Draft Manuscript for Review

"Evaluating Student Performance Prediction Using Machine Learning Models"

Journal:	Port-Said Engineering Research Journal
Manuscript Title:	Evaluating Student Performance Prediction Using Machine Learning Models

Evaluating Student Performance Prediction Using Machine Learning Models

Abstract

Machine learning plays a crucial role in addressing various challenges in data science. A widely used application of machine learning is the prediction of outcomes based on large educational datasets. This study examines a dataset of 4,424 students with 20 features. Several regression models, including Linear Regression (LR), XGBoost, Support Vector Regression (SVR), Random Forest (RF), and Stacking Regressor, were developed and compared to predict students' GPA on a 0–4 scale. Additionally, classification models such as LR, RF, XGBoost, and Support Vector Machine (SVM) were implemented to categorize students into Dropout, Enrolled, or Graduate groups. Various evaluation metrics such as accuracy, specificity, precision, recall, and F1 score are utilized to assess model performance. Furthermore, a clustering is implemented using the Principal Component Analysis (PCA) on the numerical features algorithm and Uniform Manifold Approximation and Projection (UMAP) for dimensionality reduction on high-dimensional categorical data. Students were segmented into three groups based on the Silhouette Score and Davies-Bouldin Index (DB). The clustering technique identifies three student clusters, yielding a silhouette score 0.35. The proposed system demonstrates strong predictive capabilities as the most effective model, achieving minimal Mean Squared Error (MSE) and high accuracy. These clusters are analyzed through visualizations of exam score distributions and feature averages.

Keywords: Student performance, Classification, Regression, Cluster, Machine Learning

1. Introduction

In scholarly organizations, students' academic performance measures their achievements across various subjects, aiming for academic excellence. Evaluating student performance helps teachers identify weaker students and provide appropriate guidance to enhance their learning outcomes. As a result, this process positively impacts students' academic backgrounds and fosters their overall growth. The prediction of academic performance has been increasingly supported by machine learning techniques [1-5]. However, performance prediction remains challenging due to imbalanced datasets [6]. Educational Data Mining (EDM) has gained significant attention in research, particularly in predicting student outcomes [7]. It has become a powerful tool in supporting improvements in the learning environment. In addition, Wickramasinghe et al. [8] implemented the Dynamic Indicators of Basic Early Literacy Skills (DIBELS) score as a metric to assess the development of early-stage literacy skills in K-6 students. Learning Analytics Intervention (LAI) is used in education to enhance student learning and academic performance [9]. The LAI predicts student performance and identifies at-risk students, allowing effective interventions to improve educational outcomes such as pass rates, retention, and grades. On the other hand, Alalawi et al. [10] integrated data mining and machine learning techniques to develop models using a large dataset to aid in decision-making. The proposed models are designed to predict students at risk of dropping out and identify those at risk of failing. There are several studies in predicting student performance using machine learning techniques to analyzing large datasets including the Naïve Bayes (NB) [11], Support Vector Machine (SVM) [12],

Decision Tree (DT) [13], Random Forest (RF) [14], k-Nearest Neighbours (KNN) [15] and Linear Regression (LR) [16]. Usman et al. [11] employed the NB algorithm to demonstrate the significance of feature subset selection in classifying student performance. A wrapper-based approach was utilized to improve the accuracy of the prediction algorithm. Similarly, Zaffar et al. [12] investigated correlation-based filtering as a feature selection algorithm to identify the factors influencing students' academic performance. The results demonstrated that the proposed technique enhanced the performance of the SVM model, as measured by the F-measure. Lately, Ajibade et al. [13] employed a DT classifier to examine student behavior regarding their interaction with e-learning platforms. Subsequently, Chen et al. [14] applied information gain and Laplacian score for feature ranking to identify the most relevant indicators and integrated RF with a genetic algorithm for effective multi-class classification. Other researchers have combined multiple classifiers such as RF, DT, SVM, and LR to enhance prediction performance [17-20]. Khairy et al. [17] predicted student academic performance with six features: year, midterm, practical exam, written exam, final total degree, and grade. RF, DT, NB, NN, and KNN were employed to predict students' performance. These models achieved high accuracy, ranging from 96% to 98%. Recommendation systems are particularly valuable in assisting students with selecting appropriate courses that align with their abilities and interests [18]. Ensemble learning is a machine learning approach in which multiple learning algorithms are combined to solve the same problem, rather than relying on a single model. As a result, it can be considered a multilevel prediction and classification model, often leading to improved accuracy and robustness in predictive tasks [19]. In the same context, Hussain et al. [20] proposed a new model to predict the marks and grades of students at the secondary level based on DT regression and classification. It utilized regression to predict marks and DT to predict grades. The dataset used is from the Board of Intermediate & Secondary Education (BISE). Predictive analysis applied to estimating students' achievement encompasses various machine learning models, such as regression, classification, and cluster analysis. While most recent studies have focused on classification or regression techniques, not all have limited themselves to these approaches. The integration of regression, classification, and clustering techniques for predicting student performance aims to improve accuracy and effectiveness. The motivations and contributions of this paper can be summarized as follows:

1. **Student Outcome Classification:**

Multiple classifiers, including SVM, RF, LR, Gradient Boosting algorithms (GBA), and eXtreme Gradient Boosting (XGBoost), were applied to categorize students into dropout, enrolled, and graduate groups. The models were evaluated using accuracy, precision, recall, and F1-score to inform data-driven student retention strategies.

1. **GPA Prediction:**

Several regression models, such as LR, XGBoost, Support Vector Regression (SVR), RF, and Stacking Regressor, were developed and compared to predict students' GPAs on a 0–4 scale. Model performance was assessed using Mean Squared Error (MSE) and the R^2 score to identify the most effective predictors of academic performance.

1. **Student Segmentation via Clustering:**

Unsupervised learning algorithms, including K-Means, Gaussian Mixture Models (GMM), Agglomerative Clustering, Spectral Clustering, and Density-Based Spatial Clustering of Applications with Noise (DBSCAN), were used to segment students into performance tiers (low, medium, high). The clustering results were analyzed with the Silhouette Score and DB Index to uncover latent patterns for personalized academic support.

1.2 Organization of the paper

The rest of this paper is structured as follows: Section 2 discusses related works. Section 3 provides a detailed explanation of the methodology and proposed model. Section 4 presents the experiments and results, while Section 5 concludes the paper.

2. Related Works

Advanced machine learning algorithms are essential for analyzing and predicting student academic performance. Accordingly, researchers need to investigate practical tools for modeling and assessing student performance while identifying areas of weakness to improve educational outcomes. Alsariera et al. [4] examined existing machine learning algorithms and key predictive features for student performance. They found six notable machine learning models: DT, Artificial Neural Networks (ANNs), SVM, KNN, LR, and NB. The analysis identified that the most used input features included academic, demographic, internal assessment, and family or personal attributes. The experimental results confirmed that ANNs were the most effective model, highlighting the importance of academic and individual factors in accurately predicting student performance. Consequently, Wickramasinghe et al. [8] measured DIBELS scores to evaluate literacy skills and students' academic performance. The study analyzed data from 185 first and second graders to identify distinct predictors of academic success. For first graders, sleep duration ($\beta = 41.89$) and gender ($\beta = -37.13$) significantly influence performance.

In contrast, for second graders, body mass index (BMI) ($\beta = -4.00$) and reading time ($\beta = 29.14$) were the key predictors. The study employed two machine learning techniques: NB and DT. Using end-of-year DIBELS scores as the target variable, the proposed machine learning approach demonstrated superior performance, achieving a sensitivity of 92% and a specificity of 100%, outperforming traditional models. In the same context, Alalawi et al. [9] proposed Student Performance Predictive Analytics (SPPA), a framework that leverages students' assessment marks, which are typically available to educators, to build predictive models. SPPA utilized machine learning algorithms to develop these models, allowing for the early identification of at-risk students, thereby facilitating timely and personalized interventions aimed at improving student outcomes. Furthermore, SPPA applies practical teaching principles when designing its courses. It also uses strategic interventions to detect students' learning gaps and provide appropriate support.

Ensemble methods are utilized in [13] to improve the performance of classifiers such as NB, KNN, and SVM. The XAPI dataset, which is divided into three categories demographic, academic, and behavioral characteristics requires discretization. This process involves converting numeric attributes into nominal ones. Additionally, resampling is applied to generate a random subsample of the dataset, using either sampling with or without replacement. Meanwhile, Chen et al. [14] explored multi-class classification by combining RF and genetic algorithms to identify indicators related to academic

performance. They utilized a Sequential Forward Selection (SFS) strategy to determine the optimal features. Their analysis found that 16 out of 30 features had a selection rate greater than 0.5, leading to the use of these 16 features in the prediction of academic performance. On the other hand, Sagala et al. [16] developed a predictive model using the LR algorithm, based on the computer science faculty of a private university in Indonesia. This dataset comprised student records collected between 2010 and 2020. Various preprocessing steps were implemented, including data integration, aggregation, and feature encoding, as well as the Synthetic Minority Over-Sampling Technique (SMOTE) to address class imbalance issues. Among the 20 variables examined, 10 were found to be statistically significant in predicting student academic outcomes.

Khairy et al. [17] proposed a method to predict students' exam performance by analyzing the transformation of input data into output information related to student statistics. The dataset contains 830 instances, each comprising six features: year, midterm, practical exam, written exam scores, final total degree, and grade. Several metrics were used to evaluate the algorithms' performance, such as accuracy, precision, recall, F-measure, and the confusion matrix. The results indicated that the proposed RF and DT models achieved a peak accuracy of 98.70%. The Neural Network (NN) model followed with the second-highest accuracy of 96.40%. In the same context, Kord et al. [18] applied machine learning and deep learning to predict students' performance in second-year courses based on their first-year academic results. The MU-dataset contains undergraduate student records spanning twelve years (2008–2020) and includes two types of features: predictive and recommendation-related. The proposed model expects course grades into three categories: low, medium, and high. These predicted grades were utilized within a recommendation system to rank the most suitable courses for each student and to assist in selecting the most appropriate academic department based on individual predicted performance outcomes. Hussain et al. [20] proposed a new model to predict the marks and grades of students at secondary schools to support improvements in academic administration and teaching strategies. It utilized regression to predict marks and DT to predict grades. The dataset is from the Board of Intermediate & Secondary Education (B.I.S.E). The data preprocessing was conducted to enhance data quality. A set of 30 optimal attributes from students' historical academic data was then used to train a regression model for mark prediction and a DT classifier for grade classification. The results indicate that machine learning technologies offer efficient and reliable tools for forecasting academic performance, supporting data-driven educational decision-making.

3. The methodology and proposed model

3.1 Datasets

This dataset included 4,424 higher education students, categorized by their academic status into 2,212 graduates, 1,327 dropouts, and 885 enrolled students. Among the graduates, 52% are male and 48% are female, while among the dropouts, 58% are male and 42% are female. Regarding age distribution, 70% of graduates are between 18 and 22 years, and 20% are between 23 and 30 years, whereas among dropouts, 60% are between 18 and 22 years, and 30% are over 23 years. For this study, the following variables were recorded: gender, age at enrollment, course, admission grade, number of approved curricular units in the first and second semesters, average grades of the units, scholarship status, and tuition fee payment status.

The methodology comprises two principal phases: data preparation and model training. Figure 1 depicts the overall model architecture, presenting a flowchart of the study’s workflow from input data

to generating insights for predicting student exam performance. Initially, raw data is collected from the field, which is then cleaned to eliminate inconsistencies and outliers, ensuring that only relevant information is retained for further analysis. Next, the cleaned data undergoes preprocessing, which includes filtering and applying data scaling techniques. After preprocessing, essential features are selected, and the data is transformed into a format suitable for machine learning algorithms. Subsequently, the dataset is split into training (70%) and testing (30%) subsets for model development and validation. Various predictive models are applied to the training data, and their outputs are evaluated using the testing set. This approach allows for extracting meaningful insights and facilitates knowledge discovery related to student exam performance.

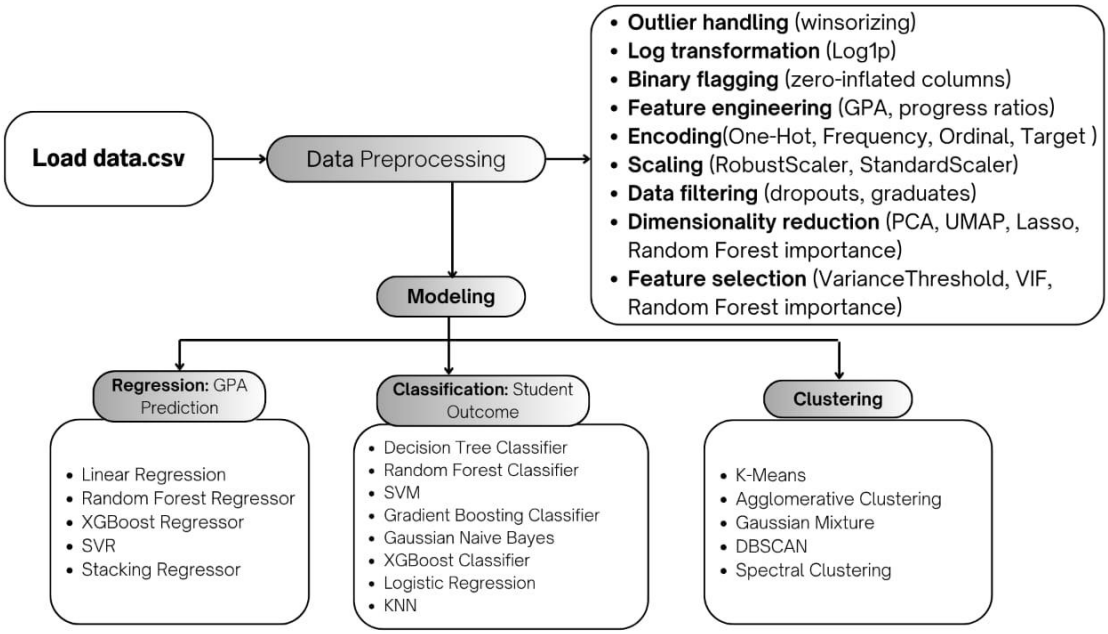


Figure 1 Workflow of the proposed methodology.

3.2 Data Preprocessing

The preprocessing pipeline converted the raw dataset into a format suitable for clustering by tackling high dimensionality, categorical variables, and varying feature scales. This was accomplished through several steps: feature engineering, encoding, scaling, feature selection, and dimensionality reduction.

- **Feature Engineering**

New features have been created from existing variables to better capture nuanced academic patterns and enhance the quality of the analysis. For example, the Academic_Consistency metric measures the stability of student performance across semesters, allowing differentiation between students who consistently perform well and those whose results fluctuate. The Weighted_GPA provides a more accurate academic indicator by placing greater emphasis on the second semester, based on the assumption that it is more difficult and relevant. Additionally, Academic_Efficiency measures how effectively students translate their enrollments into successful completions, offering insights into their level of commitment or potential learning challenges. The Engagement score is calculated by comparing the number of evaluations completed to the number of course enrollments, serving as an indirect measure of student participation and engagement. Together, these engineered features are designed to reveal subtle behavioral patterns essential for meaningful student segmentation and further predictive modeling.

- **Encoding**

Categorical variables were transformed into numerical formats using various encoding techniques. One-hot encoding was applied to nominal features to preserve category independence without introducing any artificial ordinal relationships. For high-cardinality categorical variables, frequency encoding was employed to reduce dimensionality while retaining the statistical influence of each category based on its frequency. Additionally, *Application Order Scaling* was used to transform an ordinal feature into a normalized numerical range, preserving the inherent order while ensuring compatibility with distance-based clustering methods. This encoding strategy was carefully designed to balance interpretability, model performance, and dimensionality control.

- **Scaling**

Numerical features were scaled using different techniques based on their semantic categories to normalize distributions and reduce the influence of outliers. This step is crucial for clustering algorithms such as K-Means, DBSCAN, and Agglomerative Clustering, which rely on distance calculations and are sensitive to the scale of features. Without proper scaling, variables with larger numerical ranges, such as the number of evaluations on a 0–100 scale, could disproportionately influence distance metrics compared to features like GPA, which may range from 0 to 4. Scaling was not applied uniformly across all variables to address this. Instead, each feature group's distribution and functional role were carefully analyzed, and the most appropriate scaling method was selected accordingly.

RobustScaler was applied to GPA-related features because academic grades often contain outliers, such as extremely low or high GPAs. This scaler relies on the interquartile range (IQR), which is less sensitive to such extreme values than StandardScaler or MinMaxScaler. **MinMaxScaler** was used for progress and demographic indicators, including variables like age and the number of enrollments. These features are naturally bounded and fall within known ranges, making MinMaxScaler suitable for preserving relative positioning normalizing the values. **StandardScaler** was applied to engagement and socioeconomic features, which are assumed to follow approximately normal distributions in the student population. This method standardizes features by centering them around the mean of zero and scaling them to unit variance, ensuring that deviations from the average are treated fairly. This is particularly important when average behavior, such as typical engagement levels, is significant for interpretation. Scaling techniques are summarized in Table 1.

- **Feature Selection**

Feature selection improved clustering performance by eliminating variables that contributed little or no meaningful information for distinguishing between student behaviors or academic profiles. All features, such as clustering, influence distance or similarity calculations in unsupervised learning. However, certain features may negatively impact model performance, particularly those that have minimal variance, act as noise, or add computational complexity without enhancing the ability to identify meaningful groupings. To tackle this issue, we applied variance thresholding, a low-variance filter. This method calculated the variance of each feature across the dataset and removed those with extremely low variance, meaning near-constant values across most students. For example, a feature where 98% of students share the same value does not meaningfully contribute to distinguishing between groups; instead, it increases dimensionality without adding value. Although correlation-based feature selection was considered, it was not used in this study. This approach can be sensitive, and sometimes

redundant features can still contribute to clustering by reinforcing patterns. Therefore, using variance thresholding proved to be a simple yet effective method for reducing irrelevant dimensions while preserving the informative structure in the data**.**

Table 1: Scaling techniques applied to numerical features by category

Category	Attribute	Frequency	Scaler Used
Academic Performance	Weighted_GPA; GPA; Curricular units 1st sem (grade); Curricular units 2nd sem (grade); Grade_Improvement;	5	RobustScaler
Academic Progress	Academic_Efficiency; Progress_Ratio_1st_Sem; Progress_Ratio_2nd_Sem; Academic_Consistency;	4	MinMaxScaler
Academic Engagement	Engagement_Score; Curricular units 1st sem (evaluations); Curricular units 2nd sem (evaluations); Total_Academic_Load;	4	StandardScaler
Academic history	Admission grade; Previous qualification (grade); Admission_Strength;	3	RobustScaler
Socioeconomic	GDP; Socioeconomic_Status; Unemployment rate; Inflation rate;	4	StandardScaler
Demographic	Age at enrollment_log; Application order;	2	MinMaxScaler

- **Dimensionality Reduction**

Dimensionality reduction was applied following feature selection to further streamline the dataset by compressing the cleaned set of inputs into a smaller number of informative dimensions. At the same time, feature selection removed low-impact, noisy, or irrelevant variables, and dimensionality reduction aimed to enhance clustering performance and reduce computational complexity. High-dimensional data, particularly when features are sparse or correlated, can obscure the natural structure of the data and negatively affect clustering outcomes. We employed Principal Component Analysis (PCA) on the numerical features to address these issues. The original features are mapped onto uncorrelated dimensions. As a result, we successfully reduced approximately 20 numerical features into 5 to 7 principal components, eliminating redundancy and noise while preserving the key patterns underlying student behavior and academic performance. This dimensionality reduction facilitated more accurate and efficient clustering by enabling algorithms to focus on the most informative aspects of the data.

In addition to applying PCA to numerical features, we employed Uniform Manifold Approximation and Projection (UMAP) to reduce dimensionality on high-dimensional categorical data. UMAP is a non-linear technique well-suited for capturing complex relationships, particularly in sparse binary matrices resulting from one-hot encoding. UMAP effectively preserves local relationships, keeping similar students close, and global structures that separate clusters. We used the Dice distance metric for the categorical feature space, which is optimized for binary vectors and emphasizes shared feature values rather than magnitude. This approach enabled the reduction of the high-dimensional one-hot encoded feature matrix into 3 to 5 UMAP dimensions, significantly compressing the data while retaining critical category-driven patterns.

After separately reducing numerical and categorical features based on PCA and UMAP, we concatenated the resulting components into a unified, compact dataset suitable for clustering. While this merged feature space was more efficient and informative than the original, it could still contain residual noise or complex geometric relationships. We applied a second round of UMAP on the combined PCA+UMAP feature space to further enhance the clustering structure and enable effective visualization. This final step projected the data into a 2D space, ideal for clustering algorithms (e.g., KMeans, DBSCAN) and visual inspection. The resulting low-dimensional embedding preserved student similarity and separation, making identifying behavioral clusters and outliers easier. This final UMAP projection leveraged the strengths of both linear and non-linear dimensionality reduction techniques to uncover the intrinsic structure of the dataset.

3.3 Measuring instruments

This study's development process used the Python programming language. Jupyter Notebook and Google Colab were used mainly as development environments for executing the experiments. The implementation involved libraries such as Pandas for data manipulation, Matplotlib and Seaborn for data visualization, and Scikit-learn for applying machine learning algorithms. Additionally, TensorFlow was incorporated to conduct more advanced model experimentation.

Cluster analysis is used to segment the students into three distinct groups. The clustering results are evaluated using the Silhouette Score [21] and DB Index to uncover latent patterns for personalized academic support. The DB index is calculated as the ratio of total within-cluster dispersion to the separation between clusters. Suppose the cluster C_i and C_j are two clusters, and z_i and z_j are their respective cluster centers. The distance between cluster centers z_i and z_j that corresponding clusters C_i and $C_j$can be written as:

$$S_{ij} = \|z_i - z_j\| \quad (1)$$

where $\|z_i - z_j\|$ represents the Euclidean distance between the two cluster centroids.

The scatter within *i*th cluster is computed as:

$$S_i = \frac{1}{z_i}\sum_{\bar{c}\in C}\|\bar{c} - z_i\| \quad (2)$$

DB index is then defined as:

$$DB = \frac{1}{K}\sum_{i=1}^{K} max_{i\neq j}\frac{S_i+S_j}{S_{ij}} \quad (3)$$

Here K denotes the number of clusters. The objective is to minimize the DB index to achieve better and more appropriate clustering.

Silhouette score is a measurement to evaluate the optimal number of clusters for the dataset. It computes the mean silhouette coefficient of all samples. For each object, the score can range from 1 to -1. The silhouette coefficient for a sample is calculated as the follows:

$$S(i) = \frac{c(i)-d(i)}{\max\{c(i),d(i)\}} \quad (4)$$

where *c* is the mean intra-cluster distance of the *i*th object to all other objects in the same cluster, and d is the mean nearest-cluster distance of the *i*th object to all objects in the other clusters.

A Silhouette score approaching 1 signifies that a point is well matched to its cluster, whereas a score close to 0 suggests that the data lies between clusters and is ambiguously assigned. Negative values indicate possible misclassification. Consequently, the optimal number of clusters is determined by the number that yields the highest mean Silhouette score.

3.4 Machine Learning Models for Classification

Classification is a supervised learning that used algorithms trained on datasets to automatically categorize data and improve performance over time without explicit programming. This study used four machine learning techniques called SVM, RF, GBA, DT, KNN, NB, Logistic regression and XGBoost. Classification used training datasets with labeled input data to build a model. Various evaluation metrics such as accuracy, specificity, precision, recall, and F1 score are utilized to assess model performance.

3.4.1 SVM

An SVM classifier analyzes data by maximizing the margin between the closest positive and negative data points around the decision hyperplane in an N-dimensional space, especially when distinguishing between multiple classes [22]. The effectiveness of SVM can be greatly impaired when the data is not linearly separable. In these instances, a kernel function transforms the input data into a higher-dimensional space, facilitating the identification of the optimal decision boundary. Assumed that the training data consists of pairs (x_i, y_i) for i =1...N, with $x_i \in R^d$ and $y_i \in \{-1,1\}$. The goal is to learn a classifier $f(x)$ such that: $f(x) \geq 0$ for $y_i = 1$ (positive class), $f(x) < 0$ for $y_i = -1$ (negative class). The classification of higher education students is considered an SVM problem, where we assign the feature vector of students $x = \{x_1, x_2, x_3, \ldots . x_N\}$ to a class $y_i \in$ Y or determine that the student does not belong to any class, where Y represents a set of possible classes. A linear classifier has the form as:

$$f^{SVM}(x) = w^T x + b \quad (5)$$

where w is the weight vector, b is the bias the weight vector, and x is an input data.

3.4.2 GBA

GBA develops new weak learners in a sequential manner and merges their predictions, significantly enhancing the model's overall performance. This method not only boosts accuracy but also builds a more robust and reliable predictive system. Assumed that the function $F^{*(GBA)}(x)$ maps instances x to their output values *y* to minimize the expected value of a given loss function L (y, F^{GBA} (x)).

GBA constructs an additive model of the form as [23]:

$$\hat{F}(x) = \sum_{m=1}^{M} \beta_m \, h_m(x) \quad (6)$$

where β is a weight or learning rate, h_m is a base learner, and *m* refers to the iteration number or stage in the boosting process.

At each step m, GBA computes the negative gradient of the loss function $F_m^{GBA}(x)$ with respect to the current model $F_{m-1}^{GBA}(x)$. Then it fits a new base learner $h_m(x)$to these pseudo-residuals and updates the model by adding the new learner, scaled by a weight β_m.
The pseudo-residual can be written as:

$$r_{i,m} = -\left[\frac{\delta \mathrm{L}(\mathrm{y}_i, F^{GBA}(\mathrm{x}_i))}{\delta \mathrm{G}(\mathrm{x}_i)}\right]_{F^{GBA}=F_{m-1}^{GBA}} \quad (7)$$

The model is updated as [23]:

$$F_m^{GBA}(x) = F_{m-1}^{GBA}(x) + \mu_m h_m(x) \quad (8)$$

While traditional GBA minimizes a loss function L (y, F^{GBA} (x)), XGBoost enhances this approach by adding an extra regularization term to the loss function. Additionally, XGBoost is an enhanced and scalable version of GBA, focusing on effectiveness, computational speed, and overall model performance.

The objective function of XGBoost can be expressed as:

$$L_{xgb} = \sum_{i=1}^{N} L(y_i, F^{GBA}(x_i)) + \sum_{m=1}^{M} \vartheta(h_m) \quad (9)$$

where $\vartheta(h_m)$is the regularization term that penalizes complex trees.

$$\vartheta(h_m) = \gamma L + 0.5\tau\|w\|^2 \quad (10)$$

where L is the number of leaves in the tree, *w* is the output scores of the leaves, γ controls the minimum loss reduction gain needed to split an internal node, and τ controls the regularization on the leaf weights.

3.4.3 DT

A DT is a non-parametric supervised learning method that can be used for both classification and regression tasks. This model organizes data into subsets by evaluating the values of input features, resulting in a tree-like structure. Each internal node corresponds to a feature test, each branch represents a decision outcome, and each leaf node indicates a predicted class label (in classification) or a value (in regression) [24].

The prediction $f^{DT}(x)$ assigns a class label based on the majority class in the leaf node and can be written as:

$$f^{DT}(x) = argmax\ P(c|leaf(x)) \quad (11)$$

where $P(c|leaf(x)$ is the proportion of class c in the leaf node reached by input x.

3.4.4 RF

RF is a robust and widely used technique for classification where the model builds multiple decision trees and makes predictions based on the majority vote from all trees. RF operates in three key steps: bootstrapping, feature Selection, and ensemble learning [25]. RF starts by generating multiple subsets of the training data through random sampling with replacement. At each node of every decision tree, only a random subset of features is used to determine the best split. After all the trees have been trained, their predictions are combined. The final prediction is made by taking the majority vote from all the trees.

Assumed that the goal is to find a prediction function $f^{RF}(x)$ for predicting *Y*. The prediction function is:

$$f^{RF}(x) = \frac{1}{P}\sum_{p=1}^{P} T(x) \quad (12)$$

where P denotes the number of regression trees and $T(x)$ is the prediction made by the *p-th* regression tree.

3.4.5 KNN

KNN is a simple, instance-based learning algorithm that classifies a data point based on the class of its nearest neighbors. It identifies the k closest points in the training set using a distance metric like Euclidean distance and assigns the majority class among those neighbors for classification tasks.

The prediction function represented as [26]:

$$f^{KNN}(x) = argmax \sum_{i \in N_k(x)} \prod(y_i = c) \quad (13)$$

where $N_k(x)$ is the set of the k nearest neighbors to x and $\prod(y_i = c)$is an indicator function that returns 1 if $y_i = c$, 0 otherwise.

3.4.6 NB

NB is a probabilistic classifier based on Bayes' theorem, which assumes strong independence between features. It computes the posterior probability for each class given the input features and assigns the class that has the highest probability.

The prediction function can be written as follows [27]:

$$f^{NB}(x) = argmax\ P(c) \prod_{j=1}^{n} P(x_j|c) \quad (14)$$

where $P(c)$ is the prior probability of class c and $P(x_j|c)$is the likelihood of feature x_j given class c.

3.5 Machine Learning Models for Regression

Regression is one of the most effective machine learning techniques used for prediction. It generalizes the classification problem by returning a continuous outcome variable (y) based on the values of one or more independent predictor variables (x). In the field of education, regression models are commonly employed to predict student performance, analyze dropout risks, and optimize learning outcomes. To assess the predictive performance of these models, evaluation metrics such as MSE and the coefficient of determination R^2 are used. Standard machine learning algorithms for building regression models include LR, RF, XGBoost, SVR, and Stacking Regressor.

3.5.1 LR

LR is a deterministic mathematical relationship between two variables x and y, assuming a linear relationship. It can be expressed as [28]:

$$Y = \varphi_1 + \varphi_2 x + \delta \quad (15)$$

where φ_1 and φ_2 are the regression coefficients representing the intercept and slope, respectively. δ is the error term accounting for deviations from the exact linear relationship.

The coefficient of determination R^2 measures the ability of a model to predict an outcome in the LR setting. Specifically, it indicates the proportion of the variance in the dependent variable can be explained by the independent variable in LR. The coefficient of multiple determination R^2can be calculated as:

$$R^2 = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \quad (16)$$

where N represents the number of observations, $\hat{y}_i$ represents the predicted value, $\bar{y}$ represents the mean of the actual value, and y_i represents the actual value.

R^2 ranges between 0 and 1, where 0 means that the model explains nothing, while 1 indicates that all data points lie perfectly on a straight line. However, adjusted R^2 accounts for the number of predictors and can decrease if the new variable does not provide enough explanatory power to justify the additional complexity.

MSE is calculated as the average of the squared differences between the actual and predicted values as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (17)$$

3.5.2 SVR

SVR is an extension of the SVM algorithm, which was initially designed for binary classification tasks. The primary objective of SVR is to identify a function that approximates data points within a specified tolerance, defined by a margin known as the ε-tube [29]. This ε-tube reformulates the regression problem into a convex optimization task, aimed at constructing the flattest possible function that fits the training instances within the ε-insensitive zone. In SVR, the ε-insensitive loss function is minimized, which disregards errors smaller than ε and only penalizes deviations that exceed this margin. Furthermore, SVR focuses exclusively on the data points outside the ε-tube, referred to as the support vectors.

The regression function can be written as [30]:

$$f^{SVR}(x) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)k(x_i, x_i^*) + b \quad (18)$$

where α_i, α_i^* are Lagrange multipliers found during training and $k(x_i, x_i^*)$ is a kernel function which measures the similarity between the new input x_i^* and support vector x_i.

3.5.3 Stacking Regressor

Stacking Regressor is an ensemble learning approach that combines multiple regression models to perform regression tasks more effectively than any single model [31]. This approach consists of two or more level-0 models (base learners) and one level-1 model (meta-learner) that integrates the predictions from the base models. In the level-0 stage, multiple regression models are trained on the dataset, each capturing different patterns and relationships in the data. These models generate predictions that serve as input for the level-1 model. The level-1 model then learns to optimally combine these predictions, resulting in improved overall performance of the ensemble.

3.6 Unsupervised learning models

Clustering is an unsupervised learning process that groups *n* observations into *k* clusters, where $k \leq n$. These groups are commonly referred to as clusters. Several machine learning algorithms are used for clustering in this study, including K-Means, GMM, Agglomerative Clustering, Spectral Clustering, and DBSCAN.

K-Means categorizes data into k clusters by minimizing the total squared distances between data points and their corresponding cluster centroids [32].

GMM is a parametric probabilistic clustering technique representing as a weighted sum of Gaussian component densities. The data points are derived from a mixture of several Gaussian distributions.

GMM can be expressed as [33]:

$$P(x|\beta) = \sum_{i=1}^{K} w_i\, g(x|\mu_i, \sum_i) \quad (19)$$

Where K is the number of components (clusters), w_i is the mixing coefficient for component k, and $g(x|\mu_i, \sum_i)$ represents the Gaussian densities.

Agglomerative Clustering is a hierarchical clustering method that builds nested clusters by merging or splitting the nearest pairs of clusters.

Spectral Clustering utilizes the eigenvalues of similarity matrices to cluster complex data. It performs dimensionality reduction before the clustering process.

DBSCAN is a density-based clustering algorithm that groups closely packed data points and identifies outliers.

3.7 Evaluation of trained ML algorithms

Several standard metrics were employed to evaluate the classification model's performance in predicting student outcomes, including accuracy, recall (sensitivity), precision, F1-score, and G-Mean. These metrics rely on the confusion matrix, which summarizes the number of correct and incorrect predictions made by the model [34].

- **Accuracy** measures the ratio of accurate predictions (including both passed and failed students) to the total number of cases. True Positives (TP) refer to students who passed and were correctly identified as passing, while True Negatives (TN) represent students who failed and were accurately classified. False Positives (FP) indicate students who failed but were mistakenly predicted as passing, whereas False Negatives (FN) denote students who passed but were incorrectly classified as failures.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \times 100 \quad (20)$$

- **Recall** (Sensitivity) indicates the ability of the model to correctly identify students who actually passed.

$$Recall = TP / (TP + FN) \times 100 \quad (21)$$

These metrics are computed over varying decision thresholds to better capture the trade-offs in classifier performance across different sensitivity levels. This helps in optimizing the model to balance the misclassification costs, especially in educational settings where false negatives (misidentifying a successful student as unsuccessful) can be critical.

- **F1-score:** It is a metric that combines both Precision and Recall into a single value and is particularly useful when dealing with imbalanced datasets. It is calculated using the following formula:

$$F1 = 2 \times (Precision \times Recall) / (Precision + Recall) \quad (22)$$

- **Geometric Mean (G-Mean)**:
  This metric is used to measure the balance between Sensitivity (Recall) and Specificity. It reflects the model's ability to distinguish between the two classes (students who passed and those who failed). It is calculated as

$$G - Mean = \sqrt{Sensitivity \times Specificity} \quad (23)$$

where Specificity indicates the model's ability to correctly identify students who failed, and it is calculated as:

$$Specificity = (TN) / (TN + FP) \quad (24)$$

These metrics provide a comprehensive evaluation of the model's performance, especially when there is an imbalance in the number of samples between the classes.

- **Multiclass Classification** includes three groups: dropout, enrolled, and graduate. The confusion matrix becomes an N × N matrix, where N is the number of classes. The above metrics cannot be applied directly. Instead, they are calculated per class and then averaged.
  - **Precision** $PPV(C_i)$ for a class C_i:

$$PPV(C_i) = TP(C_i) / (TP(C_i) + FP(C_i)) \quad (25)$$

  - **Recall** for a class C_i:

$$TPR(C_i) = TP(C_i) / (TP(C_i) + FN(C_i)) \quad (26)$$

4 Experimental Results and Discussion

The results are divided into three sections. The first section presents the findings of our study on classifying student outcomes (Dropout vs. Graduate) using supervised learning models, focusing on the performance of various techniques. The second section focuses on predicting student performance using regression models, focusing on the performance of multiple methods and a summary of key insights. Quantitative metrics and visualizations support these results to provide a comprehensive overview. The final section discusses predicting student performance levels (low, medium, high) using clustering models, focusing on the effectiveness of the applied techniques. Table 2 presents the hyperparameters for the competitive classification, regression, and cluster models.

Table 2 Hyper-parameters for the competitive classification, regression, and cluster models.

Models		Hyper-parameters	Value
Classification	LR	C	0.1
		penalty	l2
	RF	n_estimators	100
		max_depth	10
	SVM	C	1.0
		kernel	rbf
	DT	max_depth	5
		min_samples_split	10
	XGBoost	n_estimators	50
		learning_rate	0.05
	KNN	n_neighbors	5
	GBA	n_estimators	100
		learning_rate	0.05
Regression	RF	max_depth	15
		min_samples_split	2
	XGBoost	learning_rate	0.1
		n_estimators	100
	SVR	C	1
		epsilon	0.01
Clustering	KMeans	n_clusters	3
		Init	'k-means++'
		n_init	25
		max_iter	500
		tol	1e-5

		random_state	42
	GGM	n_components	3
		covariance_type	‘full’
		max_iter	200
		n_init	10
		random_state	43
	Agglomerative	n_clusters	3
		linkage	‘ward’
		metric	'euclidean'
	DBSCAN	eps	0.9498963074037952
		min_samples	44
		metric	'euclidean'
		algorithm	'auto'
		leaf_size	40
	Spectral	n_clusters	3
		assign_labels	'discretize'
		affinity	'nearest_neighbors'
		n_neighbors	15
		random_state	44

4.1 Model Performance Across Classification Techniques

Table 3 summarizes the performance metrics for each model, including accuracy, precision, recall, and F1 Score (all macro-averaged). GBA achieved the highest F1 score of 0.884944, with an accuracy of 0.893939, precision of 0.903899, and recall of 0.874506, closely followed by SVM with an F1 score of 0.884750 and the same accuracy of 0.893939. RF demonstrated strong performance with an F1 score of 0.882955 and an accuracy of 0.892562. Both logistic regression and XGBoost achieved an accuracy of 0.880165. However, XGBoost yielded higher precision (0.896035) but a lower recall (0.855642). In contrast, DT, KNN, and NB showed lower performance, with F1 scores of 0.856901, 0.838633, and 0.813395, respectively. These findings emphasize the effectiveness of ensemble methods, particularly GBA, in classifying student outcomes.

Additionally, Table 4 illustrates the training and testing of the models, providing insight into their overall fit. The RF model achieved the highest training accuracy (0.9349), followed closely by SVM (0.9287) and GBA (0.9239). Both GBA and SVM achieved the highest test accuracy, each scoring 0.8939, which aligns with their previously noted strong F1 Scores. Conversely, NB demonstrated the weakest performance, with the lowest training accuracy (0.8209) and test accuracy (0.8264), indicating that it struggled to generalize well to the training and test data.

Table 3 Classification Model Performance Metrics

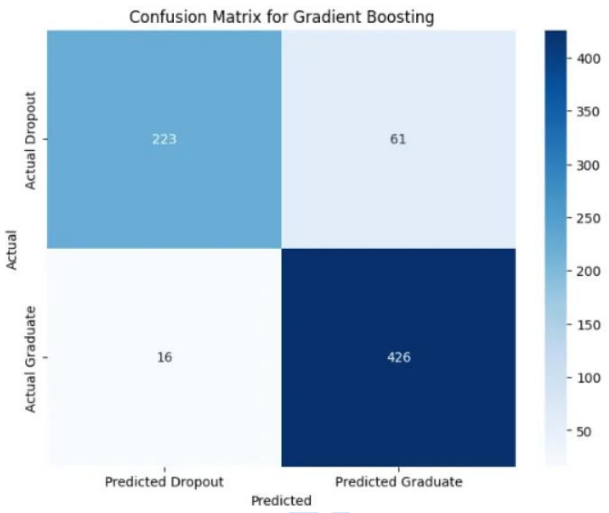
Model	Accuracy	Precision	Recall	F1 Score
GBA	0.893939	0.903899	0.874506	0.884944
SVM	0.893939	0.904960	0.873877	0.884750
RF	0.892562	0.905045	0.871487	0.882955
Logistic Regression	0.880165	0.880197	0.866341	0.872035

XGBoost	0.880165	0.896035	0.855642	0.868409
DT	0.870523	0.889150	0.843318	0.856901
KNN	0.853994	0.869483	0.825967	0.838633
NB	0.826446	0.823966	0.807119	0.813395

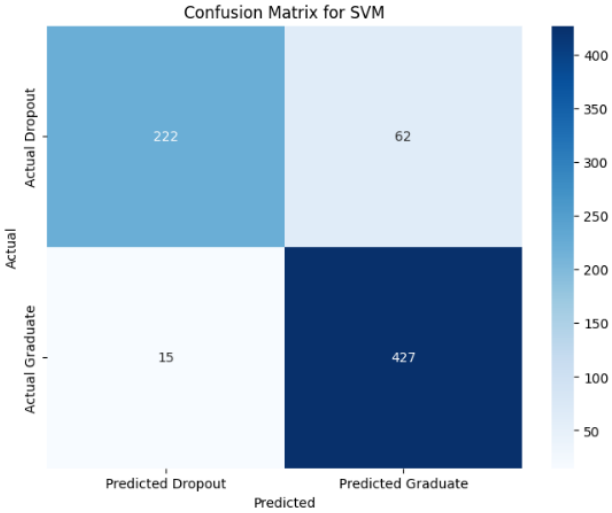
Table 4 Classification Model Training and Test Accuracy

Model	Training Accuracy	Test Accuracy
SVM	0.9287	0.8939
GBA	0.9239	0.8939
RF	0.9349	0.8926
Logistic Regression	0.8919	0.8802
XGBoost	0.8953	0.8802
DT	0.8926	0.8705
KNN	0.8957	0.8540
NB	0.8209	0.8264

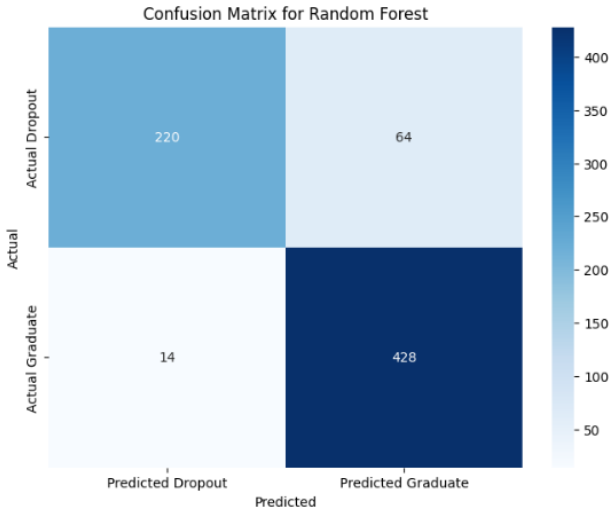
Figures 2 presents the confusion matrices for each model, illustrating the distribution of predictions across Dropout and Graduate classes. The confusion matrix for GBA in Figure 2a shows high true positive rates for both classes, consistent with its top F1 Score. Figure 2b and Figure 2c demonstrate strong performance with minimal misclassifications, aligning with their high F1 scores for SVM and RF, respectively. Figure 2d and Figure 2e show balanced predictions for logistic regression and XGBoost, though with slightly more misclassifications, particularly for the Graduate class in XGBoost, reflecting its lower recall. Figure 2f shows more errors in predicting Graduates for DT, consistent with its lower recall of 0.843318. KNN in Figure 2g and NB in Figure 2h exhibit the highest misclassification rates, particularly for Graduates, which aligns with their lower F1 scores. According to the classification analysis, GBA emerged as the best model based achieving the highest F1 score of 0.884944 and a test accuracy of 0.8939, demonstrating the effectiveness of ensemble methods in classifying student outcomes. In addition, Figure 3 displays the top 20 feature importances, underscoring the key predictors driving the classification, such as age at enrollment and father's occupation. These results offer a robust framework for predicting student success, with potential applications in educational interventions to improve retention rates.



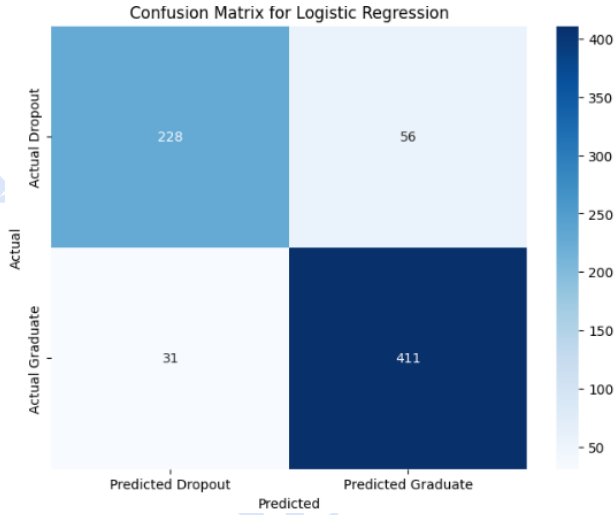
(a) GBA



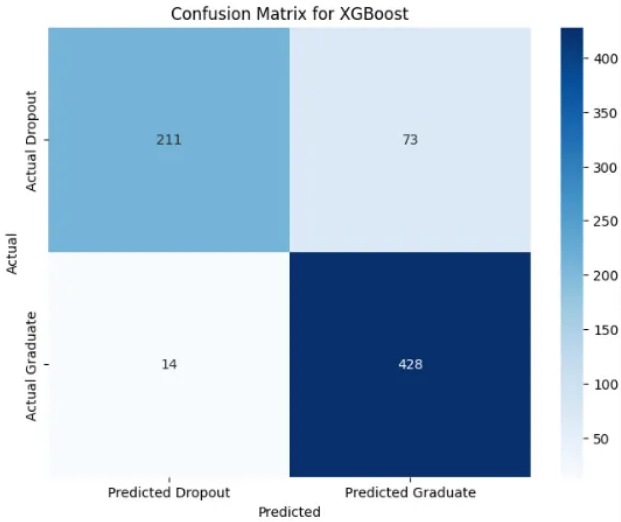
(b) SVM



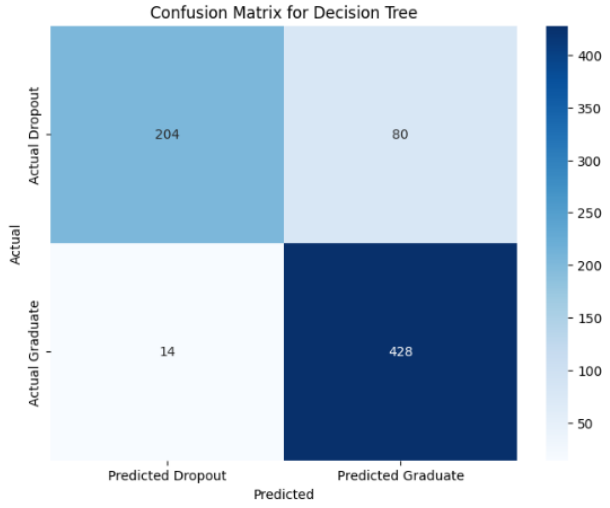
(c) RF



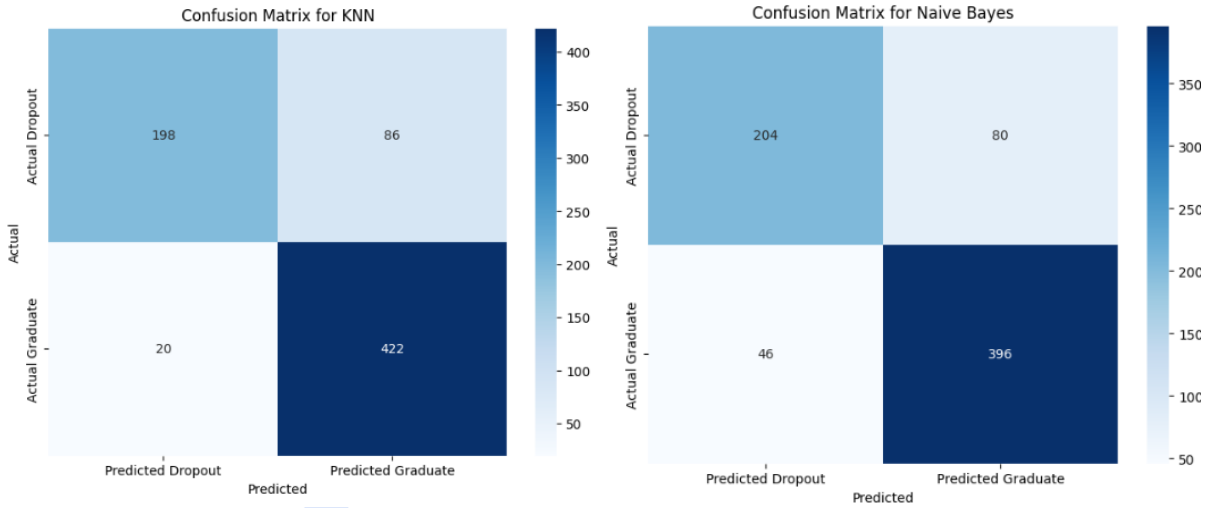
(d)Logistic Regression



(e) XGBoost



(f) DT



(g) KNN (h)NB

Figure 2: Confusion Matrix for classification models, including GBA, SVM, RF, LR, XGBoost, DT, KNN, NB

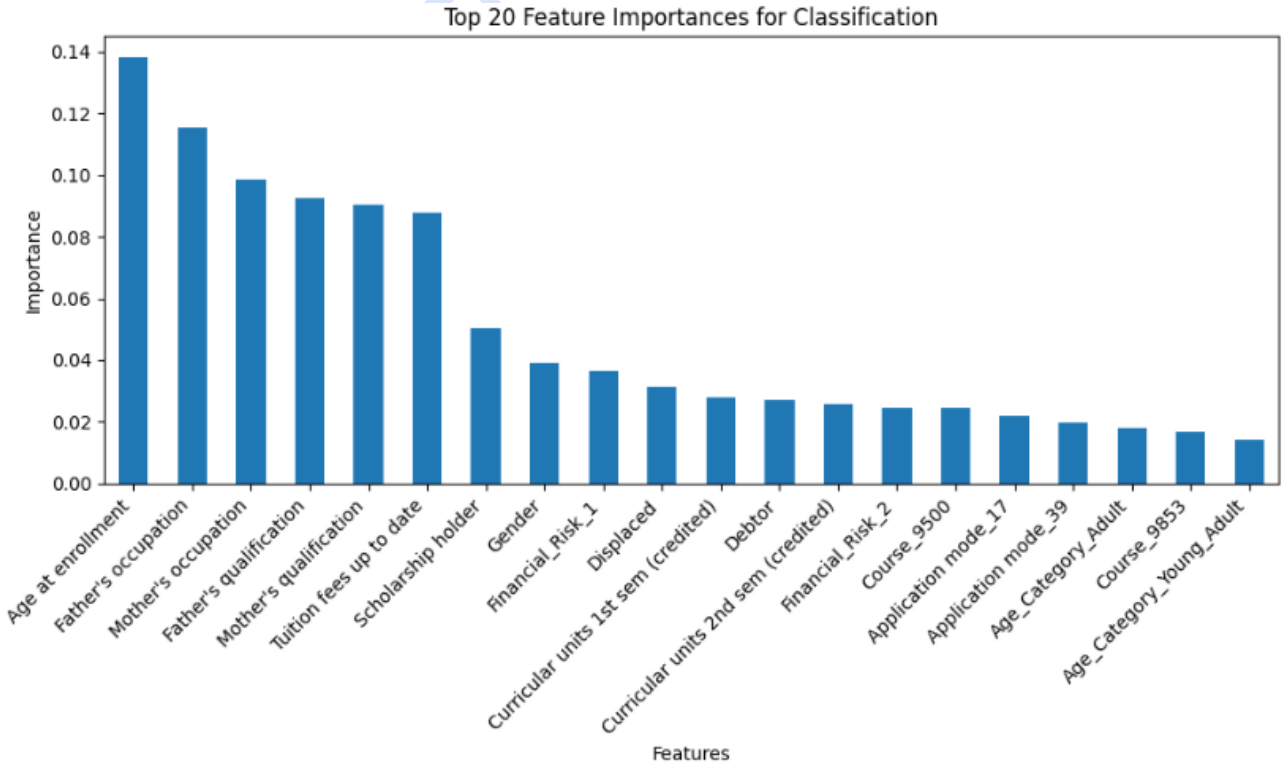


Figure 3: Top 20 Feature Importances for Classification

4.2 Model Performance Across Regression Techniques

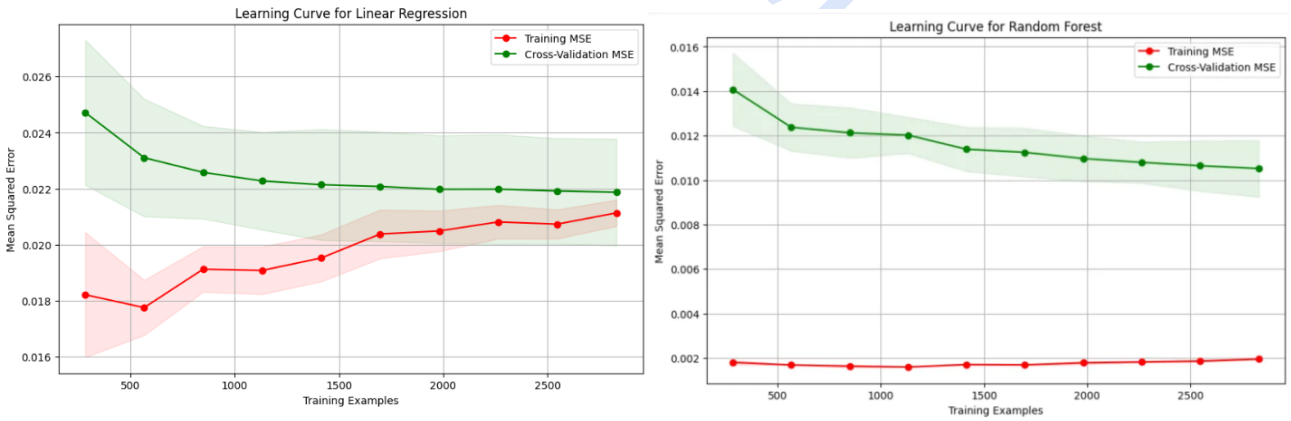
Since the performance of the base learners depends on various hyperparameters, Grid Search with Cross-Validation (Grid Search CV) was employed to identify the optimal combination of hyperparameters within predefined ranges, informed by findings from prior studies, as shown in Table 5. It summarizes the performance metrics for each model, including train MSE, test MSE, train R^2, and test R^2. The Stacking Regressor, combining the strengths of RF, XGBoost, and LR, achieved the highest test R^2 of 0.930 and the lowest test MSE of 0.010330, thereby outperforming all individual models. RF followed with a test R^2 of 0.923 and a test MSE of 0.011306, while XGBoost achieved a

test R^2 of 0.927 and a test MSE of 0.010740. SVR and LR exhibited lower performance, with test R^2 values of 0.895 and 0.821, respectively. These results highlight the efficacy of ensemble methods in predicting student performance.

Table 5 Model Performance Metrics

Model	Train		Test	
	MSE	R^2	MSE	R^2
LR	0.021210	0.858070	0.026386	0.821211
RF	0.002055	0.986246	0.011306	0.923393
XGBoost	0.002241	0.985007	0.010740	0.927226
SVR	0.005365	0.964098	0.015545	0.894670
Stacking Regressor	0.002750	0.981598	0.010330	0.930005

Learning curves for all models further illustrate their reliability in Figure 4. The Stacking Regressor demonstrates stable convergence, with CV MSE stabilizing at approximately 0.010, indicating minimal overfitting and robust generalization. RF and XGBoost also show stable learning curves, with CV MSE values converging around 0.011 and 0.010, respectively, reflecting their strong performance. SVR exhibits a higher CV MSE (around 0.015), consistent with its lower test R^2, while LR shows the highest CV MSE (around 0.022), aligning with its poorer performance and indicating potential underfitting. The results showed that the Stacking Regressor outperformed other models, achieving a test R^2 of 0.930, demonstrating the value of ensemble methods in student performance prediction. These findings provide a robust framework for understanding the effectiveness of different regression techniques in educational contexts.



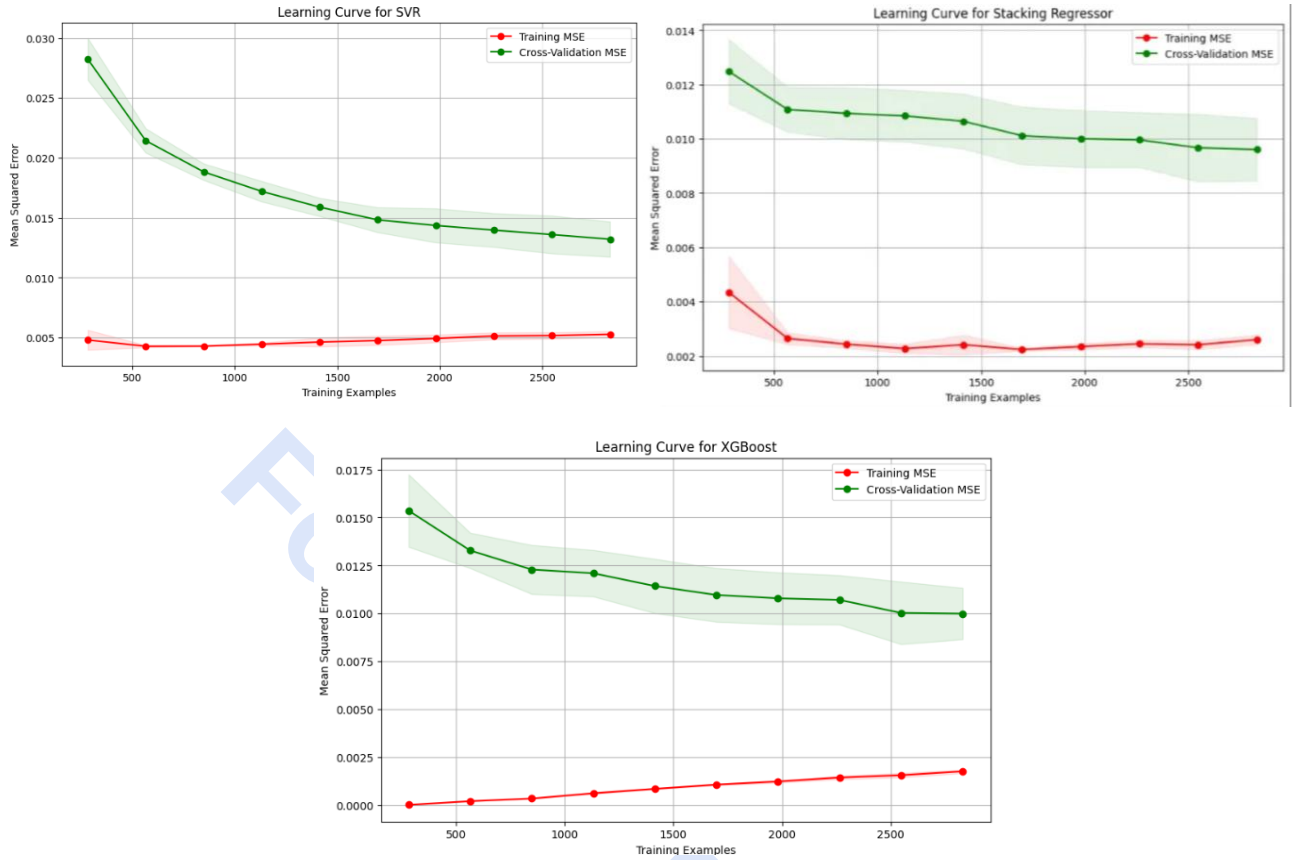


Figure 4 MSE learning curves for LR, RF, SVR, Stacking Regressor, XGBoost

4.3 Model Performance Across Cluster Techniques

Figure 5 visually compares five clustering algorithms: KMeans, Agglomerative Clustering, GMM, DBSCAN, and Spectral Clustering. These algorithms are evaluated using three key clustering quality metrics: Silhouette score (top chart), DB score (middle chart), and Balance (Gini Coefficient) (bottom chart). Each bar plot provides a side-by-side view of model performance, enabling direct and intuitive comparison across the methods. As illustrated in the figure, KMeans achieved the highest Silhouette Score, indicating well-defined and coherent clusters, while maintaining a relatively low DB Score, suggesting good separation between clusters. The results showed that KMeans was the most effective clustering algorithm.

We adopted the evaluation metrics described in Table 6 to assess clustering quality. The Silhouette score quantifies how similar each sample is to its cluster compared to others (higher values indicate better clustering). The DB score evaluates intra-cluster similarity and inter-cluster differences (lower values indicate more distinct clusters). The Gini Coefficient measures cluster size balance, with an ideal value around 0.65, indicating a fair distribution of data points across clusters.

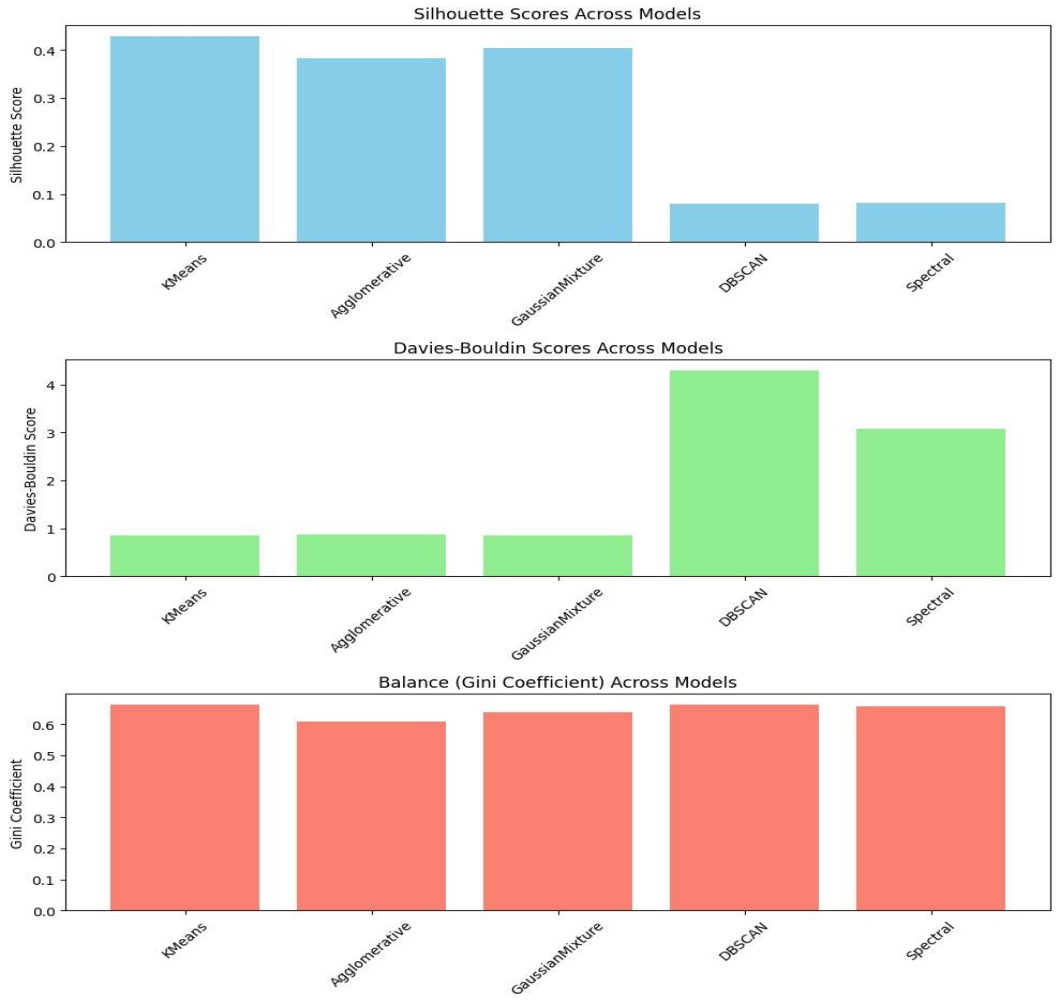


Figure 5 Comparison of five clustering algorithms

KMeans emerged as the most balanced and high-performing method, achieving a Silhouette score of 0.429, a DB Score of 0.856, and a Gini Coefficient of 0.664. Based on these metrics, it was selected as the best-performing clustering algorithm for further analysis.

Table 6 Comparison cluster algorithms

Model	Silhouette Score ↑	Davies-Bouldin Score ↓	Gini Coefficient ≈0.65
KMeans	**0.429**	**0.856**	**0.664**
Agglomerative	0.382	0.876	0.609
GMM	0.403	0.857	0.638
DBSCAN	0.080	4.302	0.665
Spectral	0.082	3.081	0.658

Figures 6, 7, and 8 illustrate the results of KMeans clustering applied to the student dataset, as visualized in a 2D UMAP embedding. Figure 6 reveals the segmentation of students into three distinct clusters labeled as Segment 0 (Low), Segment 1 (Medium), and Segment 2 (High). The clusters are visually well-separated, with minimal overlap, particularly between Segment 0 (purple, 1,460 students) and Segment 2 (yellow, 1,648 students), highlighting the effective partitioning by the KMeans algorithm. Segment 1 (teal, 1,316 students) appears more centrally located in the feature space, suggesting intermediate characteristics shared with the other two segments.

The Silhouette score of 0.429 supports a moderate cluster cohesion and separation, validating the clustering structure. Additionally, star-shaped markers denote the centroids of each cluster, offering a precise reference point for the core position of each student group within the embedded space. This visualization confirms the presence of meaningful subgroups and highlights the interpretability of the clustering outcome using KMeans.

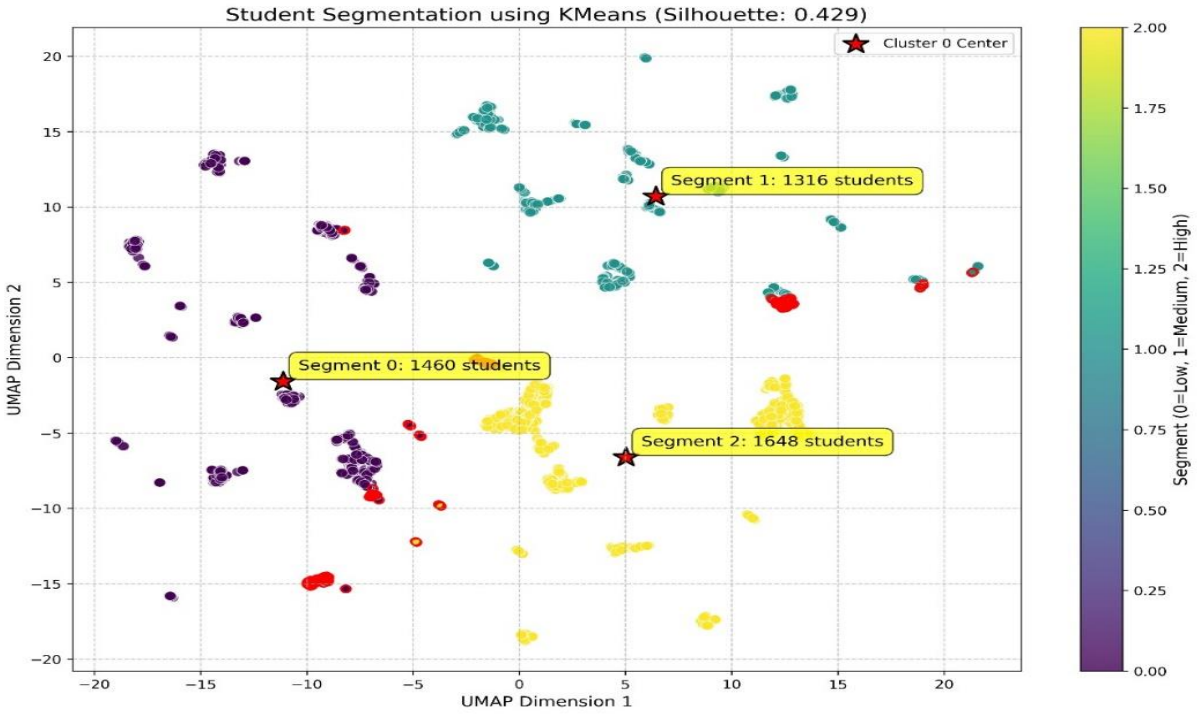


Figure 6 Student Segmentation using KMeans (Silhouette: 0.429)

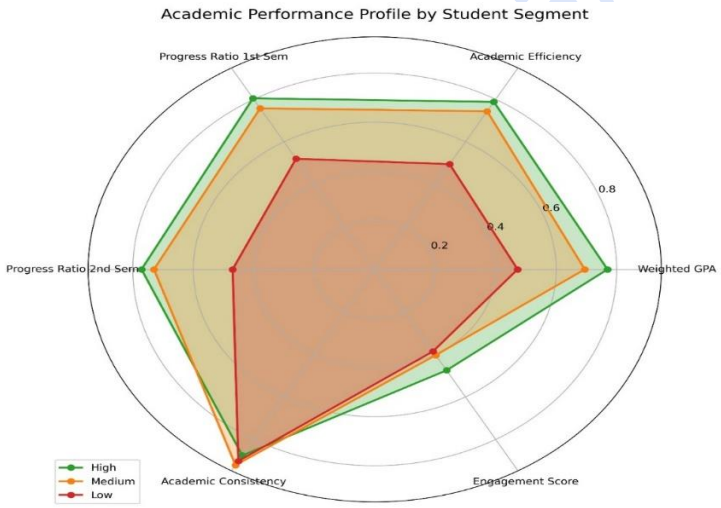


Figure 7 Academic Performance Profile by Student Segment

Figure 7 presents a comparative analysis of the three identified student segments across six key academic metrics, highlighting evident disparities in performance. The high-performing segment (green) consistently exhibits superior outcomes across all dimensions, with particularly notable

advantages in weighted GPA and academic efficiency. The medium segment (orange) maintains moderate performance levels, while the low segment (red) demonstrates considerable gaps in all metrics except academic consistency except for academic consistency. Additionally, including first and second semester progress ratios provides insight into temporal performance trends, showcasing the relative stability or fluctuation in academic outcomes over time. The hexagonal shape of each radar plot offers an intuitive and compact representation of each segment's academic profile, making it easy to visualize and interpret each group's relative strengths and weaknesses.

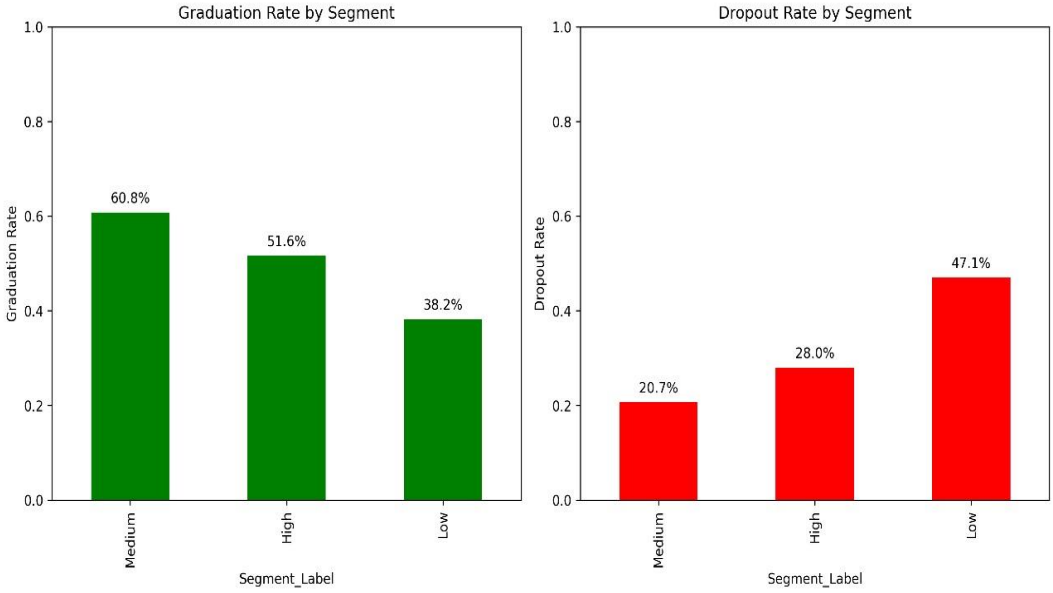


Figure 8 Graduation Rate and Dropout Rate by Segment

Figure 8 displays the distribution of key outcome metrics, graduation and dropout rates, across the three student segments, uncovering an inverse relationship between these two indicators. Notably, the medium-performing segment demonstrates the highest graduation rate (60.8%), outperforming even the high-performing segment (51.6%), while the low-performing segment falls significantly with a graduation rate of only 38.2%. The corresponding dropout rates further emphasize this trend, with the low segment showing the highest attrition rate at 47.1%, followed by the high segment (28.0%) and the medium segment (20.7%). This unexpected pattern is where the high-performing group has a lower graduation rate than the medium-performing group despite stronger academic metrics. Factors beyond academic achievement, such as motivation, support systems, or external pressures, may significantly influence student persistence and completion.

5 Conclusions

This study proposed a comprehensive framework for analyzing student performance through a robust multi-stage data science pipeline. It begins with meticulous data preprocessing, which involves engineering meaningful behavioral features such as the Academic Consistency and Engagement Score. The methodology also incorporates appropriate encoding techniques for categorical variables and customized scaling strategies to ensure the comparability of features. Feature selection is performed using low-variance filtering, and dimensionality reduction techniques like PCA and UMAP are applied. This process enables the transformation of high-dimensional data into a compact, informative feature space, suitable for further analysis. Regression models, including RF, XGBoost, and a Stacking Regressor, are employed to predict continuous academic outcomes. Among these,

the Stacking Regressor exhibited the highest predictive accuracy, effectively utilizing ensemble learning to enhance performance. Classification models were also leveraged to predict categorical academic statuses, taking advantage of the refined features generated during preprocessing. This approach contributes to a deeper understanding of student trajectories. Various clustering algorithms were assessed for unsupervised segmentation, with KMeans proving to be the most effective. Its performance, validated through metrics such as the Silhouette Score and Davies-Bouldin Index, revealed well-separated and interpretable clusters of students based on their academic performance and behavioral attributes.

Author contributions All Authors contributed equally to this work and approved the final manuscript.

Data availability Not applicable.

Declarations

Competing interests The authors declare that they have no competing interests.

6 References

[1] Bharara S., Sabitha S., and Bansal A. (2018) Application of learning analytics using clustering data Mining for Students. disposition analysis, *Education and Information Technologies*. 23(2), pp. 957-984. **https://doi.org/10.1007/s10639-017-9645-7**

[2] Nunn S., Avella J. T., Kanai T., and Kebritchi M. (2016) Learning analytics methods, benefits, and challenges in higher education: a systematic literature review, *Online Learning*. 20(2), pp. 13–29. **https://doi.org/10.24059/olj.v20i2.790**

[3] Liu M. and Yu D. (2023) Towards intelligent E-learning systems, *Education and Information Technologies*. **28**(7), pp. 7845–7876. **https://doi.org/10.1007/s10639-022-11479-6**.

[4]Alsariera Y. A., Baashar Y., Alkawsi G., Mustafa A., Alkahtani A. A., and Ali N. (2022) Assessment and evaluation of different machine learning algorithms for predicting student performance, *Computational Intelligence and Neuroscience*. **2022**, pp.1–11. **https://doi.org/10.1155/2022/4151487**.

[5]Talwar S., Talwar M., Tarjanne V., and Dhir A. (2021) Why retail investors trade equity during the pandemic? An application of artificial neural networks to examine behavioral biases, *Psychology and Marketing*. **38**(11), pp. 2142-2163. **https://doi.org/10.1002/mar.21550**

[6]Kumar S., and Sachdeva R. (2023). A Survey of Different Supervised Learning-Based Classification Models for Student's Academic Performance Prediction, International Conference on Innovative Computing and Communications. Lecture Notes in Networks and Systems, 492. Springer, Singapore. https://doi.org/10.1007/978-981-19-3679-1_44

[7]Albreiki B., Zaki N., and Alashwal H. (2021). A systematic literature review of student' performance prediction using machine learning techniques. *Education Sciences, 11*(9), 552. https://www.mdpi.com/2227-7102/11/9/552

[8]Wickramasinghe I., Aragon R. and Valles J. (2024) Influential factors on elementary students' academic performance and identifying the appropriate performance group. *Discov Educ* 3, 76. https://doi.org/10.1007/s44217-024-00167-x

[9]Alalawi K., Athauda R., Chiong R., and Renner I. (2025) Evaluating the student performance prediction and action framework through a learning analytics intervention study, Education and Information Technologies, 30, pp. 2887–2916. https://doi.org/10.1007/s10639-024-12923-5

[10]Alalawi, K., Athauda, R., and Chiong, R. (2023). Contextualizing the current state of research on the use of machine learning for student performance prediction: A systematic literature review. *Engineering Reports*, *5*(12), pp.1-25. https://doi.org/10.1002/eng2.12699

[11]Usman M. M., Owolabi O., and Ajibola A. A. (2020). Feature selection: It importance in performance prediction. *IJESC, 10*(5), pp. 25625–25632

[12]Zaffar M., Hashmani M. A., Savita K. S., Rizvi S. S. H., and Rehman M. (2020). Role of FCBF feature selection in educational data mining. *Mehran University Research Journal of Engineering & Technology,39*(4), pp. 772–778

[13]Ajibade S. S. M., Dayupay J., Ngo-Hoang D. L., Oyebode O. J., and Sasan J. M. (2022). Utilization of ensemble techniques for prediction of the academic performance of students. *Journal of Optoelectronics Laser,41*(6), pp. 48–54
[14]Chen M., and Liu Zh. (2024) Predicting performance of students by optimizing tree components of random forest using genetic algorithm, Heliyon, 10 (12). doi: 10.1016/j.heliyon.2024.e32570
[15]Pujianto U., Prasetyo W.A., and Taufani A. R. (2020) Students Academic Performance Prediction with k-Nearest Neighbor and C4.5 on SMOTE-balanced data, 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 348 -353. doi: 10.1109/ISRITI51436.2020.9315439
[16] Sagala T M N., Permai S. D., A. Gunawan, Barus R. O., and Meriko C. (2022) Predicting Computer Science Student's Performance using Logistic Regression, 2022 5th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), pp. 817-82. doi: 10.1109/ISRITI56927.2022.10052968
[17] Khairy D., Alharbi N., Amasha M. A, Areed M. F., Alkhalaf S., and Abougalala R. A. (2024) Prediction of student exam performance using data mining classification algorithms. Educ Inf Technol 29, pp. 21621–21645 https://doi.org/10.1007/s10639-024-12619-w
[18] Kord A., Aboelfetouh A., and Shohieb S. M. Academic course planning recommendation and students' performance prediction multi-modal based on educational data mining techniques. *J Comput High Educ* (2025). https://doi.org/10.1007/s12528-024-09426-0
[19]Sixhaxa K., Jadhav A., and Ajoodha R. (2022) Predicting Students Performance in Exams using Machine Learning Techniques. In: *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, pp. 635–640.
[20]Hussain S., and Khan M. Q. (2023) Student-performulator: predicting students' academic performance at secondary and intermediate level using machine learning, Annals of data science, 10, pp. 637-655
[21]Starczewski, A., and Krzyżak, A. (2015). Performance Evaluation of the Silhouette Index, Artificial Intelligence and Soft Computing. ICAISC 2015, vol 9120. Springer, Cham. https://doi.org/10.1007/978-3-319-19369-4_5
[22]Madhavan G. (2025). Support Vector Machine (SVM). In: Mastering Machine Learning: From Basics to Advanced. Transactions on Computer Systems and Networks. Springer, Singapore. https://doi.org/10.1007/978-981-97-9914-5_14
[23]Emami S., and Martínez-Muñoz G. (2025) Condensed-gradient boosting. *Int. J. Mach. Learn. & Cyber.* **16**, pp. 687-701 https://doi.org/10.1007/s13042-024-02279-0
[24]Fürnkranz J. (2011) Decision Tree. Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_204
[25]Attanasi E.D., and Coburn T.C. (2023) Random Forest, Encyclopedia of Mathematical Geosciences. Encyclopedia of Earth Sciences Series. Springer, Cham. https://doi.org/10.1007/978-3-030-85040-1_265
[26]Song X. (2024) Student performance prediction employing k-Nearest Neighbor Classification model and meta-heuristic algorithms. *Multiscale and Multidiscip. Model. Exp. and Des.* **7**, pp. 4397–4412 https://doi.org/10.1007/s41939-024-00481-9
[27]Sivasakthi M., and Padmanabhan K. R. A. (2023). Prediction of Students Programming Performance Using Naïve Bayesian and Decision Tree. Soft Computing for Security Applications. Advances in Intelligent Systems and Computing, vol 1428. Springer, Singapore. https://doi.org/10.1007/978-981-19-3590-9_8
[28]Zainol Z., Nohuddin P. N. E., Husin H. S., Rauf U. F. A., and Mutalib M. Y. A. (2024) A Regression Analysis for Predicting Student Academic Performance. Tech Horizons. SpringerBriefs in Applied Sciences and Technology. Springer, Cham. https://doi.org/10.1007/978-3-031-63326-3
[29]Amroune M. (2022) Support vector regression-bald eagle search optimizer-based hybrid approach for short-term wind power forecasting. *J. Eng. Appl. Sci.* 69, 107 https://doi.org/10.1186/s44147-022-00161-w
[30]Ahmar A. S., Rais Z., and Tunnas F. (2025) Comparative analysis of support vector regression (SVR) and SutteARIMA in predicting coal prices in Indonesia. *Qual Quant* https://doi.org/10.1007/s11135-025-02162-2
[31]Montesinos López O. A., Montesinos López A., and Crossa J. (2022). Support Vector Machines and Support Vector Regression. In: Multivariate Statistical Machine Learning Methods for Genomic Prediction. Springer, Cham. https://doi.org/10.1007/978-3-030-89010-0_9
[32]Sreevalsan-Nair J. (2022) K-Means Clustering. Encyclopedia of Mathematical Geosciences. Encyclopedia of Earth Sciences Series. Springer, Cham. https://doi.org/10.1007/978-3-030-26050-7_171-1
[33]Wang F., Zhang C., and Lu N. (2005). Boosting GMM and Its Two Applications, Multiple Classifier Systems. MCS 2005. Lecture Notes in Computer Science, 3541. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11494683_2
[34]Elsedimy E.I., Elhadidy H., and Abohashish, S. M. M. (2024) A novel intrusion detection system based on a hybrid quantum support vector machine and improved Grey Wolf optimizer. *Cluster Comput*, **27**, pp. 9917–9935. https://doi.org/10.1007/s10586-024-04458-8

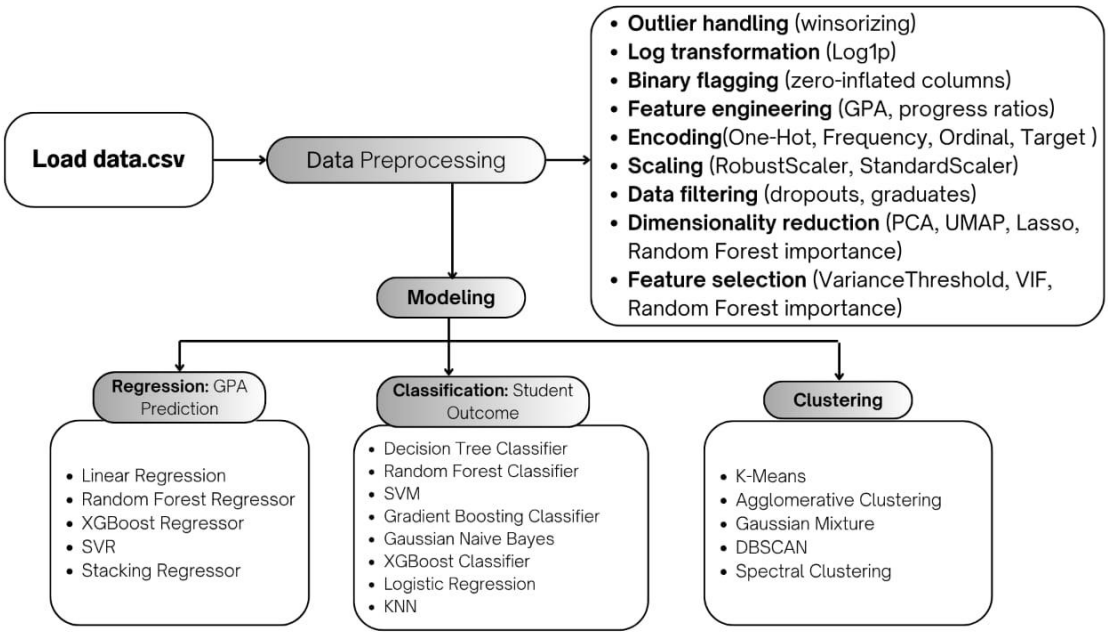
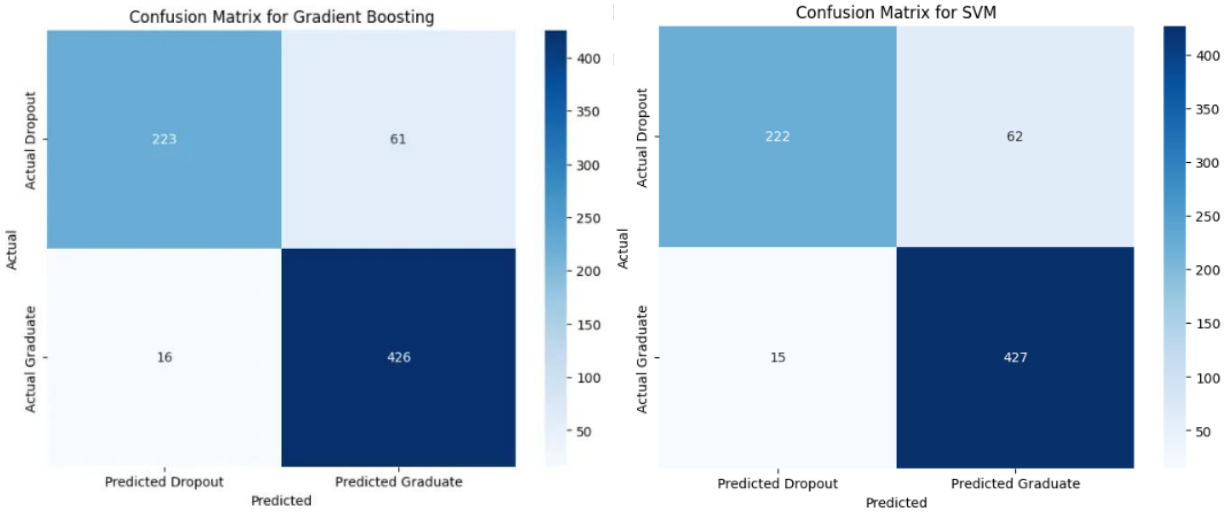
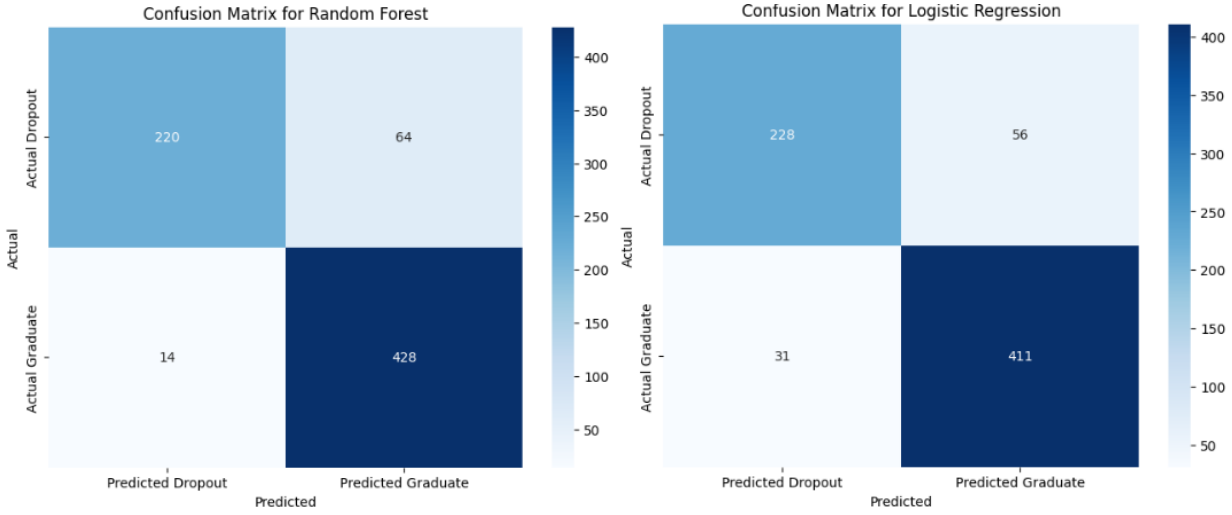


Figure 1 Workflow of the proposed methodology.



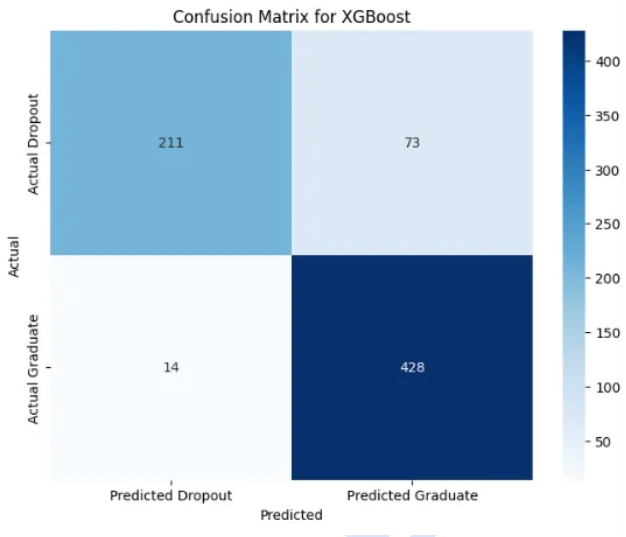
(a) GBA

(b) SVM

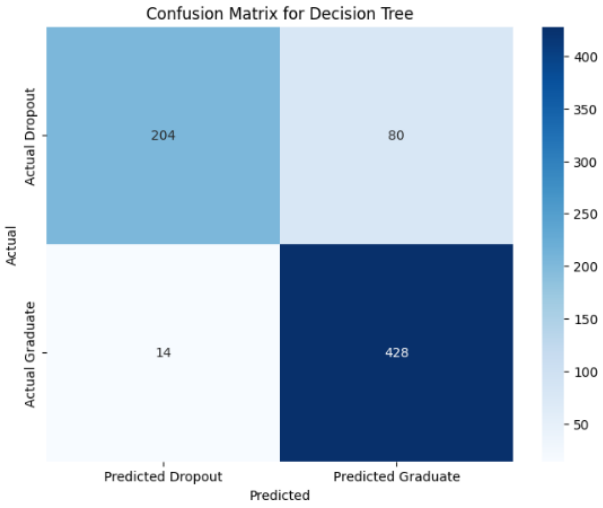


(c) RF

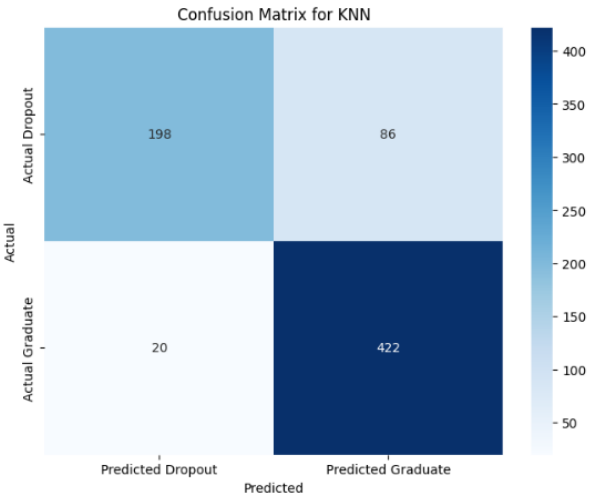
(d)Logistic Regression



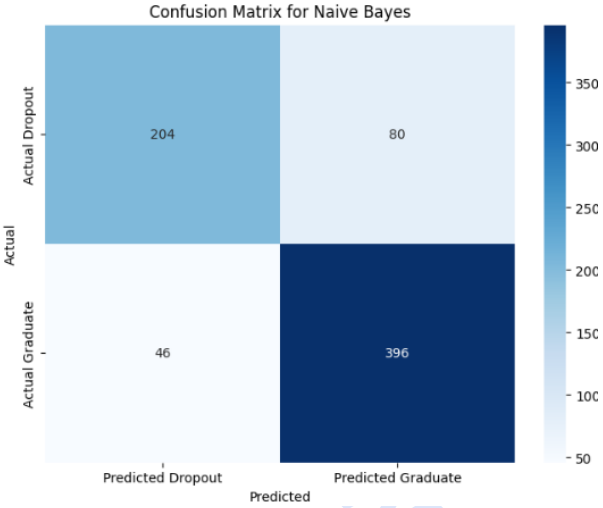
(e) XGBoost



(f) DT



(g) KNN



(h)NB

Figure 2: Confusion Matrix for classification models, including GBA, SVM, RF, LR, XGBoost, DT, KNN, NB

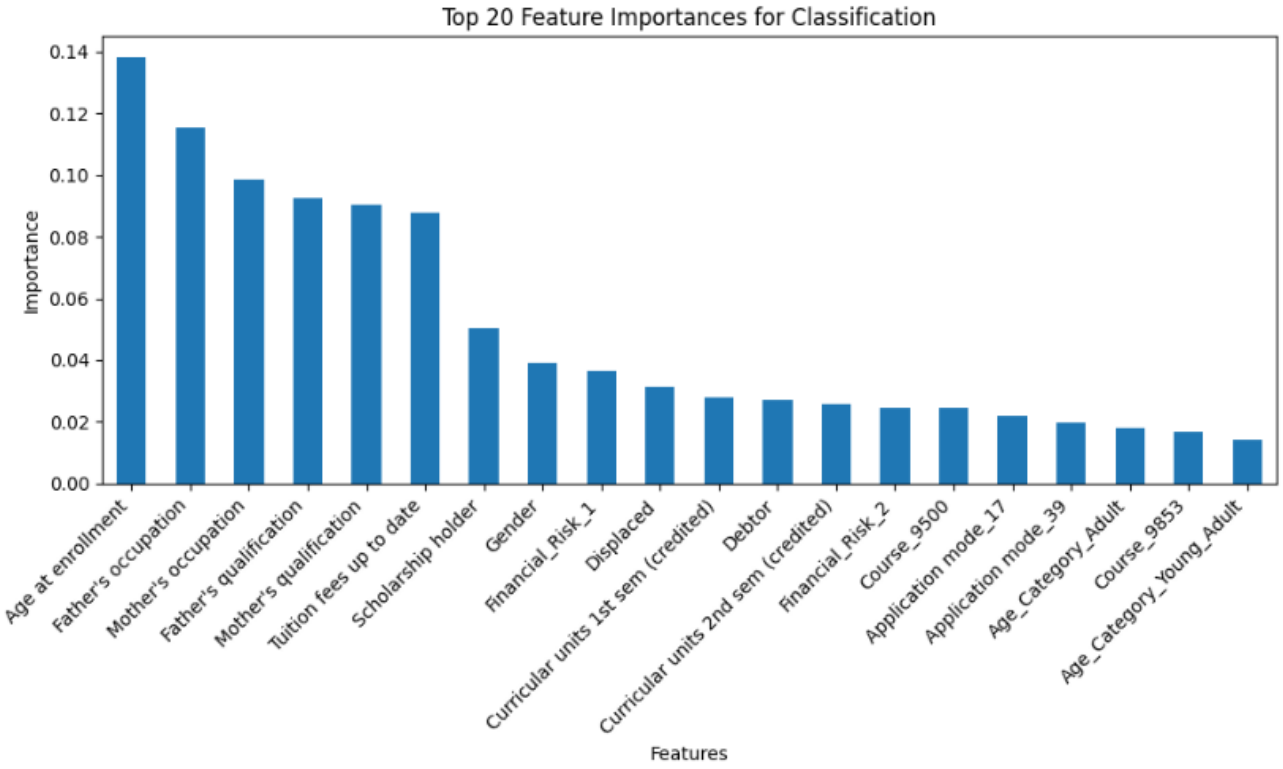
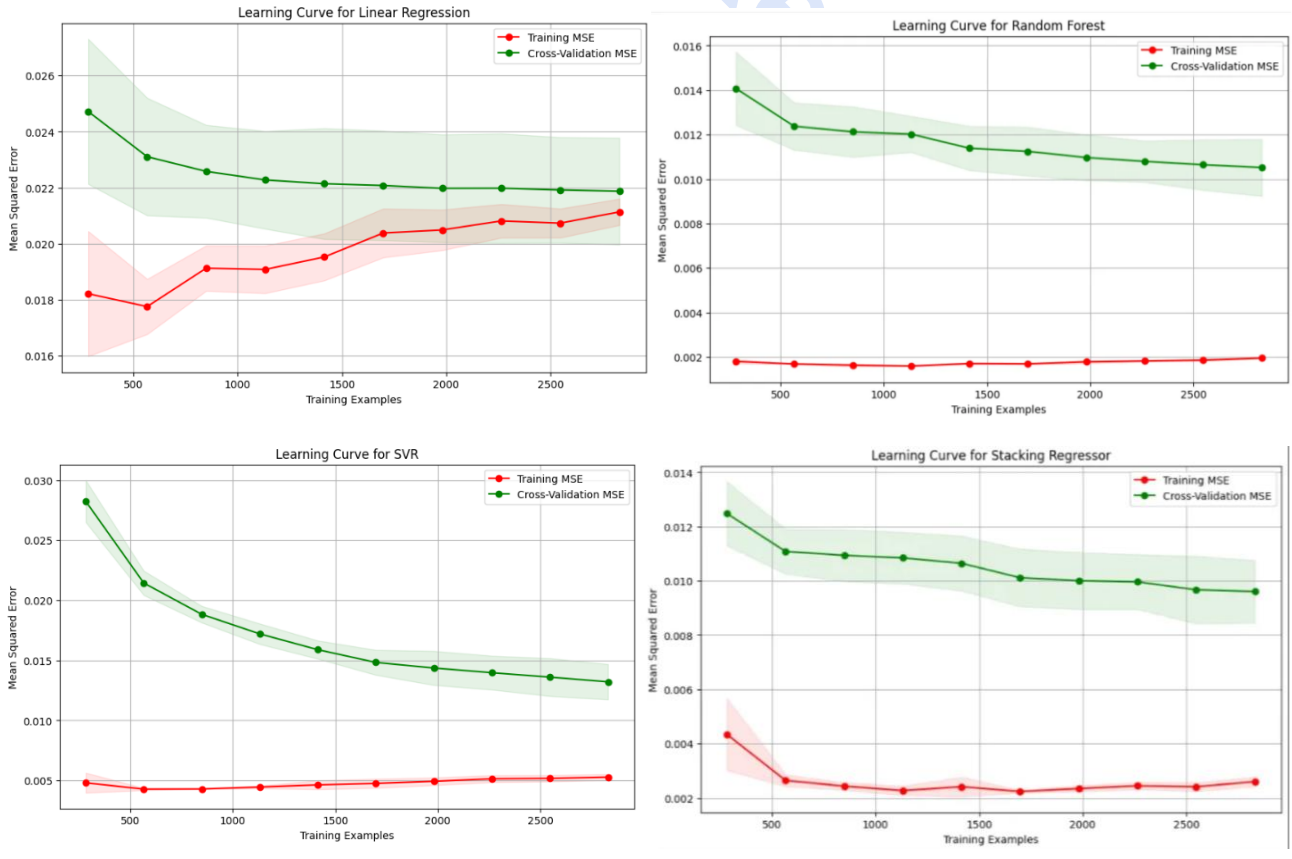


Figure 3: Top 20 Feature Importances for Classification



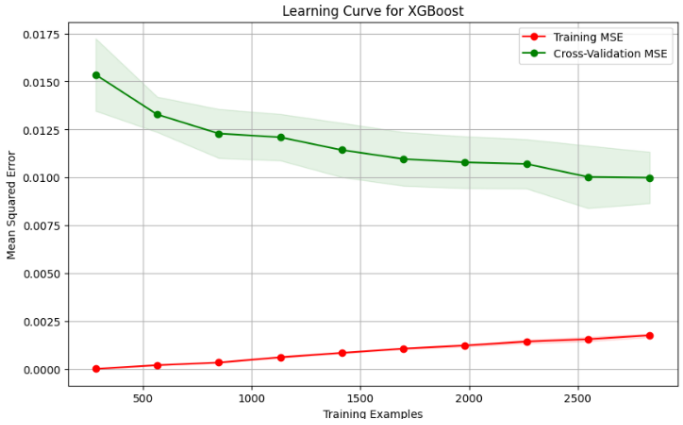


Figure 4 MSE learning curves for LR, RF, SVR, Stacking Regressor, XGBoost

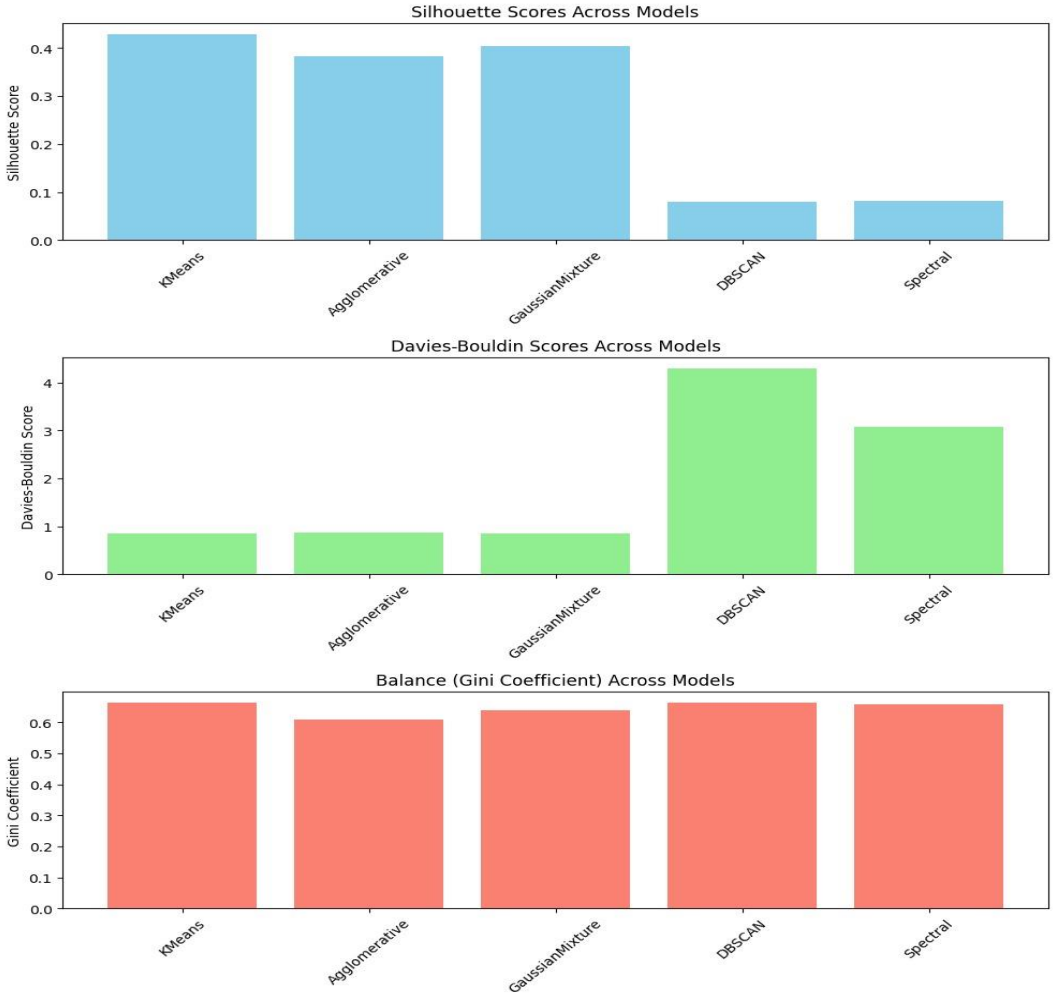


Figure 5 Comparison of five clustering algorithms

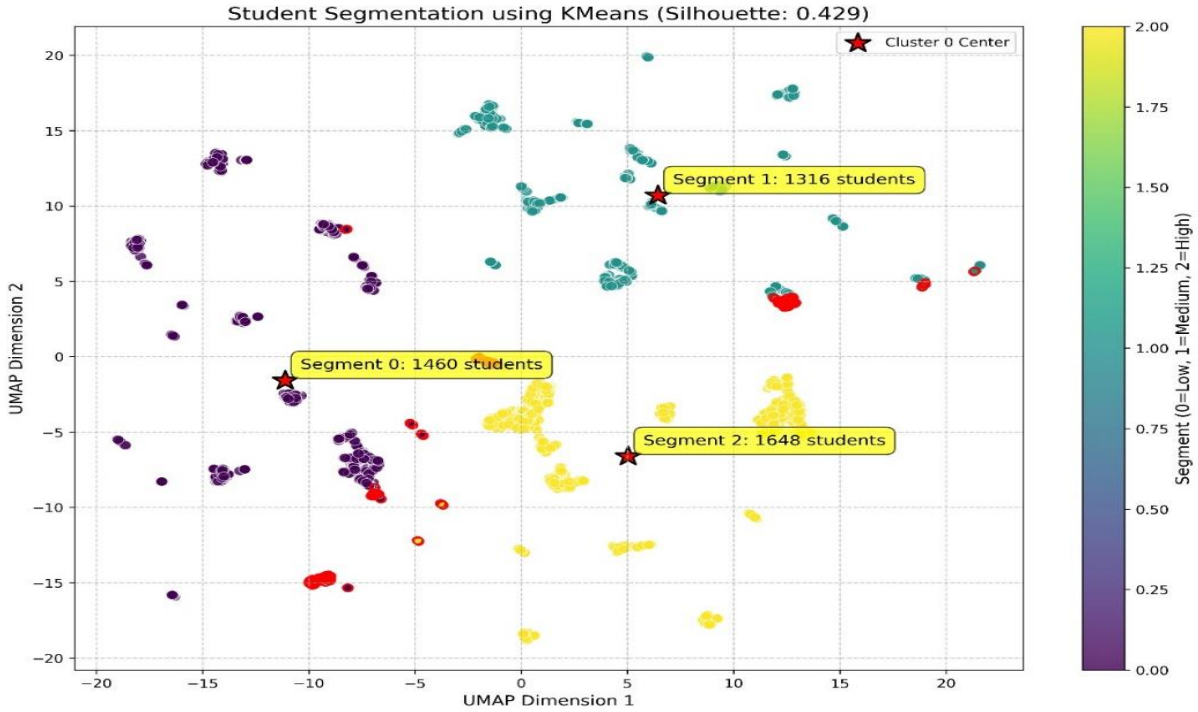


Figure 6 Student Segmentation using KMeans (Silhouette: 0.429)

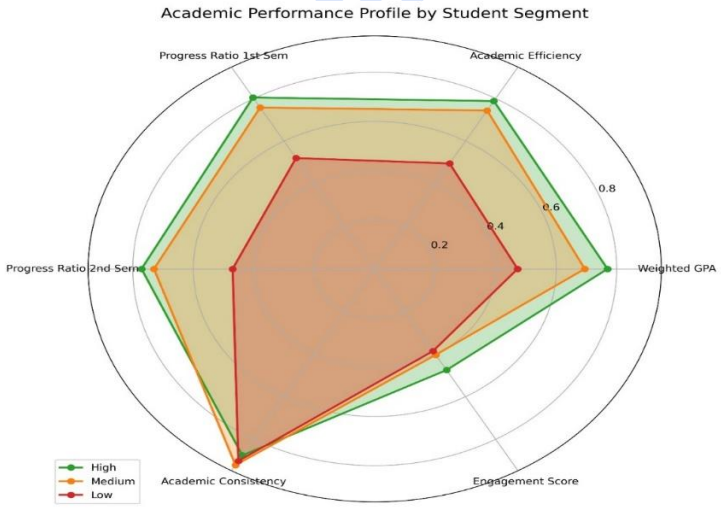


Figure 7 Academic Performance Profile by Student Segment

Table 1: Scaling techniques applied to numerical features by category

Category	Attribute	Frequency	Scaler Used
Academic Performance	Weighted_GPA; GPA; Curricular units 1st sem (grade); Curricular units 2nd sem (grade); Grade_Improvement;	5	RobustScaler
Academic Progress	Academic_Efficiency; Progress_Ratio_1st_Sem; Progress_Ratio_2nd_Sem; Academic_Consistency;	4	MinMaxScaler
Academic Engagement	Engagement_Score; Curricular units 1st sem (evaluations); Curricular units 2nd sem (evaluations); Total_Academic_Load;	4	StandardScaler
Academic history	Admission grade; Previous qualification (grade); Admission_Strength;	3	RobustScaler
Socioeconomic	GDP; Socioeconomic_Status; Unemployment rate; Inflation rate;	4	StandardScaler
Demographic	Age at enrollment_log; Application order;	2	MinMaxScaler

Table 2 Hyper-parameters for the competitive classification, regression, and cluster models.

Models		Hyper-parameters	Value
Classification	**LR**	C	0.1
		penalty	l2
	RF	n_estimators	100
		max_depth	10
	SVM	C	1.0
		kernel	rbf
	DT	max_depth	5
		min_samples_split	10
	XGBoost	n_estimators	50
		learning_rate	0.05
	KNN	n_neighbors	5
	GBA	n_estimators	100
		learning_rate	0.05
Regression	**RF**	max_depth	15
		min_samples_split	2
	XGBoost	learning_rate	0.1
		n_estimators	100
	SVR	C	1
		epsilon	0.01
Clustering	**KMeans**	n_clusters	3
		Init	‘k-means++’
		n_init	25
		max_iter	500
		tol	1e-5
		random_state	42
	GGM	n_components	3
		covariance_type	‘full’
		max_iter	200
		n_init	10

		random_state	43
	Agglomerative	n_clusters	3
		linkage	‘ward’
		metric	'euclidean'
	DBSCAN	eps	0.9498963074037952
		min_samples	44
		metric	'euclidean'
		algorithm	'auto'
		leaf_size	40
	Spectral	n_clusters	3
		assign_labels	'discretize'
		affinity	'nearest_neighbors'
		n_neighbors	15
		random_state	44

Table 3 Classification Model Performance Metrics

Model	Accuracy	Precision	Recall	F1 Score
GBA	0.893939	0.903899	0.874506	0.884944
SVM	0.893939	0.904960	0.873877	0.884750
RF	0.892562	0.905045	0.871487	0.882955
Logistic Regression	0.880165	0.880197	0.866341	0.872035
XGBoost	0.880165	0.896035	0.855642	0.868409
DT	0.870523	0.889150	0.843318	0.856901
KNN	0.853994	0.869483	0.825967	0.838633
NB	0.826446	0.823966	0.807119	0.813395

Table 4 Classification Model Training and Test Accuracy

Model	Training Accuracy	Test Accuracy
SVM	0.9287	0.8939
GBA	0.9239	0.8939
RF	0.9349	0.8926
Logistic Regression	0.8919	0.8802
XGBoost	0.8953	0.8802
DT	0.8926	0.8705
KNN	0.8957	0.8540
NB	0.8209	0.8264

Table 5 Model Performance Metrics

Model	Train		Test	
	MSE	**R^2**	**MSE**	**R^2**
LR	0.021210	0.858070	0.026386	0.821211
RF	0.002055	0.986246	0.011306	0.923393
XGBoost	0.002241	0.985007	0.010740	0.927226
SVR	0.005365	0.964098	0.015545	0.894670
Stacking Regressor	0.002750	0.981598	0.010330	0.930005

Table 6 Comparison cluster algorithms

Model	Silhouette Score ↑	Davies-Bouldin Score ↓	Gini Coefficient ≈0.65
KMeans	**0.429**	**0.856**	**0.664**
Agglomerative	0.382	0.876	0.609
GMM	0.403	0.857	0.638
DBSCAN	0.080	4.302	0.665
Spectral	0.082	3.081	0.658