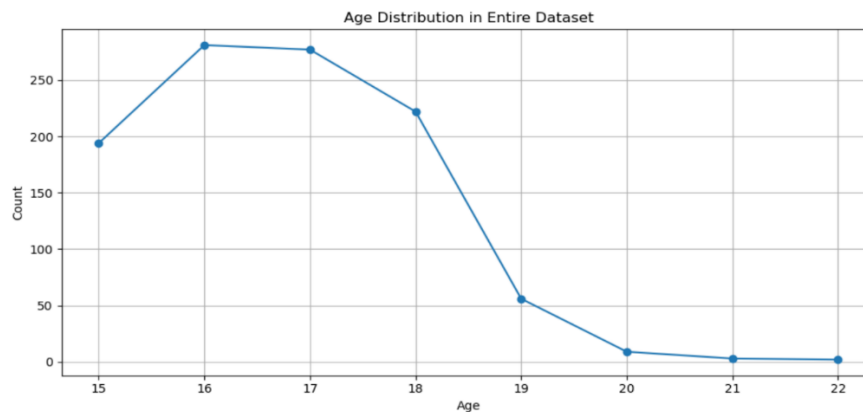


Figure 2. Region frequency in the dataset.

3.3 | Data Preprocessing

To achieve the best performance of the machine learning algorithms, preprocessing was carried out on the dataset. The two datasets available in the UCI repository (student-mat.csv and student-por.csv) were combined to increase the sample size along with diversity. Categorical variables such as school, sex, address, famsize, famsup, Mjob, Fjob, and internet were label encoded into numerical form because most machine learning algorithms require numeric input.

Quantitative variables such as age, absences, and the previous grades (G1, G2) were standardized by Standard Scaler to normalize feature distributions and prevent bias from scale differences.

Additionally, a binary target variable was created from the final grade (G3) by classifying students with 10 or above as "pass" (1) and below 10 as "fail" (0). The dataset was quite clean, and no significant missing or duplicate records were present. These preprocessing activities left the dataset nicely structured and ready for use in clustering and classification processes utilized in this study.

3.4 | Feature Selection

Feature selection was applied in this study to enhance model efficiency and find the most predictive features. The data were preprocessed, and a Random Forest classifier was employed to estimate feature importance scores. Based on these scores, the Select from Model procedure was employed to automatically select and retain the most informative features by imposing a threshold on important values. This approach is especially appropriate for high-dimensional numeric data and effectively reduces dimensionality without significantly impacting model performance. The selected features were then utilized in training models for classification. This process not only improved computational efficiency but also led to more stable and understandable machine learning results.

Table 1. The top-ranked attributes.

Feature Name	Description
Failures	Number of past class failures
Absences	Number of school absences
G1	First period grade (out of 20)
G2	Second period grade (out of 20)

Upon using the Random Forest model on the feature importance analysis, a subset of features was identified as most predictive of student academic performance. The top-ranked attributes selected by the model were (Table 1):

- Failures: Number of past class failures
- Absences: Number of absences from school
- G1: Grade in the first period
- G2: Grade in the second period

These features had the highest contribution to the classification task based on their importance scores. By using this chosen subset, the dimensionality of the dataset was reduced, making the model more interpretable and efficient without loss of predictive power. This guaranteed that the classification models were trained with the most impactful variables, thereby enhancing the credibility of student performance prediction.

3.5 | Clustering

Clustering is a method of unsupervised learning that is utilized for finding hidden patterns and grouping similar observations within a dataset. K-Means clustering was utilized in this research as an exploratory method of analyzing student data and determining natural groupings. The algorithm splits the dataset into a specified number of clusters in such a manner that data points within a cluster are closer to one another than to data points within other clusters, with each cluster being characterized by the meaning of its members.

Cluster size was fixed at $k = 3$, based on preliminary experimentations and the need for meaningful division of the student population. Having applied K-Means on preprocessed data, the DBI was computed for evaluating the quality of clusters. A smaller value of DBI is indicative of superior clustering performance, as measured by intra-cluster similarity and inter-cluster disjointedness.

Although clustering was not explicitly used for the classification problem, it provided an insight into the structure of the dataset and suggested potential groupings based on academic and behavioral features. These can inform the development of early intervention systems and support structures within schools.

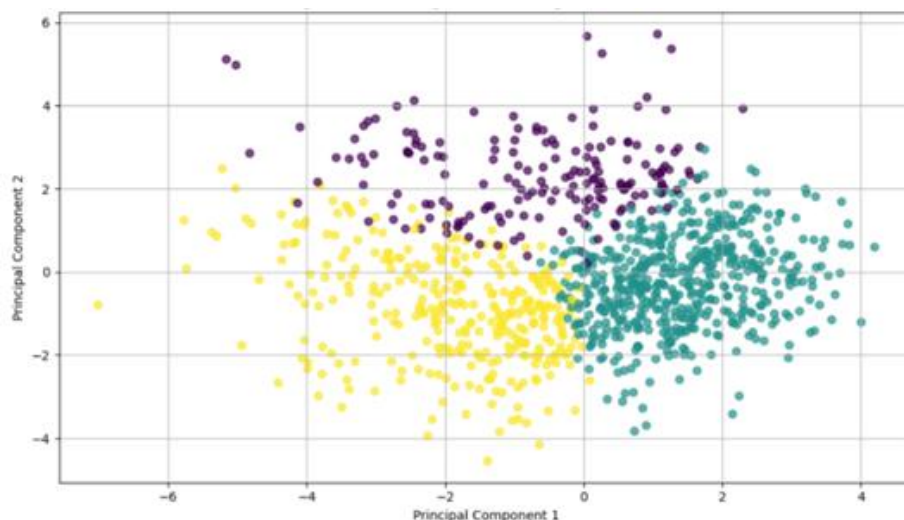


Figure 3. Clustering Students Using K-Means ($k = 3$). The clusters represent natural groupings among students based on academic and behavioral attributes after dimensionality reduction using PCA.

3.6 | Hyperparameter Optimization

Hyperparameter tuning is a critical step to improve the performance of machine learning models. It is the process of choosing the optimal set of parameters that control the learning process but are not learned during training, e.g., decision tree depth or SVM kernel type. Best hyperparameter choice ensures models generalize to new data and avoid underfitting or overfitting. Hyperparameter optimization was performed in this study

with the Grid Search CV method of the scikit-learn library. Grid search attempts all parameter combinations specified to determine the best configuration with the highest cross-validation scores. Grid search provides a reproducible and systematic method for model performance testing over a set of hyperparameter values. The following models were optimized: Support Vector Machine (SVM), Decision Tree, Naïve Bayes, and K-Nearest Neighbors (KNN). For each model, a specific grid of hyperparameters was defined, e.g., kernel and C for SVM, max_depth and min_samples_split for Decision Tree, Var smoothing for Naïve Bayes, N neighbors and weights for KNN. To enable effective evaluation, all the models were tuned with 5-fold cross-validation, and the optimal setting was selected according to validation accuracy. Each tuned model was then validated for performance according to accuracy, precision, recall, and F1-score as shown in Table 2.

Table 2. Provides a description of the hyperparameter search space utilized by each algorithm.

Algorithm	Hyperparameter	Search Values / Range
Decision Tree	max_depth	[3, 5, 7, 10]
	min_samples_split	[2, 5, 10, 20]
Naïve Bayes	var_smoothing	[0.1, 0.5, 1.0, 1.5, 2.0]
K-Nearest Neighbors	n_neighbors	[3, 5, 10, 20]
	weights	['uniform', 'distance']
Support Vector Machine	kernel	['linear', 'rbf']
	C	[0.1, 1, 10, 100]
	gamma	[0.001, 0.01, 0.1, 1]

The top-performing configuration on cross-validation was selected as the final model for each algorithm. To ensure that these models generalized and were not overfitted in tuning, their performance was retested on an independent test set. This is a critical step in verifying the robustness of the tuned models. After this systematic hyperparameter selection and validation, we were then able to increase the predictive power of our classifiers. The most optimal hyperparameters selected for all models are in Table 3.

Table 3. Optimal parameters for each algorithm.

Algorithm	Hyperparameter	Best value used
Decision Trees	Maximum depth	3
	Minimum samples split	2
Naïve Bayes	Smoothing parameter	0.5
K-Nearest Neighbors	Number of neighbors	20
	Weight function	"uniform"
Support Vector Machine	Kernel	rbf
	Regularization parameter (C)	100
	Gamma	0.001

3.7 | Prediction Methods

Four classification models were used and compared in this study: K-Nearest Neighbors (KNN), Naïve Bayes, Decision Tree, and Support Vector Machine (SVM). These models were selected as they have been proved to be effective in resolving classification problems that involve tabular educational data. An explanation of each of the methods follows.

3.7.1 | K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) algorithm is a simple but effective classification method that classifies new data points based on the labels of their closest neighbors in the training set. It uses a distance measure to determine the "closeness" of data points in the feature space.

In this study, KNN was used to forecast if the student would pass or fail based on some educational and behavioral attributes. Euclidean distance, which is the default for scikit-learn's K Neighbors Classifier, was used to measure the distance from the new instance to all training instances. The Euclidean distance between two instances x and x_i in n -dimensional space is calculated as [28]:

$$\text{Distance}(x, x_i) = \sqrt{\sum_{j=0}^n (x_j - x_{ij})^2} \quad (1)$$

This is the measure of a straight line between two points. It is quite popular in educational data mining since it is very easy to compute and interpret. The hyperparameter value of the number of neighbors (k) was optimized using Grid Search CV, and the best value obtained through 5-fold cross validation was $k = 20$. The final model was evaluated using standard performance measures on an independent test set.

3.7.2 | Support Vector Machine (SVM)

Support Vector Machine (SVM) is a strong supervised learning algorithm that is commonly used for classification problems. It identifies the optimal hyperplane that separates data points of different classes with the maximum margin. The hyperplane is the decision boundary, and the data points on the nearest side of the hyperplane are called support vectors. SVM in this study was utilized to classify students into pass or fail based on selected academic characteristics. To maximize the performance of classification, two kernels were explored: linear and radial basis function (RBF). These kernels allow the model to learn linear or non-linear data relationships. Optimal SVM configuration performing best was identified through Grid Search CV with best values for C (regularization parameter) and γ (coefficient of RBF kernel). Final models using RBF kernel performed best among all performance metrics and performed better than the other classifiers discussed in this research.

$$W \cdot X + B = 0 \quad (2)$$

where w is the weight vector (i.e., coefficients assigned to every feature), x is the input feature vector, and b is the bias term or the intercept. Maximizing margin is the fundamental objective of SVM. Margin refers to the distance from the separating hyperplane (decision boundary) to the nearest points from each class. This idea is mathematically expressed as [29] in Eq. (4).

$$\text{margin} = \frac{2}{|w|} \quad (3)$$

SVM can readily deal with nonlinearly separable data by transforming the input features to a higher space using kernel functions. In higher dimensions, one can plot a linear decision boundary even if data is nonlinearly separable in the original feature space. In this study, we initially selected the linear kernel for the Support Vector Machine (SVM) due to its simplicity and interpretability. The linear kernel allows for straightforward visualization of the decision boundary and helps in understanding the impact of features on the classification process. Linear kernels are also computationally efficient, especially when the data has a high number of features relative to the number of samples. However, although the linear kernel offers simplicity, it may be insufficient in portraying complex relationships in data. To counteract this limitation and enhance model performance, we employed GridSearchCV to conduct hyperparameter tuning, such as exploring alternative types of kernels such as radial basis function (RBF). Grid search facilitated exhaustive testing of different model configurations, tuning of parameters like the regularization factor (C) and kernel factor (γ) of non-linear kernels. The approach facilitated balance between model interpretability and prediction accuracy through the selection of the best-fitting configuration based on validation performance. Lastly, while linear kernel created a good interpretability baseline, using grid search enabled us to optimize the SVM parameters to realize predictive accuracy in real-world environments with effectiveness and generalizability across a wide range of students.

$$K(x_i, x_j) = x_i \cdot x_j \quad (4)$$

Finally, to predict the class label of a new data point x , we just plug it into the equation for the decision boundary as in [30]:

$$\text{Predicted class label} = \text{sign}(w \cdot x + b) \quad (5)$$

3.7.3 | Decision Trees

Decision Trees are among the most well-known classification algorithms due to simplicity, interpretability, and achievement in decision-making applications. They learn predictive models in the form of a tree, with feature tests at the internal nodes, decision rules in the branches, and class labels or outcomes at leaf nodes. Decision trees are particularly popular in educational data mining due to:

- Their output is easy to understand and intuitive for visualization by non-experts,
- The learned model can be explicitly interpreted in the form of a set of IF-THEN rules, which makes it convenient to integrate them into decision support systems.

Tree induction is a recursive process that starts at the root node and repeatedly divides the data set based on some attributes to get the best predictive capability. Selection of attributes for each split is usually made by employing measures such as information gain, Gini index, or chi-square statistics. Recursion terminates when the stopping criteria are met, i.e., maximum depth or minimum samples per node. The resulting leaf nodes form the predicted classes.

For this study, the Decision Tree Classifier from scikit-learn was used. The optimal parameters for maximum depth and minimum samples split were determined using grid search. These hyperparameters control the tree's complexity by avoiding overfitting and maintaining classification accuracy. Entropy and information gain are utilized to quantify the quality of every split in the tree and are defined mathematically as [31] follows:

$$\text{Entropy } H(S) = -\sum_{i=1}^n p_i \log_2(p_i) \quad (6)$$

where p_i is the proportion of data points in class i in the set S .

3.7.4 | Naïve Bayes

Naïve Bayes is an efficient and straightforward probabilistic classification algorithm which uses Bayes' Theorem assuming strong independence between features. Although the assumption here is very basic, Naïve Bayes tends to produce competitive results, especially when features are conditionally independent or when data sets are high-dimensional. The model assumes that the presence (or absence) of any given feature has nothing to do with the presence (or absence) of any other feature, given the class label. This makes Naïve Bayes highly scalable as it requires only a linear number of parameters compared to the number of features in the data. It is also computationally lightweight and can work well even when training on relatively small training sets.

In this study, Naïve Bayes was used to make student predictions (pass or fail) as a binary classification. The GaussianNB implementation was used to handle continuous feature values assuming that they are normally distributed (Gaussian). Given a specific instance with features

features $X=(X_1, X_2, \dots, X_n)$, the model determines the posterior probability for each possible class C and chooses the most likely class. Mathematically this is represented as [32]:

$$P(C | X_1, X_2, \dots, X_n) \propto P(C) \times \prod_{i=1}^n P(X_i | C) \quad (7)$$

Where: $P(C)$ = prior probability of class C , $P(X_i|C)$ = probability of observing feature X_i when class C the symbol $\propto$ is used to mean proportionality, thus the output values are scaled so that sum of the overall probability distribution = 1.

3.8 | Performance Measures

The performance of the classification models in this study was evaluated based on several common metrics derived from the confusion matrix. Accuracy as the sole measure of performance can be misleading, particularly in data sets that contain imbalanced class distributions or where false positives, and false negatives have very different costs. Therefore, additional performance measures were utilized to provide a more comprehensive view [28, 29].

- **Accuracy:** the proportion of correctly predicted instances among all predicted instances.
- **Precision:** the ratio of correctly predicted positive instances among all predicted positive instances.
- **Recall:** (or sensitivity): the proportion of actual positive instances correctly predicted by the model.

These measures give more insight into the performance and reliability of the model for different kinds of predictions. The following equations were used to calculate these evaluation measures as [33]:

$$\begin{aligned}\text{Precision} &= \frac{TP}{TP + FP} \\ \text{Recall} &= \frac{TP}{TP + FN} \\ \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN}\end{aligned}$$

Where:

TP: True Positives, TN: True Negatives, FP: False Positives and FN: False Negatives

In addition to the usual metrics, Cohen's Kappa measure was also applied in this study. This measure is particularly suitable to measure model performance in multi-class classification and class-skewed distributions. Its ability to consider agreement by chance makes it a good complement to usual accuracy-based measures. Cohen's kappa is defined as [34]:

$$k = \frac{p_o - p_e}{1 - p_e} \quad (8)$$

where p_o is the observed agreement, and p_e is the expected agreement.

4 | Results and Discussion

4.1 | Environment Setup

Experiments in this study were performed on a machine with an 11th Gen Intel Core i7-1165G7 processor, 8.00 GB RAM, and a 64-bit Windows operating system. Development and test environment was established using Anaconda distribution with Python 3, and experiments were executed using Jupyter Notebook, which provided interactive coding and visualization. Python was selected as the primary programming language due to its extensive library ecosystem and ease of use. Key libraries used are:

- Pandas: for loading, manipulation, and processing of structured data. NumPy: for numerical calculations and statistical functions. Scikit-learn (sklearn): to encompass the application of machine learning algorithms, data pre-processing, model training, hyperparameter tuning (using GridSearchCV), and performance metrics.
- Machine learning algorithms applied in this research study were Support Vector Machine (SVM), Decision Tree, Naïve Bayes, and K-Nearest Neighbors (KNN). K-Means clustering was also employed for unsupervised classification and pattern discovery as well as for feature selection by

Random Forest. Each model was also optimized using grid search to enhance the performance metrics.

4.2 | Data Preprocessing

Throughout this study, data preprocessing was performed on the UCI Student Performance Dataset, which combines data of two Portuguese high schools. Both available datasets (student-mat.csv and student-por.csv) were combined to create a comprehensive dataset. Initial preprocessing involved label encoding of categorical data (gender, address, and parental occupation) to convert them into numerical format suitable for machine learning algorithms.

Second, numerical feature normalization for attributes such as age, absences, and grades was performed using Standard Scaler to normalize data and reduce variance bias. Finally, the last target variable was obtained from G3 grade by students graded 10 or above and marked as "pass", and the rest as "fail", thus having a binary classification issue.

The dataset contained minimal missing values and duplicates, and these were managed during preprocessing. The processes guaranteed that the final dataset was clean, complete numeric, and well-suited for clustering as well as classification.

4.3 | Data Visualization

To have a better understanding of the data and uncover hidden patterns, some data visualizations were carried out. These included student age, gender, academic performance, and other feature distributions related to them. The aim was to visually see the relationship of the different attributes such as grades, absences, and failures to the final academic outcome.

The outcome variable for this study outcome was derived from the students' third period grade (G3), which was recorded into a binary pass or fail designation. This is a general indicator of a student's academic standing at the end of the course.

Among plots produced, Figure 4 indicates the distribution of age across the dataset, highlighting that most of the students are between 16–18 years old. Figure 5 also indicates the distribution of the result classes (pass or fail), with more students scoring the minimum passing grade.

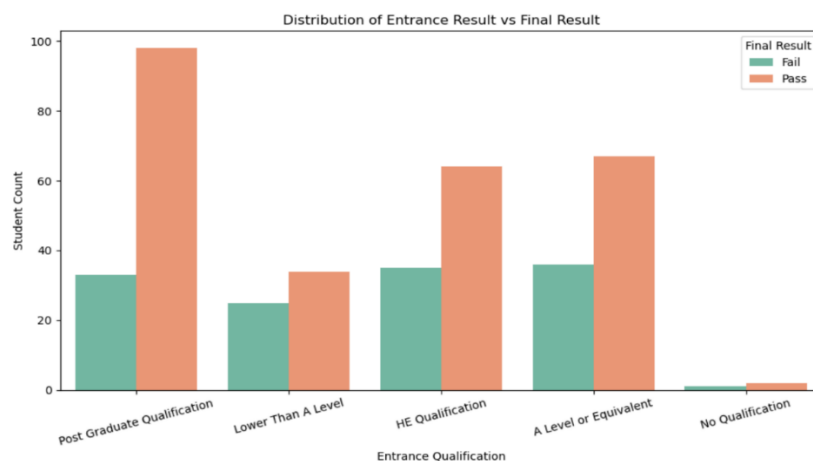


Figure 4. Entrance exam distribution along with the result.

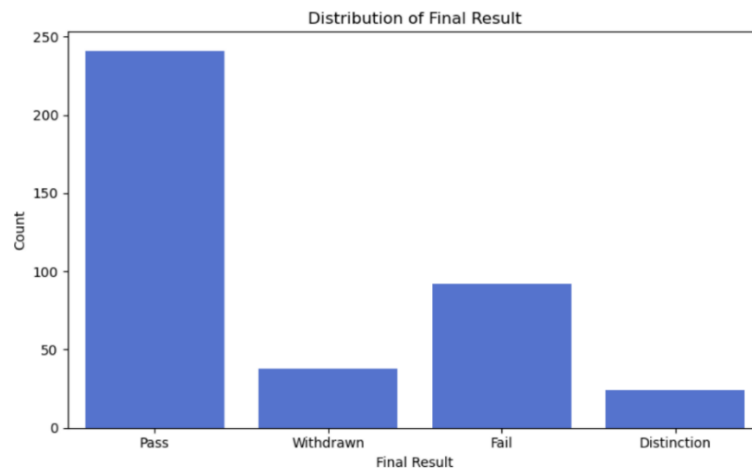


Figure 5. Result class distribution.

Figure 6 presents the distribution of students' results across the various regions. The visualization shows that regions such as South and Sidamo record the highest number of students achieving a passing grade, while the Southwestern Region reflects comparatively lower performance levels. This geographic distribution highlights regional disparities in academic outcomes, offering valuable insight for researchers to further examine underlying factors and develop region-specific support or intervention strategies.

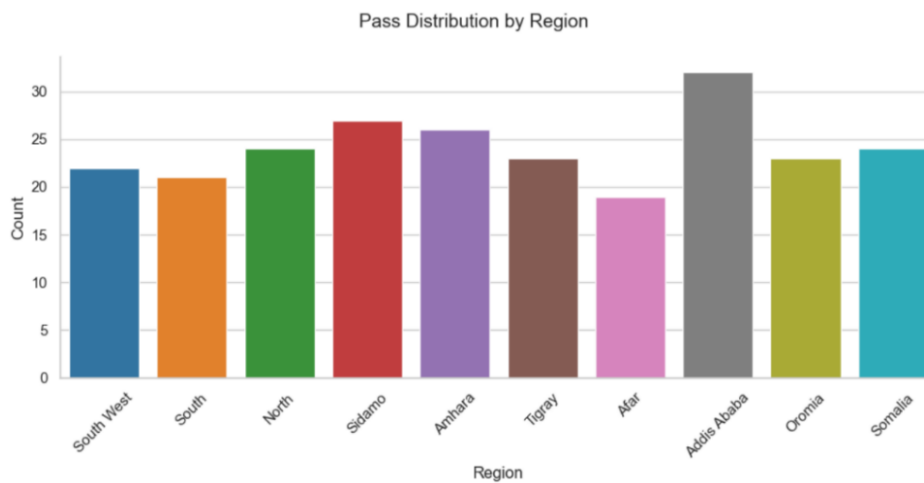


Figure 6. The distribution of region with the result.

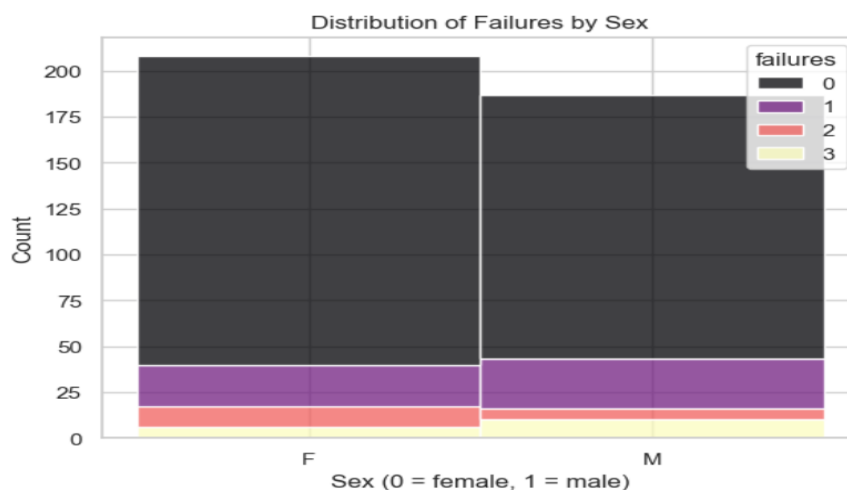


Figure 7. The distribution of failures by sex.

Figure 7 illustrates the relationship between students' number of attempts and disability status. While the dataset initially included a column for disability, further analysis revealed that this feature had limited impact on the prediction of academic outcomes. As a result, the feature was excluded during the feature selection phase to maintain model efficiency and accuracy. Only the most relevant features identified through Random Forest importance analysis were retained for training and evaluation in the predictive models.

4.4 | Cluster Analysis

In this study, K-Means clustering was applied to the scaled dataset to uncover hidden patterns in student academic performance. The algorithm grouped students into three distinct clusters (C0, C1, C2), which were subsequently interpreted as performance categories: Grade A, Grade B, and Grade C, based on the distribution of the dominant result within each cluster. Cluster C0 contained the highest proportion of Distinction students and was therefore labeled as Grade A. Cluster C1 showed a predominance of Fail outcomes, corresponding to Grade B performance, while Cluster C2 had the highest proportion of Pass students, making it representative of Grade C. This clustering approach helped in understanding the grouping behavior of students based on academic results and provided valuable insights for targeted academic support. The detailed distribution of result classes across clusters is presented in Table 4.

Table 4. Distribution of result classes.

Feature result	C0 (grades A) (%)	C1 (grades B) (%)	C2 (grades C) (%)
Distinction	48.47	0.69	2.04
Fail	0.00	2.08	10.20
Pass	51.53	81.66	47.96
Withdrawn	0.00	15.57	39.80

4.5 | Algorithms Comparison

A comparative analysis was undertaken to evaluate the predictive performance of four widely used supervised machine learning algorithms—Support Vector Machine (SVM), Decision Tree, Naïve Bayes, and K-Nearest Neighbors (KNN)—in forecasting student academic performance. Model performance was assessed using four standard evaluation metrics: accuracy, precision, recall, and F1-score, providing a comprehensive assessment of each classifier's effectiveness.

All models were trained on a preprocessed dataset, and hyperparameter optimization was conducted using the Grid Search CV mechanism to identify the optimal configuration of parameters. This tuning process, implemented with 5-fold cross-validation, significantly enhanced the models' predictive capabilities by systematically exploring parameter combinations and selecting those that maximized validation accuracy.

Although the models demonstrated strong performance prior to optimization—with SVM emerging as the leading classifier—the tuning process yielded substantial improvements across all algorithms. The optimized models achieved the following performance:

- Support Vector Machine (SVM): Accuracy 87.08%, Precision 87.21%, Recall 87.08%, F1-score 86.36%
- Decision Tree: Accuracy 87.08%, Precision 87.44%, Recall 87.08%, F1-score 86.25%
- Naïve Bayes: Accuracy 80.38%, Precision 79.57%, Recall 80.38%, F1-score 79.13%
- K-Nearest Neighbors (KNN): Accuracy 73.68%, Precision 72.06%, Recall 73.68%, F1-score 67.52%

These findings indicate that the SVM model with an RBF kernel and tuned hyperparameters provides the most robust predictive performance for this dataset, while KNN, despite its simplicity and interpretability, lagged in comparison. The overall improvements observed after hyperparameter tuning underscore the critical

role of parameter optimization in enhancing machine learning model performance and ensuring reliable classification outcomes.

Table 5. Distribution of result classes.

Prediction method	Precision	Recall	Accuracy	F1 Score
SVM	0.8721	0.8708	0.8708	0.8636
Naïve Bayes	0.7957	0.8038	0.8038	0.7913
Decision Tree	0.8744	0.8708	0.8708	0.8625
KNN	0.7206	0.7368	0.7368	0.6752

The study compared four machine learning models—SVM, Decision Tree, Naïve Bayes, and KNN—for predicting student academic performance. Naïve Bayes showed the lowest accuracy (80.38%) due to its assumption of feature independence, while SVM and Decision Tree achieved the highest accuracy (87.08%). Rigorous preprocessing and hyperparameter tuning, including Grid Search CV and 10-fold cross-validation for KNN, substantially improved model performance. The analysis also revealed demographic and regional factors influencing student outcomes, offering insights into educational interventions.

The superior performance of the Support Vector Machine (SVM), which achieved an accuracy of 87.08%, can be attributed to its ability to maximize the margin between classes, handle complex data relationships, and resist overfitting. Decision Tree classifiers followed closely with an identical accuracy of 87.08%, benefiting from their ease of interpretability and effectiveness in capturing nonlinear relationships and feature importance. Although the SVM in this study was configured with a linear kernel due to its simplicity, interpretability, and suitability for high-dimensional or linearly separable data, its performance remained robust across evaluation metrics.

Hyperparameter tuning using Grid Search CV notably enhanced the performance of all models, optimizing parameter combinations to yield more accurate predictions. The applied methodology—especially the use of repeated cross-validation—helped reduce bias and improve the generalization capability of the classifiers. Among the four models evaluated—SVM, Decision Tree, Naïve Bayes, and K-Nearest Neighbors (KNN)—SVM delivered the best results in terms of accuracy, precision, recall, and F1-score. Naïve Bayes, on the other hand, recorded the lowest accuracy at 80.38%, likely due to its strong independence assumptions among features, which may not be held in real educational datasets. This study demonstrates that predictive models built on student historical data can effectively forecast academic performance. Such models can support educational institutions in identifying students at risk and enabling timely interventions. While current results are promising, future work may explore additional algorithms or ensemble methods to further enhance prediction accuracy and system usability in real-world educational settings.

Table 6. Performance Comparison of Machine Learning Models Using the UCI Student Performance Dataset Across Different Studies.

Source / Study	Model	Accuracy	Precision	Recall	F1 Score	Notes
Hybrid Learning Models	Voting (SVM + Decision Tree)	96.63%	96.63%	96.63%	96.63%	Ensemble model combining SVM & DT
	SVM	87.08%	87.21%	87.08%	86.36%	Tuned using Grid Search CV
	Decision Tree	87.08%	87.44%	87.08%	86.25%	Best depth = 3
	Naive Bayes	80.38%	79.57%	80.38%	79.13%	Var smoothing = 0.5
	KNN	73.68%	72.06%	73.68%	67.52%	k = 20
F1000Research	SVM (all features)	90.0%	93.4%	92.9%	93.1%	50/50 train-test split
	SVM (grade history)	91.0%	92.8%	94.1%	93.4%	Focused on grade history features

	Naive Bayes (full)	89.4%	94.8%	90.7%	93.0%	Strong NB performance with all features
Preprints.org	Stacking Classifier	96.0%	96.0%	96.0%	96.0%	Binary: Pass/Distinction vs Fail/Withdrawn
	Random Forest	96.0%	96.0%	96.0%	96.0%	Very strong results on binary classification
	Extra Trees	94.0%	94.0%	94.0%	94.0%	Ensemble method with high accuracy
GitHub – rashiyadav05	SVM (Linear)	84.38%	—	—	82.0%	Used linear kernel
	SVM (Polynomial)	78.13%	—	—	74.0%	Lower performance with polynomial kernel

5 | Conclusion

The success of data mining applications in education heavily relies on the methodology adopted. In this study, rigorous steps were taken to ensure the reliability and accuracy of predictive models. Repeated k-fold cross-validation was employed to reduce sampling bias and enhance model generalizability. Furthermore, hyperparameter tuning through Grid Search CV significantly improved the performance of all classifiers.

The study demonstrated that machine learning techniques can effectively be used to predict student academic performance using historical records. Among the evaluated models—Support Vector Machine (SVM), Decision Tree, Naïve Bayes, and K-Nearest Neighbors (KNN)—the SVM classifier achieved the highest prediction accuracy of 87.08%. This performance is attributed to its ability to maximize the decision margin, handle non-linear relationships, and maintain robustness against overfitting. The Decision Tree model closely followed, also achieving 87.08% accuracy. Decision Trees proved valuable due to their interpretability, ability to handle non-linear interactions, and effectiveness in identifying the most influential features. In contrast, the Naïve Bayes model recorded the lowest performance, with an accuracy of 80.38%, which can be attributed to its assumption of feature independence and limited flexibility when dealing with correlated data. The implementation of hyperparameter tuning yielded measurable performance gains across all models. For instance, the optimized parameters led to improved F1-scores and precision values, contributing to a more accurate and balanced classification of student performance outcomes. Overall, the study confirms the utility of machine learning classifiers in educational data mining and their potential to support early intervention strategies by predicting student success. It also highlights the importance of proper preprocessing, feature selection, and model optimization in achieving high-performance outcomes. However, the study does present some limitations, including the use of a single dataset from a specific educational context, which may limit the generalizability of the findings. Future research should explore more diverse datasets, incorporate additional socio-behavioral and psychological variables, and assess the models across multiple institutions. Moreover, emphasis should be placed on improving model interpretability and validating predictions through external benchmarks to strengthen applicability and trust in such predictive systems.

Funding

This research has no funding source.

Data Availability

The datasets generated during and/or analyzed during the current study are not publicly available due to the privacy-preserving nature of the data, but are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there is no conflict of interest in the research.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

- [1] Baashar, Y., Alkaws, G., Ali, N., Alhussian, H., & Bahbouh, H. T. (2021). Predicting student's performance using machine learning methods: A systematic literature review. In 2021 International Conference on Computer and Information Sciences (ICCOINS) (pp. 357–362). IEEE.
- [2] Yadav, S. K., & Pal, S. (2012). Data mining: A prediction for performance improvement of engineering students using classification.
- [3] Liu, M., & Yu, D. (2023). Towards intelligent E-learning systems. *Education and Information Technologies*, 28(7), 7845–7876.
- [4] Mitchell, T. M. (2006). The discipline of machine learning. *Machine Learning*, 9.
- [5] Fy, O., et al. (2017). Supervised machine learning algorithms: Classification and comparison. *International Journal of Computer Trends and Technology*, 48(3), 128–138.
- [6] Delavari, N., Phon-Amnuaisuk, S., & Beikzadeh, M. R. (2008). Data mining application in higher learning institutions. *Informatics in Education*, 7(1), 31–54.
- [7] Nunn, S., Avella, J. T., Kanai, T., & Kebritchi, M. (2016). Learning analytics methods, benefits, and challenges in higher education: A systematic literature review. *Online Learning*, 20(2), 13–29.
- [8] Bharara, S., Sabitha, S., & Bansal, A. (2018). Application of learning analytics using clustering data mining for students' disposition analysis. *Education and Information Technologies*, 23(2), 957–984.
- [9] Baradwaj, B. K., & Pal, S. (2012). Mining educational data to analyze students' performance.
- [10] Aulakh, K., Roul, R. K., & Kaushal, M. (2023). E-learning enhancement through educational data mining with COVID-19 outbreak period in backdrop: A review. *International Journal of Educational Development*, 101.
- [11] Helm, J. M., Katt, B. M., Daly, C. A., Yang, S., Liao, S., Kumar, N., Kavousi, Y., Changoor, N. R., Varghese, V., Poggie, R. A., Schwarzkopf, R., & Mont, M. A. (2020). Machine learning and artificial intelligence: Definitions, applications, and future directions. *Current Reviews in Musculoskeletal Medicine*, 13(1), 69–76.
- [12] Suryadevara, C. K. (2018). Predictive modeling for student performance: Harnessing machine learning to forecast academic marks. *International Journal of Applied Science and Engineering*, 8(12).
- [13] Talwar, S., Talwar, M., Tarjanne, V., & Dhir, A. (2021). Why retail investors trade equity during the pandemic? An application of artificial neural networks to examine behavioral biases. *Psychology & Marketing*, 38(11), 2142–2163.
- [14] Kotsiantis, S., Patriarchas, K., & Xenos, M. (2010). A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education. *Knowledge-Based Systems*, 23(6), 529–535.
- [15] Sani, B. A., & Badamasi, H. Machine learning algorithms to predict student's academic performance. *Bakolori Journal of General Studies*.
- [16] Alsariera, Y. A., Alshamasin, M. S., Al-Tashi, Q., Rais, H. M., & Mirjalili, S. (2022). Assessment and evaluation of different machine learning algorithms for predicting student performance. *Computational Intelligence and Neuroscience*, 2022. <https://doi.org/10.1155/2022/1527601>
- [17] Hasan, H. M. R., Rabby, A. K. M. S. A., Islam, M. T., & Hossain, S. A. (2019). Machine learning algorithm for student's performance prediction. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1–7). IEEE.
- [18] Bhutto, E. S., Siddiqui, I. F., Arain, Q. A., & Anwar, M. (2020). Predicting students' academic performance through supervised machine learning. In 2020 International Conference on Information Science and Communication Technology (ICISCT) (pp. 1–6). IEEE.
- [19] Alamri, L. H., Almutairi, M. S., Alshammari, A. S., Alharbi, N. S., & Alsubaie, N. M. (2020). Predicting student academic performance using support vector machine and random forest. In 2020 3rd International Conference on Education Technology and Management (pp. 100–107).
- [20] Albreiki, B., Zaki, N., & Alashwal, H. (2021). A systematic literature review of student performance prediction using machine learning techniques. *Education Sciences*, 11.
- [21] Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31.
- [22] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications, and research directions. *SN Computer Science*, 2.
- [23] Cui, M., Sun, W., Peng, X., Yu, J., & Guo, Z. (2020). Introduction to the k-means clustering algorithm based on the elbow method. *Auditing and Accounting*, 1(1), 5–8.

- [24] Al-Aidaroos, K. M., Bakar, A. A., & Othman, Z. (2010). Naive Bayes variants in classification learning. In 2010 International Conference on Information Retrieval & Knowledge Management (CAMP) (pp. 276–281). IEEE.
- [25] Mucherino, A., Papajorgji, P. J., & Pardalos, P. M. (2009). K-nearest neighbor classification. In *Data Mining in Agriculture* (pp. 83–106).
- [26] Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 24.
- [27] Ahmed, E., & Letta, A. (2023). Book recommendation using collaborative filtering algorithm. *Applied Computational Intelligence and Soft Computing*, 2023. <https://doi.org/10.1155/2023/6695400>
- [28] Bishop, C. M. (2023). *Pattern Recognition and Machine Learning* (2nd ed., pp. 251–255). Springer.
- [29] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- [30] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 121–167.
- [31] Sepúlveda-Fontaine, S. A., & Amigó, J. M. (2024). Applications of entropy in data analysis and machine learning: A review. *Entropy*, 26(12), 1126. <https://doi.org/10.3390/e26121126>
- [32] Liu, K., & Tang, C. (2023). Privacy-preserving Naive Bayes classification based on secure two-party computation. *AIMS Mathematics*, 8(12), 28517–28539.
- [33] Akhila, M., Srjana, P., Gopinath, M., & Susmitha, V. (2024). *International Journal of Engineering & Science Research*, 14(2), 1004–1016.
- [34] Westover, A. M., Westover, T. M., & Westover, M. B. (2024). Interrater reliability estimation via maximum likelihood for Gwet's chance agreement model. *Open Journal of Statistics*, 14, 481–491. <https://doi.org/10.4236/ojs.2024.144028>

Disclaimer/Publisher's Note: The perspectives, opinions, and data shared in all publications are the sole responsibility of the individual authors and contributors, and do not necessarily reflect the views of Sciences Force or the editorial team. Sciences Force and the editorial team disclaim any liability for potential harm to individuals or property resulting from the ideas, methods, instructions, or products referenced in the content.