

Research Article

Student Performance Prediction Using Machine Learning Algorithms

Esmael Ahmed 

Information System, College of Informatics, Wollo University, Dessie 7200, Ethiopia

Correspondence should be addressed to Esmael Ahmed; esmael.ahmed@wu.edu.et

Received 2 January 2024; Revised 4 April 2024; Accepted 6 April 2024; Published 30 April 2024

Academic Editor: Ahmad Al- Omari

Copyright © 2024 Esmael Ahmed. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Education is crucial for a productive life and providing necessary resources. With the advent of technology like artificial intelligence, higher education institutions are incorporating technology into traditional teaching methods. Predicting academic success has gained interest in education as a strong academic record improves a university's ranking and increases student employment opportunities. Modern learning institutions face challenges in analyzing performance, providing high-quality education, formulating strategies for evaluating students' performance, and identifying future needs. E-learning is a rapidly growing and advanced form of education, where students enroll in online courses. Platforms like Intelligent Tutoring Systems (ITS), learning management systems (LMS), and massive open online courses (MOOC) use educational data mining (EDM) to develop automatic grading systems, recommenders, and adaptive systems. However, e-learning is still considered a challenging learning environment due to the lack of direct interaction between students and course instructors. Machine learning (ML) is used in developing adaptive intelligent systems that can perform complex tasks beyond human abilities. Some areas of applications of ML algorithms include cluster analysis, pattern recognition, image processing, natural language processing, and medical diagnostics. In this research work, K-means, a clustering data mining technique using Davies' Bouldin method, obtains clusters to find important features affecting students' performance. The study found that the SVM algorithm had the best prediction results after parameter adjustment, with a 96% accuracy rate. In this paper, the researchers have examined the functions of the Support Vector Machine, Decision Tree, naive Bayes, and KNN classifiers. The outcomes of parameter adjustment greatly increased the accuracy of the four prediction models. Naïve Bayes model's prediction accuracy is the lowest when compared to other prediction methods, as it assumes a strong independent relationship between features.

1. Introduction

Education is essential for a productive life, motivating self-assurance and providing necessary resources. With the advent of technology, such as artificial intelligence, higher education institutions are incorporating technology into traditional teaching methods [1]. Student academic performance is a crucial indicator of educational progress, influenced by factors like gender, age, teaching staff, and learning. Predicting academic success has gained interest in education. A strong academic record improves a university's ranking and increases student employment opportunities, as it is a primary factor evaluated by employers [2]. Modern learning institutions face challenges in analyzing performance, providing high-quality education, formulating

strategies for evaluating students' performance, and identifying future needs. Student intervention plans are implemented at the entry level and during subsequent periods, helping universities develop and evolve intervention plans effectively. E-learning is a rapidly growing and advanced form of education, where students enroll in online courses. Platforms like intelligent tutoring systems (ITS), learning management systems (LMS), and massive open online courses (MOOC) take advantage of EDM in developing automatic grading systems, recommenders, and adaptive systems. Despite e-learning being a less expensive and more flexible form of education, it is still considered a challenging learning environment due to the lack of direct interaction between students and course instructors. Three main challenges associated with e-learning systems include the

lack of standardized assessment measures, high dropout rates, and difficulty in predicting students' specialized needs due to lack of direct communication. Long-term log data from e-learning platforms can be used for student and course assessment [3].

Numerous machine-learning algorithms have been discovered to be efficient for specific learning tasks. They are particularly helpful in poorly understood fields where people might lack the expertise necessary to create efficient knowledge-engineering algorithms [4]. In general, machine learning (ML) investigates algorithms that conclude from examples provided externally (the input set) to develop general hypotheses that make predictions about instances to come [5]. On the other hand, data mining is crucial in sifting through a massive amount of data to find relevant information. Making decisions is aided by it. Data mining has many important uses in the field of education [6]. Learning analytics focuses on the gathering and analysis of data from learners to optimize learning materials and enhance learners' learning experiences [7]. This need can be met and potential improvements in course design and delivery can be suggested by classifying students based on their profiles. To analyze the factors impacting student performance and student dropout, the major goal is to identify meaningful indicators or metrics in a learning context and to examine the interactions between these metrics utilizing the ideas of learning analytics and educational data mining [8]. Finding noteworthy patterns in educational databases is a practice known as "educational data mining." It aids educators in foreseeing, enhancing, and assessing students' academic standing. Students can enhance learning activities, enabling management to enhance system performance [9]. Educational data mining (EDM) has significantly influenced recent developments in the education sector, providing new opportunities for technologically enhanced learning systems based on students' needs.

This research significantly contributes to the field of EDM by advancing the prediction of student performance using machine learning techniques. By addressing the challenges faced by modern learning institutions and leveraging innovative methodologies, the study offers valuable insights into enhancing academic outcomes. The research explores the integration of machine learning algorithms into traditional teaching methods, demonstrating how these can improve student performance analysis and educational outcomes. It uses *K*-means clustering with Davies' Bouldin method to identify clusters and significant features influencing student performance, providing a deeper understanding of academic success factors. The study also compares various machine learning algorithms, including Support Vector Machine (SVM), Decision Tree, Naïve Bayes, and *K*-Nearest Neighbors (KNN), to evaluate their predictive performance in predicting student outcomes. The research addresses technical gaps in predicting student performance by focusing on alternative algorithms over artificial neural networks (ANNs). The study employs rigorous methodologies, such as repeated *k*-fold cross-validation and hyperparameter optimization, to ensure robust and reliable prediction outcomes. The proposed model

stands out with its innovative clustering technique, comprehensive comparative analysis, and practical application in forecasting student performance. This emphasizes the relevance and impact of the research findings in educational practice.

2. Related Works

EDM's state-of-the-art methods and application techniques play a central role in advancing the learning environment. The discipline explores, researches, and implements data mining (DM) methods, incorporating multi-disciplinary techniques for its success. It extracts valuable and intellectual insights from raw data to determine meaningful patterns that improve students' knowledge and academic institutions [10]. The acquired information is processed and analyzed using different machine-learning methods to improve usability and build interactive tools on the learning platform. Machine learning is part of artificial intelligence (AI), where ML systems learn from data, analyze patterns, and predict outcomes. The growing volumes of data, cheaper storage, and robust computational systems have led to the rebirth of machine learning from pattern recognition algorithms to Deep Learning (DL) methods [11]. The University of Cordoba implemented a grammar-guided genetic programming algorithm, G3PMI, to predict student failure or success in a course. The algorithm has a 74.29% accuracy rate. The Vishwakarma Engineering Research journal created a platform for forecasting student performance using machine learning algorithms, using attendance and related subject marks [12]. Somiya College Mumbai developed a model for predicting student performance, which accurately expressed correlations with past academic results. With data set growth, neural network output improved, reaching 70.48 percent precision. Artificial neural networks (ANNs) were used by Talwar et al. to forecast student success in exams, achieving a high precision of 85% [13]. Kotsiantis et al. estimated student success using machine learning techniques, finding the Naïve Bayes strategy with a higher average accuracy of 73%. The Eindhoven University of Technology assessed the efficacy of machine learning for dropout student outcome prediction using various machine learning approaches, with the J48 classifier being the most effective model [14]. Researchers from three Indian universities analyzed a data set of university students using different algorithms, comparing the accuracy and recall values. The ADT decision tree architecture provided the most correct outcomes. The University of Minho, Portugal, evaluated the accuracy of decision trees, random forests, vector support machines, and neural networks in evaluating students' success in math and Portuguese language subjects. Another paper predicted student success at the beginning of an academic cycle based on academic records, achieving an accuracy of 85% [15].

The study in [16] investigates machine learning (ML) approaches for predicting student performance in tertiary institutions. Using 29 studies, six ML models were identified: decision tree, artificial neural networks (ANNs), support vector machine (SVM), *K*-nearest neighbor (KNN), linear

regression, and Naive Bayes (NB). ANN outperformed other models and had higher accuracy levels. The analysis revealed an increasing number of research in this domain and a broad range of ML algorithms applied, suggesting ML can be beneficial in identifying and improving academic performance areas [16].

Research in [17] aims to predict student performance using Artificial Intelligence, aiming to help students avoid poor results and groom them for future exams. By identifying dependencies and course requirements, teachers can provide appropriate advice to students. The system can help teachers monitor students and provide tailored assistance, reducing student lag. The research achieved a 94.88% accuracy rate, benefiting both students and teachers.

Research work stated in [18] presents a model for predicting students' academic performance using supervised machine learning algorithms like support vector machine and logistic regression. The sequential minimal optimization algorithm outperforms logistic regression in accuracy. The research aims to help educational institutes predict future student behavior and identify impactful features like teacher performance and student motivation, ultimately reducing dropout rates.

As described in [19] student performance in the final exam could be affected by many factors. The study uses Support Vector Machines (SVM) and Random Forest (RF) algorithms to predict final grades in mathematics and Portuguese language courses. The results show that binary classification achieves a 93% accuracy rate, while regression has the lowest RMSE of 1.13 in RF. This early prediction can help educational organizations provide solutions for students with low performance, enhancing their academic results. The study aims to enhance the performance of educational organizations.

According to recent research [20], contemporary academic institutions have difficulties assessing student achievement, delivering high-quality instruction, and analyzing performance. According to a comprehensive analysis of the literature on EDM from 2009 to 2021, machine learning (ML) approaches are utilized to forecast the risk of and dropout rates among students. The majority of research employs data from online learning environments and student databases. To improve student performance and predict risk and dropout rates, machine learning techniques are essential. The researchers recommended that future studies ought to concentrate on developing effective dynamic and ensemble techniques for predicting student performance and delivering automated corrective measures. This will support educators in developing appropriate solutions and meeting precision education goals.

Therefore, despite the aforementioned research works, a lot of work should be done on predicting student performance. Because there were technical gaps observed in existing works such as less accurate predictions and undiscovered features. In EDM research, alternative algorithms such as decision trees, SVM, KNN, and Naïve Bayes are favored over ANNs for predicting student outcomes due to their accessibility and ease of use. While ANNs boast high prediction accuracy, their adoption is limited by the

specialized technical skills required for effective implementation. Consequently, these more accessible algorithms are widely used in educational contexts, leading to the underutilization of ANNs. This study aims to enhance prediction accuracy by comparing and refining the performance of SVM, KNN, DT, and Naïve Bayes, which are commonly employed and easier to apply in EDM practices. Therefore, this research work presents a support vector machine with some performance enhancement. In addition, it presents a comparative study among KNN, SVM, decision trees, and Naïve Bayes. Compared to existing approaches, our proposed platform relies on more accurate student performance predictors. Moreover, our approach shows the discovery of less accuracy and undiscovered features using hyperparameter tuning with enhanced performance.

3. Materials and Methods

Nowadays, machine learning (ML) is used in developing adaptive intelligent systems that can perform complex tasks that are beyond human abilities [21]. Some of the areas of applications of ML algorithms include cluster analysis, pattern recognition, image processing, natural language processing, and medical diagnostics, to mention just a few. Cluster analysis, also known as clustering, is an unsupervised machine learning technique for identifying and grouping related data points in large datasets without concern for the specific outcome [22]. In this research work, K-means, a clustering data mining technique, using Davies' Bouldin method is used to obtain clusters to find the important features affecting students' performance.

3.1. Methodology. The proposed model in this study has four components which are data preprocessing, hyperparameter tuning, recommender model, and model evaluation. However, these main components incorporate other elements. The general architecture of the model is presented in Figure 1.

First, dataset collection involves collecting the data from the Wollo University learning management system called A+. Next, we utilized three stages of data preprocessing. The data preprocessing consists of data cleaning, categorization, and reduction to make the dataset ready to train the data mining algorithms. Then, we utilized feature extraction to determine the most informative features. After this, we used hyperparameter tuning for the enhancement of the algorithm. Hyperparameter tuning is used for the automatic enhancement of the hyperparameters of a model. Hyperparameters are all the parameters of a model that are not updated during the learning and are used to configure the algorithm to lower the cost function of the learning rate for the gradient descent algorithm. We apply this to the features which are fed into the algorithm. In this study, hyperparameter tuning is used just to enhance the loop of model learning to find the set of hyper-parameters leading to the lowest error on the validation set. Thus, a validation set has to be set apart, and a loss has to be defined. In this study, we clustered to predict a student's final result based on

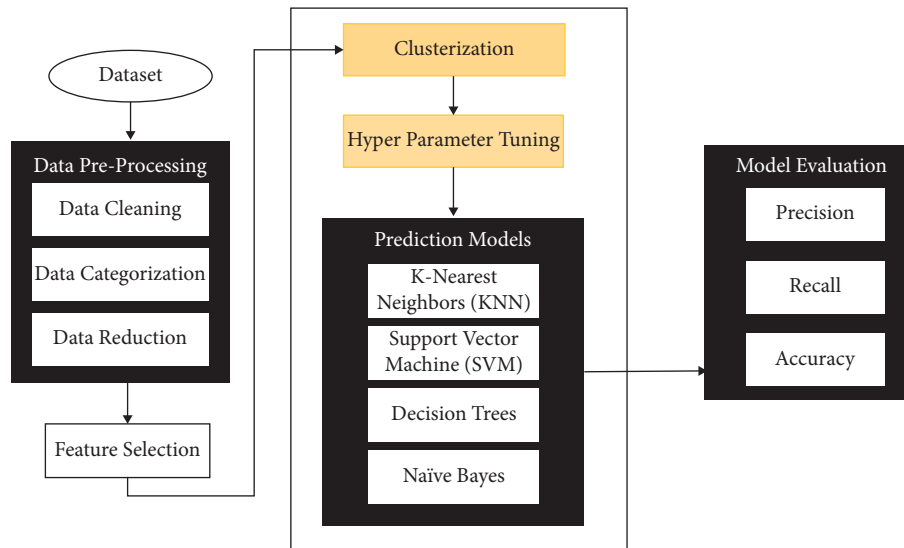


FIGURE 1: Workflow of the proposed methodology.

gender, region, entrance_result, num_of_prev_attempts, studied_credits, and disability using various prediction models and choosing the best prediction model. The clustering algorithm used in this study is K-Means. Model building involves developing a wide range of models using prediction methods. Finally, evaluating the model: It involves testing the validity of the model against each other and the goals of the study. Using the model involves making it a part of the decision-making process.

3.2. Dataset. The dataset was gathered from Wollo University and the Kombolcha Institute of Technology. The student's data from the academic years 2017–2022 was exported from the student information portal system. There were 8 columns in the final dataset. The dataset's columns contain the student's ID, gender, region, entrance_result, num_of_prev_attempts, studied_credits, disability, and final_result. Upon removing missing data, the dataset had information on 32,582 students. These data are inconsistent and dirty, so data preprocessing has been done. Since the quality of input data has an impact on the predictive model, data preparation has paramount importance. The researcher pre-processes the data using Python software. The major problems of the original data set that needs data pre-processing are attributes have so many missing values; the data contain duplicated records, in the original dataset. After eliminating incomplete data, the dataset comprised 32,005 students in the dataset. For instance, Figure 2 shows the region frequency distribution in the dataset.

3.3. Data Preprocessing. The dataset was preprocessed before being fitted into the models to guarantee the best possible performance from them. Our data was mostly non-numerical, so there was much preprocessing needed. In this study, we used three stages of preprocessing. Firstly, we utilized data cleaning to detect missing values and noisy data that could corrupt the dataset. Next, we employed data

categorization to handle numerical values. Label coding was employed to standardize the data. The purpose of the label encoder was to convert categorical values such as distinction, pass, withdrawn, pass, and fail into numeric numbers. Numerical values are more suitable for machine learning algorithms than categorical ones. Categorical data often take the form of strings or categories, have a finite number of possible values, and only have two categories. There is an inherent order in the first ordinal data categories. When encoding ordinal data, the information about the order in which the category is given is kept. They each have an ordinal relationship within the table "entrance result." To map it, their ordinal equivalent numbers are used. The second category is made up of nominal data, which lack an inherent order. Nominal data is encoded with the presence or absence of features taken into account. In the table, "region," "disability," and "final_result" are nominally related to each other. Finally, we made a data reduction to reduce and organize data to simplify the objective behind running and processing the data. Besides, the matrix was sparse, with most elements being zero should be dropped.

3.4. Feature Selection. In our dataset, we encountered a mix of numerical and categorical variables, necessitating a thoughtful curation process. We treated numerical features and categorical features differently to accommodate their distinct characteristics. Each feature was accompanied by a brief description, providing context to aid in the subsequent analysis. To identify the most informative characteristics within this diverse dataset, we employed the random forest algorithm as our primary tool. This algorithm is well-suited for feature selection due to its ability to handle various types of features effectively. Our goal was to iteratively train a random forest model under a 5-fold cross-validation setup. This method not only helps in assessing the model's performance but also allows us to determine the optimal number of features. The choice of 5-fold cross-validation is both strategic and computationally effective. This technique

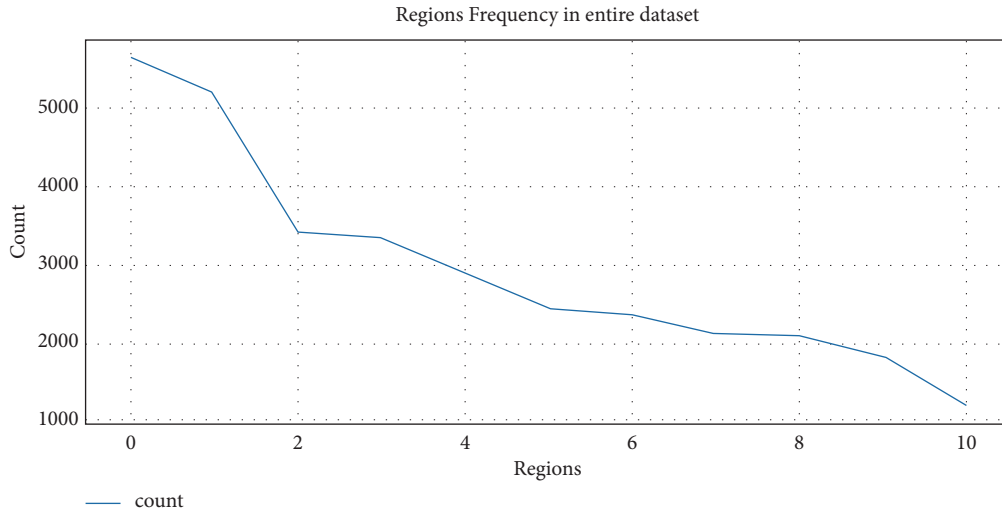


FIGURE 2: Region frequency in the dataset.

involves partitioning the dataset into five subsets or “folds” and using four of them for training while reserving the fifth for validation. This process is repeated five times, with each fold taking a turn as the validation set. The smaller number of folds is particularly suitable for our dataset, ensuring computational efficiency while still providing robust insights. Moreover, employing a relatively modest number of folds is advantageous because it allows each fold to represent a meaningful subset of the data. Given the dataset’s size, this approach ensures that each iteration captures a diverse and representative sample, contributing to the overall reliability of the model’s performance evaluation.

Following the application of the random forest algorithm for feature selection, a comprehensive analysis identified the following attributes as the most informative for subsequent predictive modeling:

- (i) Gender: The gender of the student.
- (ii) Region: The geographic region associated with the student’s place of origin.
- (iii) Entrance Result: The outcome of the entrance examination undertaken by the student.
- (iv) Number of Previous Attempts: The number of times the student has attempted the course or examination previously.
- (v) Studied Credits: The total number of credits the student has completed and currently undertaking.
- (vi) Disability: The presence or absence of any disabilities reported by the student.
- (vii) Final result: The previous academic outcome achieved by the student.

These selected features were deemed to possess the most significant impact on predicting student performance, based on the rigorous analysis conducted with the random forest algorithm. By leveraging this curated subset of attributes, we aimed to enhance the predictive accuracy of our subsequent modeling endeavors, thereby facilitating more informed decision-making in educational contexts.

3.5. Clusterization. An unsupervised learning method called clustering can be used to find patterns or structures in the data that are hidden. The data are divided into homogenous groups via clustering, which makes the observations in one group more similar to one another than to the observations in other groups. We have utilized *K*-means clustering among the many partition-based clustering algorithms.

The *k*-means clustering algorithm is utilized in this study to cluster the student data. The *K*-means clustering algorithm divides n observations into k clusters, each of which contains the observation that corresponds to the cluster with the closest mean. Every iteration’s *k*-means output clusters may be unique. *K*-means is therefore run numerous times on the dataset to obtain trustworthy clusters, and the clusters are created based on all of the iteration results. After clustering the students, the 3 clusters are assigned Grades A, B, and C depending on the metric values of the features that are the cluster with the highest metric values is assigned Grade A, the second-highest metric value with B, and the last cluster with C. For the selection of *K* in *K*-means clustering the Elbow method has been used. The elbow method is one of the most popular ways to find the optimal number of clusters [23]. This method uses the concept of WCSS (within-cluster sum-of-squares) value. The elbow method formula is shown in equation [23]:

$$\sum P_i P_i C_1 + \sum P_i P_i C_2 + \dots + \sum P_i P_i C_k, \quad (1)$$

$\sum P_i \text{ in Cluster } l \text{ distance } (P_i C_l)^2$: The sum of the squares representing the separations between each data point and its cluster centroid is what determines the other two elements. We have utilized the Euclidean distance to calculate the separation between the data points and the centroid. The plot’s sharp bend looks at the relationship between the number of clusters and the derived WCSS values. *K*, like an arm, is considered the best value of *K*. Figure 3 depicts the elbow method graph [23].

We employed repeated *k*-fold cross-validation to reduce the bias related to the samples. The entire dataset is split into *k* equal-sized, mutually exclusive subsets for *k*-fold cross-

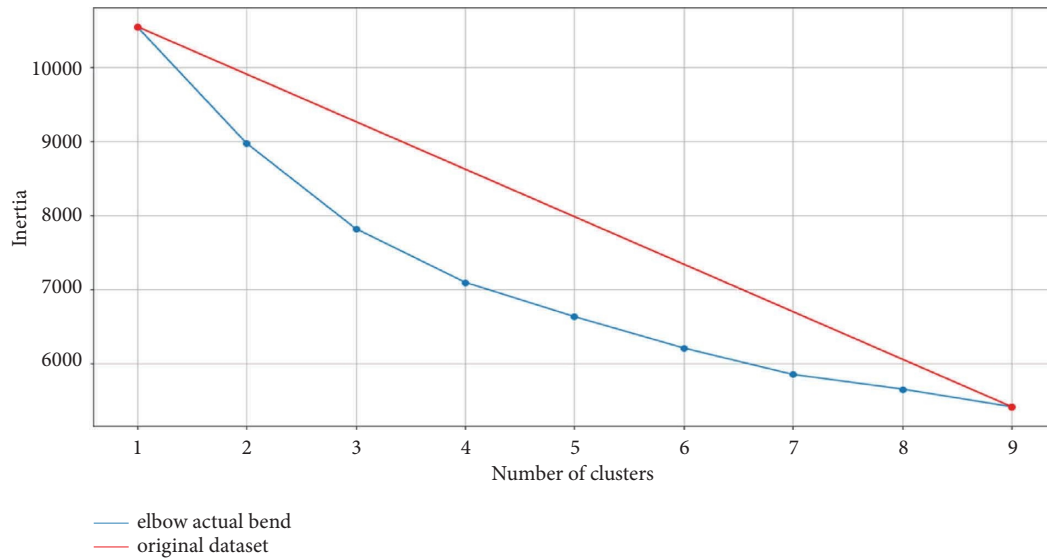


FIGURE 3: Elbow method graph.

validation. The classification and regression models are trained and tested k times, with each test being performed on the fold that was not used for training. One confusion matrix contains the prediction outcomes from the k experiments. The accuracy and other metrics are afterward calculated using this confusion matrix. In this investigation, the value of k was set at 10, or 10-fold cross-validation, and it was carried out three times.

3.6. Hyperparameter Optimization. Algorithm parameter tuning, before presenting the results or getting a system ready for production, tuning is a crucial step for enhancing algorithm performance. Optimization of hyperparameters is another name for it [24]. Making a computer system that can automatically create models from data without requiring laborious and time-consuming human involvement is the aim of machine learning. Setting parameters for learning algorithms before using the models is one of the challenges [24].

In machine learning, finding the best hyperparameter settings is like searching for a needle in a haystack. In our research, we use grid search to navigate this complex search space. We fine-tune model parameters by comparing predictions to actual values, aiming for the highest accuracy. However, tweaking hyperparameters presents unique challenges, which can be addressed with techniques like dataset pruning.

Automated hyperparameter optimization (HPO) is essential in modern machine learning to simplify the process and improve model performance. Despite its importance, HPO faces significant hurdles, such as expensive function evaluations and unclear optimization goals. While grid search is a common method, it has limitations in handling complex spaces and continuous parameters. Although HPO shows promise for transforming machine learning, its widespread use is limited by ongoing challenges. Overcoming these obstacles is essential to fully leverage

automated hyperparameter optimization in various fields, from industry to scientific research.

In this study, the following steps are involved: The first step is to select the appropriate type of model for predicting student performance. We choose using factors such as the nature of the data, the complexity of the problem, and the desired outcome. Therefore, we employed models including decision trees, SVM, KNN, and Naïve Bayes. Second, upon selecting the models, we examine their parameters and proceed to establish the hyperparameter space. These hyperparameters, encompassing factors like learning rate and regularization strength, significantly influence the learning process of the model. By utilizing our model particulars, we construct a hyperparameter space characterized by a spectrum of values for each parameter to be explored during the tuning phase. Third, to traverse the hyperparameter space, a grid search algorithm has been employed. Grid search meticulously explores all feasible combinations of hyperparameters within predetermined ranges, while random search randomly samples hyperparameter values within specified intervals. Bayesian optimization, on the other hand, employs probabilistic models to discern the most promising regions of the hyperparameter space for further exploration. Next, to evaluate model performance and mitigate overfitting, we employ the cross-validation scheme approach. This entails partitioning the data into multiple subsets and training and evaluating the model repeatedly, each time utilizing a different subset for validation. Through this iterative process, we can measure the model's ability to generalize to unseen data. Finally, we tuned hyperparameters via cross-validation, and then we assessed each model configuration's performance using predefined evaluation metrics. Table 1 shows the search values/range for hyperparameters of each algorithm.

The configuration demonstrating optimal performance on the validation set is designated as the final model. It is imperative to scrutinize the model's performance on an

TABLE 1: Search values/range for hyperparameters.

Algorithm	Hyperparameter	Search values/range
Decision trees	Maximum depth	[3, 5, 7, 10]
	Minimum samples split	[2, 5, 10, 20]
Naïve bayes	Smoothing parameter	[0.1, 0.5, 1.0, 1.5, 2.0]
K-nearest neighbors	Number of neighbors	[3, 5, 10, 20]
	Weight function	["uniform," "distance"]
Support vector machine (SVM)	Kernel	["linear," "rbf," "poly"]
	Regularization parameter (C)	[0.1, 1, 10, 100]
	Gamma	[0.001, 0.01, 0.1, 1]

independent test set to safeguard against overfitting during the tuning process. Adhering to these systematic procedures enables us to effectively fine-tune the hyperparameters of our machine learning models, thereby enhancing their performance and yielding superior results on our dataset. Table 2 shows the optimal parameters used for model enhancement.

Table 2 presents parameters that were selected based on the results obtained through grid search and optimization techniques. They represent the configurations that yielded the best performance for each respective algorithm.

3.7. Prediction Methods. Four prediction/classification algorithms are utilized in this study, and they are contrasted with one another. They are KNN, Naive Bayes, decision trees, and support vector machines. These algorithms are employed due to their excellent modeling abilities for classification-type prediction issues. The short descriptions of the prediction techniques are provided below.

3.7.1. K-Nearest Neighbor (KNN). K-nearest neighbors (KNN) is a fundamental machine learning algorithm widely used for classification tasks. It relies on the principle of similarity, where new data points are classified based on the majority class of their nearest neighbors in the feature space.

In the context of this study, KNN is applied to predict student performance categories, such as distinction, pass, withdraw, or fail, based on their features. The algorithm calculates the cosine similarity between the attributes of each student's record and those of other students in the dataset. Based on the class of the k-nearest neighbors, the KNN algorithms classify new data [25]. It involves finding the top K-nearest neighbors for the class of student performance (i.e., final result categorized as distinction, pass, withdraw, fail, etc.). Class of student performance from the list of nearest neighbors are combined to predict the unknown class. The K-nearest neighbor classifier usually applies either the Euclidean distance or the cosine similarity between the training tuples and the test tuple but, for this research work, the cosine similarity (equation (2)) approach has been applied in implementing the KNN model for our prediction model. The KNN algorithm involved in predicting student performance based on student historical record work is as follows [26]:

Step 1: Compute the mean final result value of every student according to the user-student performance class matrix.

Step 2: Calculate similarity based on the distance function

Step 3: Find K neighbors of the class by searching for the K class closest to a specific student performance class which is most similar to a specific student in terms of attributes.

Step 4: Predict the top N similar student performance class for similar students.

In the study, the value of k in the k-nearest neighbors (kNN) algorithm was determined through grid search, a technique used to train and evaluate models using different values of k . After employing 10-fold cross-validation, the optimal value of k was found to be 8 based on performance metrics.

$$\text{sim}(x, xi) = \frac{\sum_{j=1}^n (X_j \cdot Xi_j)}{\sqrt{\sum_{j=1}^n (X_j)^2} \cdot \sqrt{\sum_{j=1}^n (Xi_j)^2}} \quad (2)$$

Compute the distance between the data point to be classified (x) and each point in the training dataset (xi).

3.7.2. Support Vector Machine (SVM). The goal of support vector machines (SVM), which are a subset of generalized linear models, is to make predictions based on a linear combination of features obtained from the variables [27]. SVM translates the input data to a high-dimensional feature space, where the input data becomes more comprehensible, using both linear and nonlinear kernel functions. SVM determines the mathematical definition of a hyperplane that divides training data into classes, with data points from the same class located on the same side of the hyperplane. Once the best hyperplane is identified, it can be used to classify new data into one of the classes [27]. The decision boundary in SVM is represented by a hyperplane as shown in the following equation:

$$w \cdot x + b = 0, \quad (3)$$

where w is the weight vector (coefficients of the features), x is the input feature vector, and b is the bias term or intercept. SVM aims to maximize the margin (equation (4)) which is

TABLE 2: Optimal parameters for each algorithm.

Algorithm	Hyperparameter	Best value used
Decision trees	Maximum depth	7
	Minimum samples split	2
Naïve bayes	Smoothing parameter	1.0
K-nearest neighbors	Number of neighbors	10
	Weight function	"Distance"
Support vector machine (SVM)	Kernel	rbf
	Regularization parameter (C)	10
	Gamma	0.01

the distance between the decision boundary and the nearest data points of each class.

$$\text{Margin} = \frac{2}{\|w\|}. \quad (4)$$

SVM can handle nonlinearly separable data by mapping the input features into a higher-dimensional space using kernel functions. The decision boundary in the higher-dimensional space becomes linear, even if it was non-linear in the original feature space.

In this study, the linear kernel was selected for the Support Vector Machine (SVM) as defined in equation (5). The choice of the linear kernel was made due to its simplicity and interpretability, making it easier to understand the decision boundary and the relationship between features and the target variable. Additionally, linear kernels are computationally efficient and can perform well when the data is linearly separable or when the number of features is high compared to the number of samples. While the linear kernel is advantageous in terms of clarity, it may not capture these complexities effectively. Therefore, to address this limitation and enhance the model's predictive performance, we employed hyperparameter tuning techniques, notably grid search.

The use of grid search facilitates a systematic exploration of various model configurations, including different kernel functions (such as linear, polynomial, or radial basis functions) and their associated parameters. This approach ensures that we strike a balance between interpretability and predictive accuracy, catering to the nuances present in the dataset while still maintaining clarity in decision-making.

Essentially, the choice of a linear kernel at the very beginning is per the need for intelligibility and simplicity, but the use of grid search later on enables us to optimize the model configuration taking complexity and performance into account. The aim is to develop a model that can properly anticipate outcomes in real-world settings and analyze data efficiently. To this end, we desire to include a grid search in the SVD model.

$$K(x_i, x_j) = x_i \cdot x_j. \quad (5)$$

Finally, to predict the class label of a new data point x , we simply plug it into the equation of the decision boundary:

$$\text{Predicted class label} = \text{sign}(w \cdot x + b). \quad (6)$$

3.7.3. Decision Trees (DTs). One of the most used methods for prediction is the decision tree. Decision trees are preferred by most researchers for the following reasons: (1) Decision tree outputs are more accessible and clearer for the user, making them more transparent to the user. (2) They can be simply incorporated into the decision support system by being transformed into a collection of IF-THEN rules. To build a tree with the maximum potential prediction accuracy, this technique recursively divides data into branches. Different algorithms, including information gain and chi-square statistics, are utilized to build the tree, and the variables for each node are chosen based on these results [28]. The complete tree is built by repeating this procedure for every node. Decision trees frequently produce outcomes that are easier to understand and more accurate in decision-making. The decision tree's initial node is known as the root node, and its subsequent nodes are known as the leaf nodes. The end node refers to the tree's final node. The specific algorithm employed and the number of values for the chosen variable determine how many branches the decision tree will have [28].

Decision Trees aim to find the optimal splits that maximize the information gain or minimize the entropy at each node. Entropy is a measure of impurity in a set of data points, and information gain quantifies the reduction in entropy achieved by a split. The equations for entropy and information gain are as follows:

$$\text{Entropy}(H(S)) = - \sum_{i=1}^n p_i \log_2(p_i), \quad (7)$$

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

3.7.4. Naïve Bayes. A straightforward probabilistic classifier, the naive Bayes classifier is based on the application of the Bayes theorem with strong independence assumptions between the features. The scalability of the Naive Bayes classifier is excellent [29]. In proportion to the number of variables in the learning issue, numerous parameters are needed. Simple Bayes and independent Bayes are additional names for naive Bayes models. Given the class variable, Naive Bayes assumes that a feature's value is independent of all other features [29]. Despite any potential relationships between the features, a Naive Bayes classifier considers each feature to contribute independently to the likelihood.

To predict the class label of a new data instance, Naive Bayes calculates the posterior probability $P(C|X_1, X_2, \dots, X_n)$ for each class C and selects the class with the highest probability. In this study, using Bayes' theorem and the naive assumption, the posterior probability can be calculated using equation:

$$P(C|X_1, X_2, \dots, X_n) \propto P(C) \times \prod_{i=1}^n P(X_i|C), \quad (8)$$

where: $P(C)$ is the prior probability of class C , $P(X_i|C)$ is the likelihood of observing feature X_i given class C , \propto denotes proportionality, indicating that the probabilities are scaled to sum up to 1.

3.8. Performance Measures. In this work, the effectiveness of a categorization strategy was summarized using a confusion matrix. When there are more than two classes in a dataset or when there are not an equal number of observations in each class, classification accuracy might be deceptive. Confusion matrix calculation provides a clearer picture of the classification model's successes and shortcomings. Performance is measured based on precision, recall, and accuracy. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

The easiest performance metric to understand is accuracy, which is just the proportion of properly predicted observations to all observations. One can believe that our model is the best if it has a high level of accuracy [30]. Yes, accuracy is an excellent indicator, but only when the values of the false positive and false negative rates are nearly equal in the datasets. Therefore, one has to look at other parameters to evaluate the model's performance [30]. The recall is the ratio of correctly predicted positive observations to all observations in the actual class [31]. In this study, performance is measured based on the following parameters, as shown in equations (9)–(11).

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (9)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (10)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (11)$$

where TP = True Positive, TN = True Negative, FP = False Positive, and FN = False Negative.

Additionally, we employed Cohen's kappa statistic, which is an excellent indicator that effectively handles both multi-class and imbalanced-class issues. We occasionally encounter a multi-class classification challenge in machine learning. Because of this, we employ these measurements.

Cohen's kappa is defined as [32]:

$$k = \frac{p_o - p_e}{1 - p_e}, \quad (12)$$

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

4. Results and Discussion

4.1. Environment Setup. This research used a machine with an 11th Gen Intel Core i7-1165G7 processor, 8.00 GB RAM, and a 64 bit operating system. Special tools and programs were used to conduct the experimentation, including Anaconda IDE for Python 3, Jupyter Notebook for data visualization, and Microsoft Excel for data handling. Python was chosen due to its easy-to-learn syntax and the availability of libraries like Numpy, Pandas, Scikit-learn, and Sklearn. Numpy calculates mean values; Pandas fetches data from files, creates data frames, and handles data frames. Scikit-learn, also known as Sklearn, contains machine learning tools like classification, regression, clustering, dimensionality reduction, model selection, and preprocessing. The study used, data preprocessing model selection and used popular machine learning algorithms like Naive Bayes, Decision Trees, KNN, and SVM.

4.2. Data Preprocessing. This study employed the dataset from Wollo University, Kombolcha Institute of Technology, which includes information on students from 2017 to 2022. After removing missing data, the data was pre-processed using Python software. The original dataset had numerous missing values and duplicated records, requiring data preprocessing to ensure the quality of input data for the predictive model. So, the dataset underwent rigorous preprocessing to address issues such as missing values and duplicated records. Python software was utilized for data preprocessing to ensure the quality of input data for subsequent predictive modeling tasks.

4.3. Data Visualization. Various visualizations were generated to provide insights into different aspects of the dataset. Figures such as entrance exam distribution along with final result, distribution of final result classes, regional distribution, disability frequency, and gender distribution among students were presented to aid in understanding the dataset's characteristics and patterns.

The outcome of a student's academic performance or achievement at the end of a certain period, typically an academic year has been used as the final result. It encompasses the overall assessment of the student's progress, including factors such as grades, credits earned, and any additional distinctions or qualifications attained. In the context of this study, the final result likely indicates the culmination of a student's academic endeavors within the specified time frame, providing a comprehensive measure of their performance and success. As Figure 4 indicates the equivalent result for the minimum requirement to join the university is dominant over others. However, the lower level also has a high number of occurrences. The researcher analyses the data to magnify a sense of what additional work should be performed to quantify and extract insights from the data. The distribution of the final result class is presented in Figure 5.

Figure 6 shows the distribution of regions by final result. As it indicates, the southwestern region has less count compared to the others. However, the southern region and

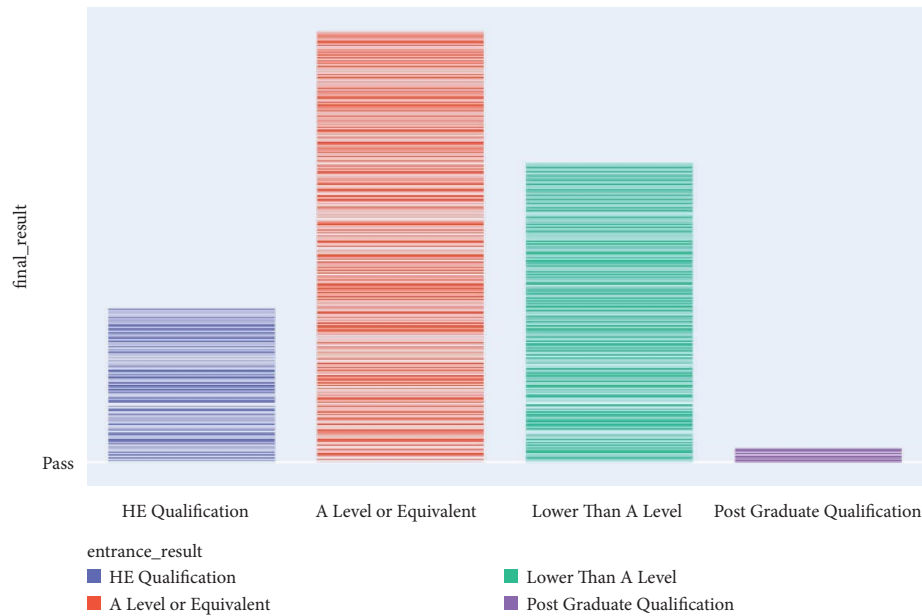


FIGURE 4: Entrance exam distribution along with the final result.

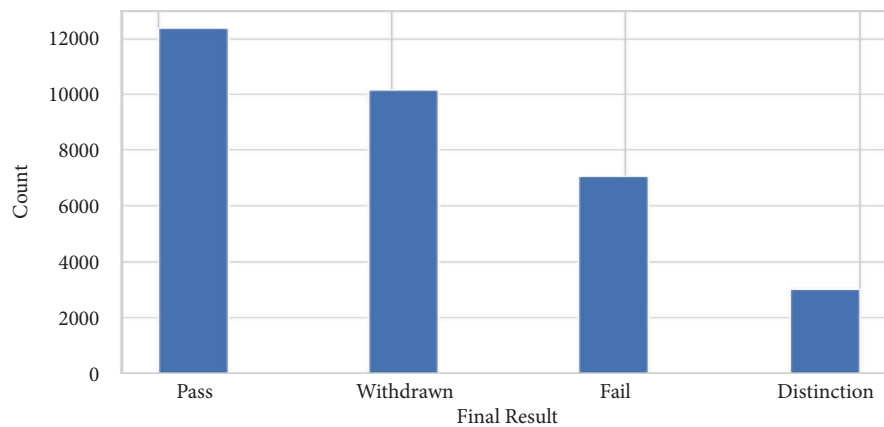


FIGURE 5: Final result class distribution.

Sidamo region have the highest count. Researchers used this distribution to analyze and magnify the highest final result of the students in their home region.

The distribution of the students with a disability concerning the number of attempts to pass the admission exam is inspected in Figure 7. After analysis and feature extraction, the column disability has been dropped because this is insignificant in the scope of the study. Therefore, the proposed model considers only the significant features based on the feature selection outputs.

4.4. Cluster Analysis. Clusters were identified based on final result classes, allowing for a deeper understanding of the distribution of student performance across different categories. This analysis enabled the identification of distinct clusters and their characteristics, aiding in targeted interventions and support strategies. Based on the final results, we can classify clusters as grades A, grades B, and grades C, as shown in Table 3.

The majority of the Grade A students come from the Oromia Region, Addis Ababa, and Amhara Region. Grades C students are mostly found in the South Region, Somalia Region, and Sidamo Region. It is noticed that the Grade B and C students share a common region which is the South Region, Somalia Region, and Sidamo Region. The connection between the regions of Grade B and C students can be understood by analyzing cluster C2. Table 4 shows the region features distribution.

In cluster C2 as shown in Table 5, the majority of the males have clustered but when other clusters are taken into account it is inferred that more females are scored Grade A when compared to males.

Grade A and B mostly have their entrance results as “A Level or Equivalent”. Slightly more than half of the Grade C have their entrance result as “Lower than A level” as shown in Table 6. This infers that students in Grade C at this educational level find it possible to understand their courses and hence drop them. It is inferred that as the educational

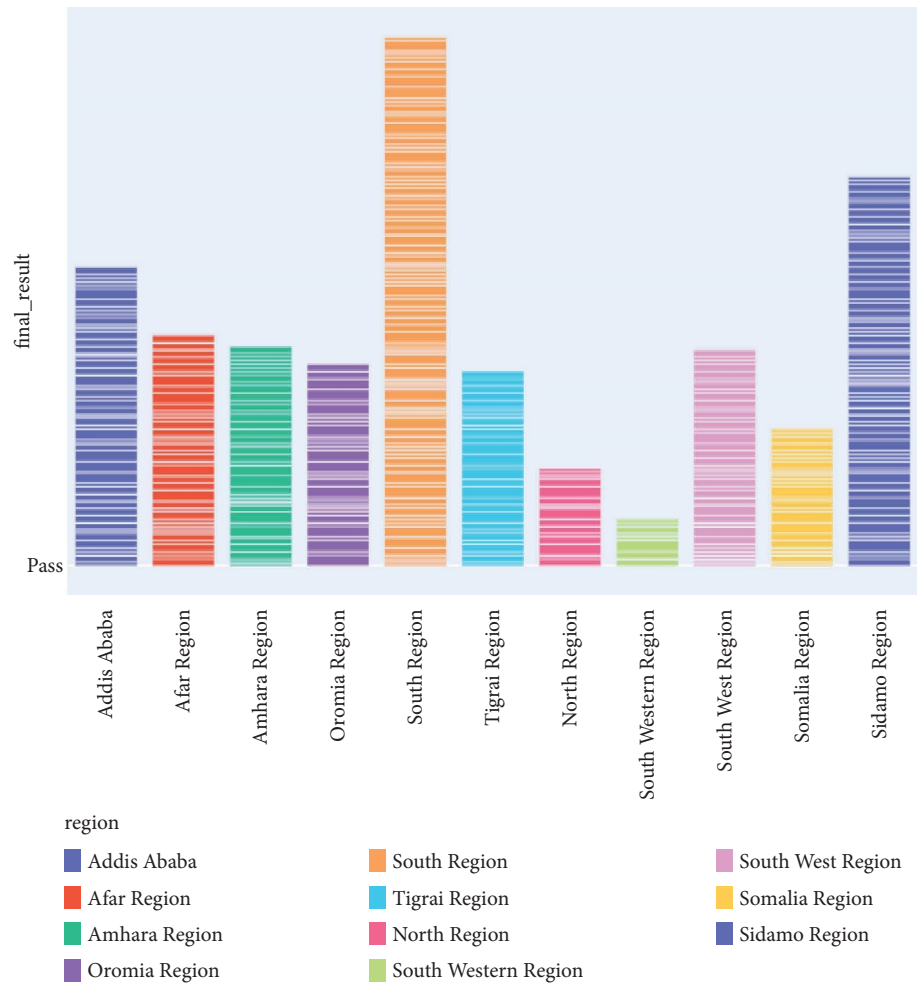


FIGURE 6: The distribution of region with the final result.

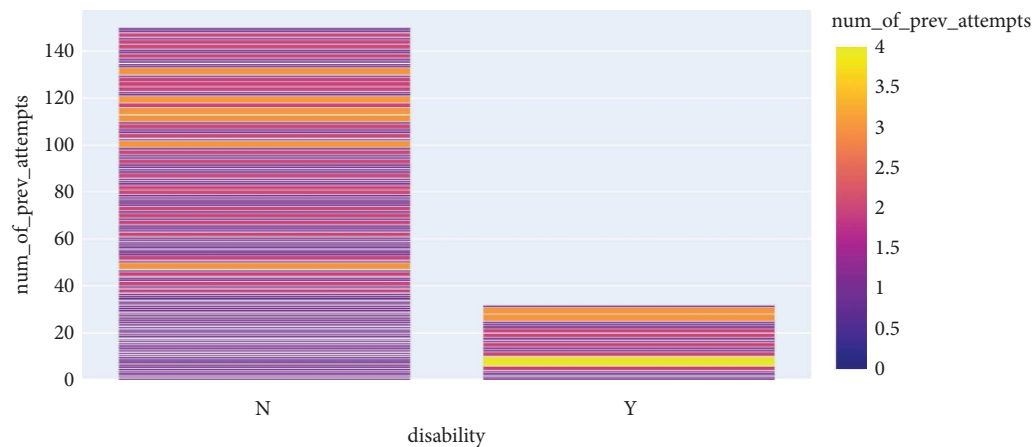


FIGURE 7: Disability frequency with the number of attempts.

level increases people’s understanding of the course also increases, and the dropout rate decreases. This is the same for failed students.

4.5. Algorithms Comparison. A comparative analysis of various machine learning algorithms, including decision trees, Naïve Bayes, support vector machine (SVM), and K-

TABLE 3: Distribution of final result classes.

Feature final result	C0 (grades A) (%)	C1 (grades B) (%)	C2 (grades C)
Pass	0	0	77.34
Fail	0	100	8.54
Distinction	100	0	5.09
Withdrawn	0	0	9.03

TABLE 4: Regional features distribution.

Feature region	C0 (grades A) (%)	C1 (grades B) (%)	C2 (grades C) (%)
Addis Ababa	11.88	10.01	5.90
Amhara Region	11.10	6.84	6.90
Oromia Region	13.40	4.52	2.96
Tigray Region	7.41	12.45	8.87
South Region	3.40	13.54	13.94
South Western Region	4.94	10.13	6.90
Somalia Region	9.26	8.79	9.03
Afar Region	8.73	3.66	7.72
Sidamo Region	7.73	11.74	11.17

TABLE 5: Gender distribution among final result cluster.

Feature gender	C0 (grade A) (%)	C1 (grade B) (%)	C2 (grade C) (%)
Male	40.80	62.09	50.41
Female	59.20	37.91	49.39

TABLE 6: Entrance qualification analysis.

Feature entrance result	C0 (grade A) (%)	C1 (grade B) (%)	C2 (grade C) (%)
No formal quals	0.62	1.71	0
Lower than a level	19.75	37.36	58.36
A level or equivalent	57.72	51.28	32.22

nearest neighbors (KNN), was conducted to evaluate their effectiveness in predicting student outcomes. The performance of each algorithm was assessed based on metrics such as precision, recall, accuracy, and kappa statistics.

Decision trees (DT) are widely used for classification and prediction, including predicting student performance, dropout rates, and final GPA. Naive Bayesian is a popular classification algorithm due to its simplicity, computational efficiency, and high accuracy. In educational settings, Naïve Bayes is used to predict student performance based on previous semester results, achieving the highest accuracy in forecasting graduate students' GPAs. Support Vector Machine is another accurate technique for student performance prediction. Ramesh et al. [33] examined the accuracy of Naïve Bayes Simple, multilayer perceptron, SMO, J48, and REP tree techniques for predicting student performance, finding multilayer perceptron as the most appropriate algorithm, but SMO is a competitive one. In this study, we conduct a comparative study among KNN, SVM, decision tree, and naïve Bayes classifier.

Table 7 shows the outcomes of the prediction models that were employed in this investigation. Table 6 displays the outcomes of prediction algorithms after adjusting the

parameters. As shown by the findings, SVM Linear gave the best prediction results before parameter adjustment, with a 95.4% accuracy rate, followed by decision tree with a 90.9% accuracy rate, and Naive Bayes with a 77.3% accuracy rate. The outcomes of parameter adjustment greatly increased the accuracy of the three prediction systems. SVM Linear's prediction precision increased from 95.4% to 96.0%. Decision tree accuracy increased from 90.9% to 93.4%. The Nave Bayes model's prediction accuracy increased the greatest, from 77.3% to 83.3%.

The prediction accuracy of Naïve Bayes is the lowest when compared to other prediction methods. This can be attributed to Naïve Bayes assuming the strong independent relationship between the features.

The findings of this study provide a comprehensive understanding of student performance prediction in higher education. By employing rigorous data preprocessing and feature selection techniques, the study establishes a robust predictive model, ensuring the reliability of subsequent analyses.

The comparative analysis of machine learning algorithms, including support vector machine (SVM), decision tree, Naïve Bayes, and K-nearest neighbors (KNN), confirms

TABLE 7: Prediction results of all methods before parameter tuning.

Prediction method	Precision	Recall	Accuracy	Kappa statistic
SVM linear	0.9402	0.9789	0.9541	0.9305368
Naïve bayes	0.8186	0.8943	0.7738	0.6545808
Decision tree	0.8890	0.8784	0.9099	0.8632813
KNN	0.8441	0.8584	0.8538	0.8232813

their effectiveness in predicting student outcomes. These findings align with existing literature, validating the versatility and accuracy of these classifiers in educational settings.

In the study, we employed grid search, a method used to train and assess models with various values of k in the k -nearest neighbors (kNN) algorithm. Following the application of 10-fold cross-validation, we determined the optimal value of k to be 8, as it yielded the best performance metrics.

The study uncovers patterns in student performance across regions and demographic groups, highlighting disparities and intervention opportunities. The lower prediction accuracy of Naïve Bayes (83.3%) compared to SVM (96.0%) and decision tree (93.4%) can be attributed to its strong independence assumption, sensitivity to feature correlations, and limited model flexibility. The detailed result has been presented in Table 8.

SVM's superior performance (96.0% accuracy) stems from its margin maximization, ability to handle nonlinear relationships, and robustness to overfitting. Decision trees (93.4% accuracy) excel in interpretability, handling nonlinear relationships, and identifying feature importance, making them valuable predictors. In the study, the linear kernel was selected for the Support Vector Machine (SVM). This decision was based on several factors: the linear kernel's simplicity and interpretability, its computational efficiency, and its ability to perform well with high-dimensional data or when the data is linearly separable. These qualities make the linear kernel a suitable choice for analyzing and interpreting the decision boundary and the relationship between features and the target variable in SVM classification.

Moreover, using hyperparameter tuning with these algorithms has an improvement in model performance compared with the existing methods as shown in Table 6. Predicting a student's performance could be helpful in various contexts related to the university-level learning process. Numerous papers have been produced that analyze distinct characteristics or aspects crucial to comprehending and enhancing pupils' academic achievement. This study has developed a model that, with the aid of historical student records, can assist students in improving their exam performance by foretelling student achievement. Therefore, it is clear that the issue is one of classification, and the suggested model assigns a student to a category depending on the information provided. The methodology used affects data mining success. To lessen sample-related bias in our investigation, we used repeated k -fold cross-validation. One of the causes of accurate prediction outcomes is this. The accuracy of the prediction models was then further increased by parameter tuning or hyperparameter optimization. The results showed an increase in accuracy after parameter

TABLE 8: Prediction results of all methods after parameter tuning.

Prediction method	Precision	Recall	Accuracy	Kappa statistic
SVM linear	0.9418	0.9843	0.9603	0.9398497
Naïve bayes	0.8918	0.8943	0.8332	0.7428229
Decision tree	0.9018	0.9043	0.9341	0.9004462
KNN	0.8941	0.8984	0.8738	0.7738735

adjustment. Additionally, researchers have looked into the functions of K -nearest Neighbor, Naive Bayes, Decision trees, and Support Vector Machine classifiers. Using the dataset, we develop models, after which we assess the student's performance. The findings indicate that the decision tree is the second-best predictor, with 93.4% accuracy, and the support vector machine is the best, with 96.0% accuracy. The accuracy of Nave Bayes is the lowest at 83.3%. Although the constructed model can offer accurate predictions, there is still much work to be done to incorporate these proposed methods into other predictive algorithms to generate a better performance and user experience.

5. Conclusions

The methodology used affects data mining success. To lessen sample-related bias in our investigation, we used repeated k -fold cross-validation. One of the causes of accurate prediction outcomes is this. The accuracy of the prediction models was then further increased by parameter tuning or hyperparameter optimization. The results showed an increase in accuracy both before and after parameter adjustment. This study demonstrated how data mining tools may forecast students' grades when used with a solid methodology. This study explores the effectiveness of machine learning algorithms in predicting student outcomes in higher education. The results show that Support Vector Machine (SVM), decision tree, Naïve Bayes, and K -nearest neighbors (KNN) classifiers are more versatile and accurate than Naïve Bayes (83.3%). Naïve Bayes' lower prediction accuracy can be attributed to several factors, including its strong independence assumption, sensitivity to feature correlations, limited expressiveness, imbalanced-class distribution, and lack of model flexibility. SVM achieved the highest accuracy of 96.0% compared to other classifiers like Decision Tree, Naïve Bayes, and KNN. SVM's superior performance is due to margin maximization, nonlinear separability, robustness to overfitting, handling high-dimensional data, effective kernel functions, and fewer hyperparameters. Decision trees, on the other hand, achieved the second-highest accuracy of 93.4% among the classifiers evaluated in the study. Decision trees provide a transparent and interpretable model, making it easier for users to understand the decision-making process. They can handle nonlinear relationships by recursively partitioning the feature space into subsets based on feature thresholds. They also rank features based on their importance in the classification process, identifying key predictors of the target variable and providing valuable insights into the underlying data distribution. Decision trees are robust to irrelevant features or noisy data because they selectively choose

features that contribute to improving classification accuracy. Their scalability allows them to handle large volumes of data efficiently while maintaining high predictive accuracy. The outcomes of parameter adjustment greatly increased the accuracy of the three prediction systems. SVM Linear's prediction precision increased from 95.4% to 96.0%. Decision tree accuracy increased from 90.9% to 93.4%. The Nave Bayes model's prediction accuracy increased the greatest, from 77.3% to 83.3%.

The study uses advanced machine learning algorithms to predict student performance, enhancing accuracy and enabling early intervention. It also allows for personalized interventions based on individual needs, optimizing resource allocation. The study provides insights into the effectiveness of different ML algorithms, enabling informed decision-making for educators and policymakers. It also emphasizes continuous improvement through longitudinal studies and stakeholder feedback, ensuring the models remain relevant and effective in addressing evolving challenges in education and student support. However, it has limitations, including a small sample size single institution focus, and parameter tuning sensitivity. Future research should focus on larger, more diverse datasets, longitudinal analysis, incorporating additional variables, improving model interpretability, and external validation. These could enhance the robustness and generalizability of predictive models, provide deeper insights into performance factors, and improve transparency and trust in predictive models. By addressing these limitations and pursuing future directions, researchers can contribute to the development of more accurate and actionable predictive models for improving student outcomes.

Data Availability

The data used to support the findings of this study are included within the supplementary information file(s).

Disclosure

The manuscript has been presented with the reference number [4433087] on https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4433087 as preprinted to work together on theories and findings and received comments from the scholars [34].

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

I would like to take this opportunity to express my heartfelt gratitude to the Wollo University ICT teams for their assistance with data collection.

Supplementary Materials

The dataset, originating from Wollo University's Kombolcha Institute of Technology, encompasses student data from 2017 to 2022. It contains 32,582 records with eight columns:

student ID, gender, region, entrance result, number of previous attempts, studied credits, disability status, and final result. This dataset facilitates demographic studies, academic trend analysis, and identification of factors influencing student outcomes. (*Supplementary Materials*)

References

- [1] Y. Baashar, G. Alkaws, N. Ali, H. Alhussian, and H. T. Bahboub, "Predicting student's performance using machine learning methods: a systematic literature review," in *Proceedings of the 2021 International Conference on Computer and Information Sciences (ICCOINS)*, pp. 357–362, Kuching, Malaysia, June 2021.
- [2] S. K. Yadav and S. Pal, "Data mining: a prediction for performance improvement of engineering students using classification," 2012, <https://arxiv.org/abs/1203.3832>.
- [3] M. Liu and D. Yu, "Towards intelligent E-learning systems," *Education and Information Technologies*, vol. 28, no. 7, pp. 7845–7876, 2023.
- [4] T. M. Mitchell, "The discipline of machine learning," *Machine Learning*, vol. 9, 2006.
- [5] O. Fy, A. Jet, O. Awodele, J. O. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: classification and comparison," *International Journal of Computer Trends and Technology*, vol. 48, no. 3, pp. 128–138, 2017.
- [6] N. Delavari, S. Phon-Amnuaisuk, and M. R. Beikzadeh, "Data mining application in higher learning institutions," *Informatics in Education*, vol. 7, no. 1, pp. 31–54, 2008.
- [7] S. Nunn, J. T. Avella, T. Kanai, and M. Kebritchi, "Learning analytics methods, benefits, and challenges in higher education: a systematic literature review," *Online Learning*, vol. 20, no. 2, pp. 13–29, 2016.
- [8] S. Bharara, S. Sabitha, and A. Bansal, "Application of learning analytics using clustering data Mining for Students. disposition analysis," *Education and Information Technologies*, vol. 23, no. 2, pp. 957–984, 2018.
- [9] B. K. Baradwaj and S. Pal, "Mining educational data to analyze students' performance," 2012, <https://arxiv.org/pdf/1201.3417.pdf>.
- [10] K. Aulakh, R. K. Roul, and M. Kaushal, "E-learning enhancement through Educational Data Mining with Covid-19 outbreak period in backdrop: a review," *International Journal of Educational Development*, vol. 101, Article ID 102814, 2023.
- [11] J. M. Helm, A. M. Swiergosz, H. S. Haeberle et al., "Machine learning and artificial intelligence: definitions, applications, and future directions," *Curr. Rev. Musculoskelet. Med.*, vol. 13, no. 1, pp. 69–76, 2020.
- [12] C. K. Suryadevara, "Predictive modeling for student performance: harnessing machine learning to forecast academic marks," *International Journal of Applied Science and Engineering*, vol. 8, no. 12, 2018.
- [13] S. Talwar, M. Talwar, V. Tarjanne, and A. Dhir, "Why retail investors trade equity during the pandemic? An application of artificial neural networks to examine behavioral biases," *Psychology and Marketing*, vol. 38, no. 11, pp. 2142–2163, 2021.
- [14] S. Kotsiantis, K. Patriarcheas, and M. Xenos, "A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education," *Knowledge-Based Systems*, vol. 23, no. 6, pp. 529–535, 2010.
- [15] B. A. Sani and H. Badamasi, "Machine learning algorithms to predict student's academic performance," *Bakolori Journal of General Studies*, vol. 12, no. 2, pp. 3656–3671, 2021.

- [16] Y. A. Alsariera, Y. Baashar, G. Alkawsi, A. Mustafa, A. A. Alkahtani, and N. Ali, "Assessment and evaluation of different machine learning algorithms for predicting student performance," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–11, 2022.
- [17] H. M. R. Hasan, A. K. M. S. A. Rabby, M. T. Islam, and S. A. Hossain, "Machine learning algorithm for student's performance prediction," in *Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–7, Kanpur, India, July 2019.
- [18] E. S. Bhutto, I. F. Siddiqui, Q. A. Arain, and M. Anwar, "Predicting students' academic performance through supervised machine learning," in *Proceedings of the 2020 International Conference on Information Science and Communication Technology (ICISCT)*, pp. 1–6, Karachi, Pakistan, April 2020.
- [19] L. H. Alamri, R. S. Almuslim, M. S. Alotibi, D. K. Alkadi, I. Ullah Khan, and N. Aslam, "Predicting student academic performance using support vector machine and random forest," in *Proceedings of the 2020 3rd International Conference on Education Technology Management*, pp. 100–107, London, UK, June 2020.
- [20] B. Albreiki, N. Zaki, and H. Alashwal, "A systematic literature review of student performance prediction using machine learning techniques," *Education Sciences*, vol. 11, no. 9, p. 552, 2021.
- [21] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, 2021.
- [22] I. H. Sarker, "Machine learning: algorithms, real-world applications, and research directions," *SN Computer Science*, vol. 2, no. 3, pp. 160–221, 2021.
- [23] M. Cui and others, "Introduction to the k-means clustering algorithm based on the elbow method," *Auditing in Accounting*, vol. 1, no. 1, pp. 5–8, 2020.
- [24] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in Neural Information Processing Systems*, vol. 24, 2011.
- [25] A. Mucherino, P. J. Papajorgji, P. M. Pardalos, A. Mucherino, P. J. Papajorgji, and P. M. Pardalos, "K-nearest neighbor classification," *Data Mining in Agriculture*, pp. 83–106, 2009.
- [26] E. Ahmed and A. Letta, "Book recommendation using collaborative filtering algorithm," *Applied Computational Intelligence and Soft Computing*, vol. 2023, Article ID 1514801, 12 pages, 2023.
- [27] J. H. Min and Y.-C. Lee, "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters," *Expert Systems with Applications*, vol. 28, no. 4, pp. 603–614, 2005.
- [28] P. K. Mall, R. K. Yadav, A. K. Rai, V. Narayan, and S. Srivastava, "Early warning signs of Parkinson's disease prediction using machine learning technique," *Journal of Pharmaceutical Negative Results*, vol. 15, pp. 4784–4792, 2022.
- [29] K. M. Al-Aidaroos, A. A. Bakar, and Z. Othman, "Naive Bayes variants in classification learning," in *Proceeding of the 2010 International Conference on Information Retrieval and Knowledge Management (CAMP)*, pp. 276–281, Selangor, Malaysia, June 2010.
- [30] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: an overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 2000.
- [31] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," 2020, <https://arxiv.org/abs/2010.16061>.
- [32] T. Byrt, J. Bishop, and J. B. Carlin, "Bias, prevalence and kappa," *Journal of Clinical Epidemiology*, vol. 46, no. 5, pp. 423–429, 1993.
- [33] V. Ramesh, P. Parkavi, and K. Ramar, "Predicting student performance: a statistical and data mining approach," *International Journal of Computer Applications*, vol. 63, no. 8, 2013.
- [34] E. Ahmed, "Analysing and Predicting Student Performance," 2023, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4433087.