MATHEMATICAL MODELING AND COMPUTING, Vol. 11, No. 3, pp. 814–825 (2024)

Predicting students' academic performance and modeling using data mining techniques

Jedidi Y.[1], Ibriz A.[1], Benslimane M.[1], Hachmoud A.[1], Tmimi M.[1], Hajjioui Y.[1], Rahhali M.[2]

[1] *Innovative Technologies Laboratory, EST, Sidi Mohamed Ben Abdellah University, Fez, Morocco*
[2] *ENSA, Sidi Mohamed Ben Abdellah University, Fez, Morocco*

(Received 7 January 2024; Revised 13 August 2024; Accepted 15 August 2024)

In educational institutions and universities, the issue of study interruptions can be addressed by using e-learning. As a result, this field has recently attracted a lot of attention. In this study, we applied four machine-learning methods to predict students' academic progress: logistic regression, decision trees, random forests, and Naive Bayes. The Open University Learning Analytics Dataset (OULAD), which contains a subset of the OU student data, was the source of the student data for all of these techniques. There is information regarding the students' VLE interactions as well as their demographics. Nowadays universities frequently use data mining techniques to analyze available data and extract knowledge and information that helps in decision making. The percentage split and the 10-fold cross-validation are used to measure and compare the prediction performance of four classifiers. When employing the percentage split, it was shown that the Naive Bayes classifier performs better than other classifiers, obtaining an overall prediction accuracy of 93%. This study aims to assist teachers in enhancing students' academic performance.

Keywords: *student's performance prediction; big data; educational data mining (EDM); machine learning; classifiers.*

2010 MSC: 68T10 **DOI:** 10.23939/mmc2024.03.814

1. Introduction

Due to the advent of innovative information and communication technology, including big data technology and cloud computing, the technology stack for human learning is fast changing. Moreover, learning methods change daily. As a result, e-learning systems must develop new strategies and tools to fulfill the growing demands of millions of learners worldwide.

Learning efficiency in universities is regarded as one of the most essential factors in a country's development, so it is critical that the universities adopt measures to improve their quality programs. These interventions can be planned after measuring the students' performance, as the advanced failure rate computation can assist academic institutions in making preventive measures to reduce this rate. When analyzing huge educational databases to estimate student success, however, most institutions of higher learning face issues [1]. This is due to the fact that they only employ traditional statistical approaches instead of new and efficient prediction tools like Educational Data Mining "EDM", which is the most widely used method for evaluating and predicting student performance [2]. EDM is the process of collecting meaningful data and patterns from a large educational database in order to predict student achievement [3]. Student performance can be improved more successfully through more effective strategic programs as a result of greater knowledge.

In fact, classification is among the most valuable data mining strategies in e-learning. Classification is a predictive analytic technique that provides predictions about datasets based on known results from various datasets [4]. Predictive models have the goal of helping us to forecast the undetermined values of variables of importance based on the known values of other variables. Learning a mapping from an input set of vector measurements to a scalar output is what predictive modeling is all about [5]. The

data is mapped into preset groupings of classes by classification [6]. For the most accurate prediction of student performance, prediction models that integrate all psychological, social, personal, and other variables are required. The order to predict student success with high precision is useful for identifying students with low academic achievement early on. It is necessary for the teacher to provide additional help to the identified students to enhance their performance in the future.

In this regard, the current study's objectives were set in order to help low academic achievers in university education, and they are as follows:

— Generation of a predictive variable data source.
— Based on discovered predictive variables, we construct a prediction model using classification data mining methods.
— Validation of the suggested model for higher education students. Which data mining prediction technique among Decision Trees (DT), Naive Bayes, logistic regression, and random forests performs best in this study?

2. Related work and research gap

Student performance is the most essential determinant of the quality of a university in higher education. Because of its importance in decision making, EDM is presently the technique most widely employed by researchers to predict and evaluate student performance.

There are two primary aspects in predicting student performance: attributes and prediction methods [7]. Student CGPA is the most utilized indicator in predicting student performance at university, according to this study [7]. Many studies have employed it (for example, [8, 9]). Quiz grades, assessments, lab work, and final exam marks are all typical criteria used by researchers to predict student achievement at university (e.g. [10]). Other factors such as social interaction networks and activities have been employed in a few studies (e.g. [11]).

On the other hand, input variables such as student demographics and extracurricular activities are rarely employed in the development process to predict student success at university. This is the study's main focus.

Students' performance prediction models have been built using a variety of data mining techniques, including classification and regression [7]. When the outcome variables are categorical (or discrete), the classification technique is used, while when the outcome variables are numerical, the regression technique is used (or continuous). In higher education, classification is the most widely used data mining approach [12]. K-Nearest Neighbor, Naive Bayes, and Decision Tree are just a few of the algorithms that could be used to predict student performance using the classification technique.

Researchers usually utilize decision tree techniques to predict student performance; for example, in [13], the authors Mishra T., Kumar D. employed various implementations of the decision tree technique to create a performance prediction model focuses on students' social integration. Authors Quadri M. M., Kalyankar N. V. in [14] employed decision tree techniques to predict student dropout.

For predicting student performance, various data mining classification algorithms have been used.

In the study [10], authors Arsad P. M., Buniyamin N., Artificial Neural Network (ANN) was used to predict the academic achievement of 505 students in 8 semesters. Authors Natek S. and Zwilling M., in study [15] developed a strategy to predict student success in certain courses using limited student small samples using Decision Trees (32 and 42 students). Authors Gray G., McGuinness C., and Owende P., in study [16] used SVM to predict students at risk's performance during their first year of study using a data set of 1 074 individuals.

3. Methodology

The approach recommended in this study to improve academic achievement prediction for students is a Data Mining technique. Data collection, preprocessing, classification, and interpretation are the four

primary phases of this methodology (see Figure 1). Data collection involves obtaining all information on students that is available while taking into account variables that affect student performance.

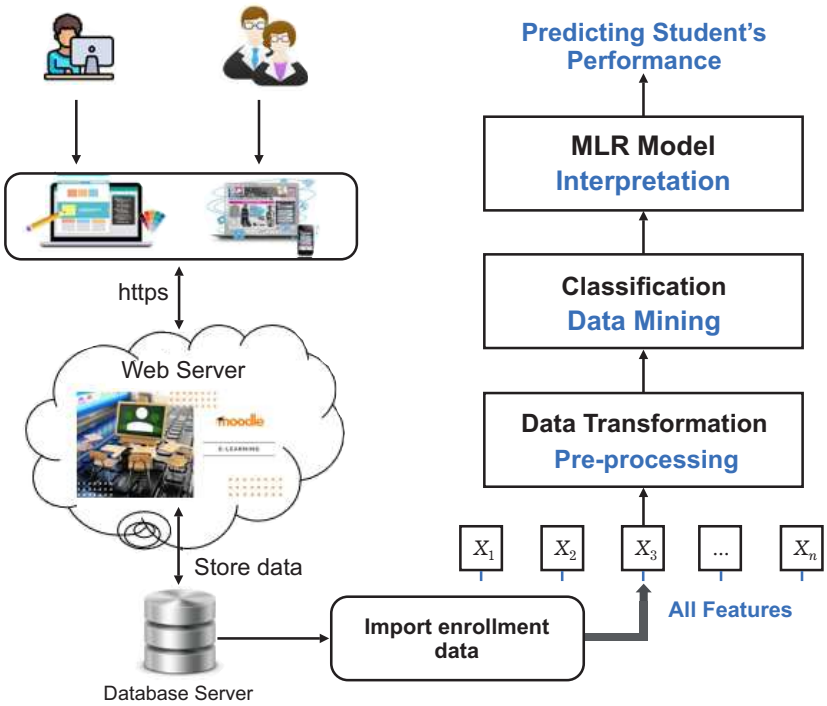

Fig. 1. A proposed approach to improve the prediction of students' performance.

3.1. Dataset and data preprocessing

The Open University Learning Analytics dataset (OULAD), which includes a portion of the OU student data from 2013 and 2014, was used as the source of the data for this study. Both demographic information about the students and information about their VLE interactions are included. The OULAD is an assortment of tabular student data from the academic years 2013 and 2014. Each table has unique data that can be connected to information from other tables via identifier columns. The dataset's data are organized as depicted in Figure 1. The student is the main focus because the dataset is student-oriented. Data on students' registrations for the modules and demographics is also included. The dataset includes the outcomes of the students' assessments for each triplet of student-module-presentation. A summary of each student's daily activity is kept in the log of their interactions with the Virtual Learning Environment (VLE). The dataset, which has 32 593 registered students and 22 module presentations, is freely accessible at `https://analyse.kmi.open.ac.uk/open_dataset`. The Open Data Institute (`http://theodi.org/`) has accredited OULAD [17].

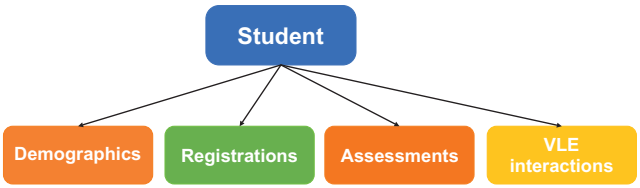

Fig. 2. Global dataset organization.

The student's registration for the modules and information about his or her demographics are linked. The dataset includes logs of student interactions with the VLE and the outcomes of the assessments for each participant-module-presentation triplet.

In general, we identify three sorts of data:

— **Demographics:** Refers to the fundamental data about the learners, such as their age, gender, region, level of education previously attained, etc.
— **Performance:** Represents the outcomes and accomplishments of students during their time studying at the Open University.
— **Learning Behavior:** The VLE's track of student activities is called learning behavior.

Through the corresponding "student" table, table studentInfo is connected to the assessments table and vle courses. The dataset's precise structure is displayed in Figure 3. Using the column id_student, the studentInfo can be connected to the studentAssessment, studentVle, and studentRegistration tables. Using the identification columns code_module and code_presentation, the table courses relates to the assessments, studentRegistration, vle, and studentInfo. Last but not least, the vle connects to studentVle via the id_site and the assessments table relates to studentAssessment using the id_assessment [17]. Each table is described in detail in the following subsections. Table studentInfo: Table 1 includes demographic data about the students as well as their performance on each module they took. There are 32 593 rows in it.

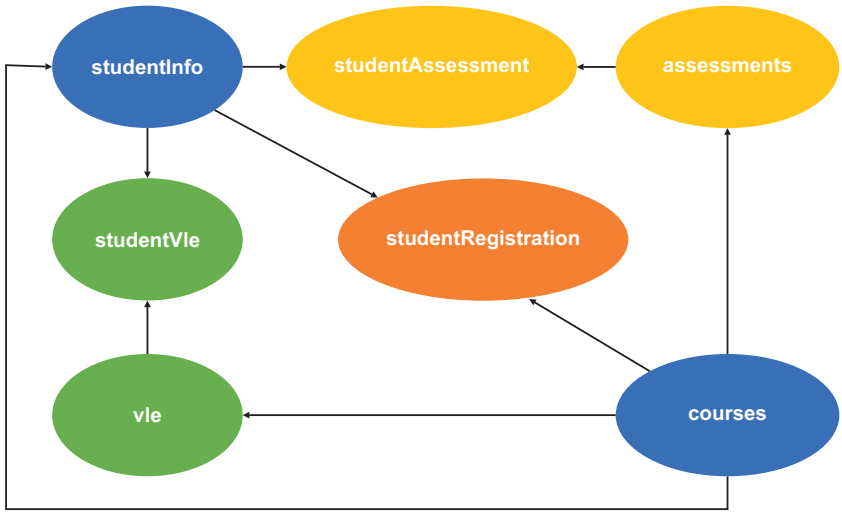

Fig. 3. Detailed dataset architecture.

Table 1. The variables associated to studentInfo.

Attribute	Description and Type
code_module	The student's registered module's identification code. (Nominal)
code_presentation	presentation identification number used to register the student for the module. (Nominal)
id_student	the distinctive student ID number. (Numeric)
gender	student's gender. (Binary)
region	the area where the student resided during the module's presentation. (Nominal)
highest_education	the greatest degree of education a student has at the time of the module presentation. (Nominal)
imd_band	the IMD band of the location in which the student was residing at the time of the module's presentation. (Nominal)
age_band	a band of student's age. (Nominal)
num_of_prev_attempts	the quantity of attempts the learner has made at this module. (Numeric)
studied credits	the total credits earned for the modules the student is enrolled in. (Numeric)
Disability	whether the student has announced a disability is indicated. (Binary)
final_result	final result of the student for the module presentation. (Nominal)

Table **studentAssessment**: The outcomes of the students' assessments are listed in the Table 2. No results are recorded if the student does not turn in the assessment. There are 173 912 rows in it.

Table **assessments**: The module-presentation assessments are listed in Table 3. Every presentation often includes an assortment of assessments before the final exam. There are 206 rows in the table.

Table **studentVle**: Information regarding students' interactions with the VLE is contained in the studentVle table. There are 10 655 280 rows in it, and the variables are as follows: code_module, code_presentation, id_student, id_site, date, and sum_click (e-number of interactions a student had with the material).

Table 2. The variables associated to studentAssessment.

Attribute	Description and Type
id_assessment	the identification number for the assessment. (Numeric)
id_student	the distinctive student ID number. (Numeric)
date_submitted	the day the assessments are due. (Numeric)
is_banked	the status flag stating that the assessment outcome was carried over from the previous presentation. (Binary)
score	the result of this assessment for the student. There is a range of 0 to 100. A score of less than 40 is considered a failure. The scores fall between 0 and 100. (Numeric)

Table 3. The variables associated to assessments.

Attribute	Description and Type
id_assessment	the identification number for the assessment. (Numeric)
code_module	The student's registered module's identification code. (Nominal)
code_presentation	presentation identification number used to register the student for the module. (Nominal)
assessment_type	A specific assessment. There are three different sorts of exams: the final exam (Exam), tutor marked assessments (TMA), and computer marked assessments (CMA). (Nominal)
date	Information regarding the deadline for the assessment. (Numeric)
weight	The assessment's weight. Exams are typically assessed independently and given a weight of 100%. (Numeric)

Table **vle**: The materials that are accessible through the VLE are included in the vle table. These are frequently PDF files, HTML pages, etc. There are 6 364 rows in it, and the variables are as follows: code_module, code_presentation, id_site, activity_type, week_from, and week_to.

3.2. Data preprocessing

Before using classification techniques, the dataset must be prepared by pre-processing the data. It is crucial to remember that the reliability and quality of the information at hand will immediately impact the result of this work. To rule out any abnormalities, a careful investigation of the variables and their related values is done in this work. We used three main preprocessing tasks in the present study:

Feature selection: We carefully examine our dataset to find features that have a bigger influence on our output variable. To identify the suitable attributes, we applied a ranking algorithm.

Imbalanced data: When the number of instances in one class is much less than the number of instances in the other class, the data is unbalanced. As a result, the classifier takes more samples during the training phase from the classes with the highest number of instances. To take care of the issue of data imbalance, we used SMOTE.

Data transformation: Removing inconsistencies in the dataset through data transformation is a crucial step that makes it more suitable for data mining [18]. The majority of data mining algorithms only operate on numeric variables, therefore convert strings to numeric variables. As a result, it is necessary to convert non-numerical data into numerical variables. The most popular techniques involve encoding strings with values between [0 and $(N-1)$], where N is the amount of values.

It is interesting to include the performance in each assessment since it serves as a good gauge of the students' understanding of the material and because it determines the final evaluation grade. But because there are so many different courses, each with its own structure, it would be impossible to develop a feature for every test. We will construct one feature, which is the ultimate grade determined by the score, in order to include the assessments. Final exams will likewise be separated from the other assessments due to the differences in their status and inclusion in the final evaluation.

The student interaction stream with the content available for reference over the course of the term is included in the datasets related to the VLE (Virtual Learning Environment). We can determine a student's level of engagement with their subjects from this data, as well as whether they studied it thoroughly and how they used the material.

The studentInfo table contains a variety of information on the students, but the student's **final result** is the variable we are most interested in when developing our prediction model.

Since the data had multiple missing values, handling those missing values was done in the subsequent data preprocessing stage. The approach employed was to replace missing values with the most common data in that category rather than deleting them.

3.3. Methods and models employed

A data mining technique called classification is used to recognize, distinguish, classify, and understand items according to a predetermined class. A supervised machine learning technique called classification trains the machine by giving it instances. Therefore, during the training phase, the machine is trained by giving it a dataset to use as training data. This might also be considered as data analysis. The dataset is delivered to the cross validation operator using several classifiers, which stores the knowledge as models in the database.

For this work, there are numerous models for classification. In this study, four different types of models: logistic regression, random forests, Naive Bayes, and decision trees, are used. They are further discussed.

Decision tree: The process of modeling data in a tree-like form is called decision tree modeling. It is a prediction model in the form of a tree, with the root node at the top and the leaf node at the bottom, which represents the outcome of the data. The C4.5 algorithm or the CART algorithm can be used for modeling the decision tree. The items are categorized into predetermined classes using a decision tree. The decision tree is often referred to as a tree of classification when it is used for classification purposes. The decision tree can be further used to infer a "if, then" rule that will help analyze the data thoroughly and appropriately classify it into the correct classes [19].

The following are the formulas for the decision tree's learning rule and loss function:

Loss Function (Gini Impurity): The following formula is used to determine a decision tree node's Gini impurity:

$$\mathrm{Gini}(D) = 1 - \sum_{i=1}^{c} P_i^2,$$

where D is the dataset at the current node, P_i is the proportion of instances of class i in the node, and c is the number of classes.

Learning rule: To reduce the impurity, the decision tree learning rule iteratively divides the dataset according to its features. The Gini gain for a binary split is computed as follows:

$$\Delta\,\mathrm{Gini}(D, A) = \mathrm{Gini}(D) - \left(\frac{|D_1|}{|D|} \cdot \mathrm{Gini}(D_1) + \frac{|D_2|}{|D|} \cdot \mathrm{Gini}(D_2)\right),$$

where A is a candidate feature, D is the current dataset, D_1 and D_2 are the subsets of D resulting from the split based on the threshold of feature A. The data is divided based on the feature and threshold that the computer determines will maximize the Gini gain. The decision tree is made by using this procedure recursively.

Random forest: A classification ensemble learning technique called random forest builds numerous unpruned classification trees during the training phase using the bootstrap sampling approach on the training data. The mean from all unpruned tree classifications in the phase of testing is found to provide the final predicted output for a randomly chosen feature [20]. Since a Random Forest is an ensemble of decision trees, its loss function is not clearly defined. But we can talk about the learning rule and the main ideas behind Random Forest training. Random Feature Selection: In a decision tree, a random subset of features is taken into account for each split. This adds character to the trees and strengthens the ensemble. Decision Tree Training: Every Random Forest decision tree is developed by

dividing the data according to a measure of impurity (such as Gini impurity) and recursively choosing features.

Logistic regression: By fitting data to a logistic function, the type of regression known as "logistic regression" can predict the probability that an event will occur [21]. Similar to other types of regression analysis, logistic regression uses a number of predictor variables that may be categorical or numerical [22].

The definition of the logistic regression concept is

$$h(\theta x) = g(\theta^T x),$$

when the definition of the function g is a sigmoid function:

$$g(z) = \frac{1}{1+e^{-z}}.$$

The sigmoid function has unique characteristics that produce values in the range $[0, 1]$.

The logistic regression cost function is as follows:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\log\left(h_\theta(x^{(i)})\right) - \left(1-y^{(i)}\right)\log\left(1-h_\theta(x^{(i)})\right)\right).$$

In order to determine the minimum of this cost function, we will utilize the built-in function in machine learning named fmin_bfgs, which, given a fixed dataset (of x and y values), determines the optimum parameters for the logistic regression cost function.

Logistic loss (cross-entropy loss): For binary classification, the logistic loss (cross-entropy loss) is provided by

$$J(\beta) = -\frac{1}{m}\sum_{i=1}^{m}\left(y_i\log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right),$$

where m is the number of training examples, y_i is the true class label (0 or 1) for the i-th example, $\hat{y}_i$ is the predicted probability that the i-th example belongs to class 1.

Learning rule (gradient descent): In order to minimize the logistic loss, the model parameters, or coefficients, are frequently updated using the gradient descent process. For every parameter β_j, the update rule is provided by

$$\beta_j = \beta_j - \alpha\frac{\delta_j}{\delta_{\beta_j}},$$

where α is the learning rate, and $\frac{\delta_j}{\delta_{\beta_j}}$ is the partial derivative of the loss with respect to β_j. For logistic regression, the partial derivative is calculated as

$$\frac{\delta_j}{\delta_{\beta_j}} = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}_i - y_i\right)x_{ij}.$$

Naive Bayes: Another supervised learning technique, as well as the statistical method for classification, is the Bayesian classification [23]. Assumes a fundamental probabilistic model, which allows for the principled capture of model uncertainty through the determination of outcome probabilities. The Bayesian classification's main benefit is its ability to deal with prediction issues. Let us look at a general probability distribution $P(x_1, x_2)$ with two possible values. Without reducing generality, the Bayes rule yields the following equation:

$$P(x_1, x_2) = P(x_1|x_2)\,P(x_2).$$

Similar, the following equation is obtained if there is a second class variable, c:

$$P(x_1, x_2|c) = P(x_1|x_2, c)\,P(x_2|c).$$

The following result is obtained from generalizing the situation with both variables to a conditional independence assumption for a set of variables x conditioned on a further variable c:

$$P(x|c) = \prod_{i=1}^{n} P(x_i|c).$$

Gaussian Naive Bayes: Loss Function: According to Gaussian Naive Bayes, the characteristics have a Gaussian, or normal, distribution. Given the class, the likelihood function is used to describe the probability of detecting a given feature value,

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right).$$

The negative log-likelihood is regarded as the loss function and is the log-likelihood that is frequently used:

$$J(\theta) = -\sum_{i=1}^{n}\sum_{j=1}^{k} I(y^{(i)} = C_j) \log P(x_i^{(i)}|C_j),$$

where n is the number of training examples, k is the number of classes, C_j represents the j-th class, θ includes the class priors $P(C_j)$, means μ_y, and variances σ_y^2.

Learning rule: Using maximum likelihood estimation, the parameters (means and variances) are determined based on the training data,

$$\mu_y = \frac{\sum_{i=1}^{n} I\big(y^{(i)} = C_j\big) x_i^{(i)}}{\sum_{i=1}^{n} I\big(y^{(i)} = C_j\big)},$$

$$\sigma_y^2 = \frac{\sum_{i=1}^{n} I\big(y^{(i)} = C_j\big)\big(x_i^{(i)} - \mu_y\big)^2}{\sum_{i=1}^{n} I\big(y^{(i)} = C_j\big)}.$$

4. Experimental setup

The primary goal of the study is to determine whether the explanatory (input) variables included in the model can predict the class (output) variable. The classification model is constructed using several types of algorithms, each of which employs a different categorization methodology. The Python programming language and its Integrated Development Environment (IDE) were used for implementing the models. Machine learning prediction models were implemented using the Scikit-learn package [24]. Cross validation (using 10 folds and applying the algorithm 10 times, each time using 9 folds to train it and 1 fold to test) and percentage split (using 2/3 of the dataset for training and 1/3 for testing) are the two testing methods that are applied to each classifier.

The main goal of the data mining project that has been presented is to predict student performance at the university using a set of attributes that reveal information about the students. The categorical target variable "final result" was chosen as the target variable in this case, or the concept to be learned by the data mining method. It has four values (categories): "Fail", "Pass", "Distinction", and "Withdrawn". We eliminate instances where a student has withdrawn their registration for a module. So we have three unique categories: "Fail", "Pass", "Distinction".

To achieve our objectives, we conducted various experiments. Our first goal was to predict student academic achievement. The second goal was to reduce the number of attributes. The final goal was to compare the classification accuracy of various classifiers.

Four common evaluation metrics: accuracy, recall, precision, and f-score are utilized in our tests to assess the performance of the classification algorithms. They are denoted as follows:

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

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

where the terms FN, FP, TP, and TN are False Negative, False Positive, True Positive, and True Negative, respectively.

5. Result

We evaluated, contrasted, and analyzed a dataset using four classifiers: logistic regression, random forests, Naive Bayes, and decision trees. All of the relevant attributes were used to evaluate all four classifiers. We employed two testing methods, the percentage split (using 2/3 of the dataset for training and 1/3 for testing) and the tenfold cross validation, which implies that the dataset was divided into 10 equal-sized subsets at random each time using 9 folds to train it and 1 fold to test. Table 4 displays the findings of our first experiment, which applied the percentage split for all features.

Table 4. Comparison of classifiers using all attributes.

Classifier	Accuracy	Precision	Recall	F1-Score
Logistic Regression	68.73%	51.74%	68.73%	56.45%
Random Forest	60.59%	55.39%	60.59%	57.51%
DecisionTree	54.20%	54.61%	54.20%	54.40%
Naive Bayes	53.46%	55.09%	53.46%	53.91%

Logistic Regression outperformed other classifiers in terms of accuracy rate, achieving 68.73%, which measures a classifier's efficacy. The fact that Random Forest is the winner in terms of precision with 55.39% demonstrates its predictive strength. Recall, which measures sensitivity, indicates that Logistic Regression also performs better with a score of 68.73%. Random Forest score better in F1-Score with 57.51% than the others classifiers. In the second experiment, we used algorithms for feature selection include the ranking methods, with each algorithm choosing a set of attributes. Ultimately, an attribute chosen by more algorithms is regarded as the best attribute. In our example, we chose features with the high frequency. Table 5 lists our top four features.

Table 5. Best four attributes and descriptions.

Attribute	Description
studied_credits	the total credits earned for the modules the student is enrolled in. (Numeric)
num_of_prev_attempts	the quantity of attempts the learner has made at this module. (Numeric)
exam_score	the final grade given by the score which is the result of the assessment for the student. (Numeric)
sum_click	the average amount of clicks per material. (Numeric)

On this reduced dataset, four classifiers are again used with the percentage split and also the 10 fold cross-validation. Table 6 shows the outcome of this reduced dataset. Based on the accuracy, precision, and Recall, Random Forest is the top classifier with 86.68% for accuracy. Also, F1-Score which is combines the precision and recall scores of a model, revealed that Random Forest score better with 86.71% than the other classifiers. Additionally, when we compare Tables 6 and 6, we can find modest increases and decreases in values.

Table 6. Comparison of classifiers using best features.

Classifier	Accuracy	Precision	Recall	F1-Score
Logistic Regression	86.42%	86.27%	86.42%	86.07%
Random Forest	86.68%	86.74%	86.68%	86.71%
DecisionTree	83.59%	83.72%	83.59%	83.64%
Naive Bayes	85.74%	86.48%	85.74%	85.93%

Table 7. Comparison of classifiers using the percentage split and the 10 fold cross-validation.

Classifier	Accuracy	
	Percentage split	10 Fold cross-validation
Logistic Regression	86.42%	87.48%
Random Forest	86.68%	85.84%
DecisionTree	83.59%	83.32%
Naive Bayes	85.74%	85.62%

In Table 7, we compare the accuracy results for all classifiers using the percentage split and the 10 fold cross-validation. According to the percentage split in Figure 4, Random Forest is the top classifier with 86.68% for accuracy, but for the 10 fold cross-validation, Logistic Regression is the greatest classifier with 87.48%.

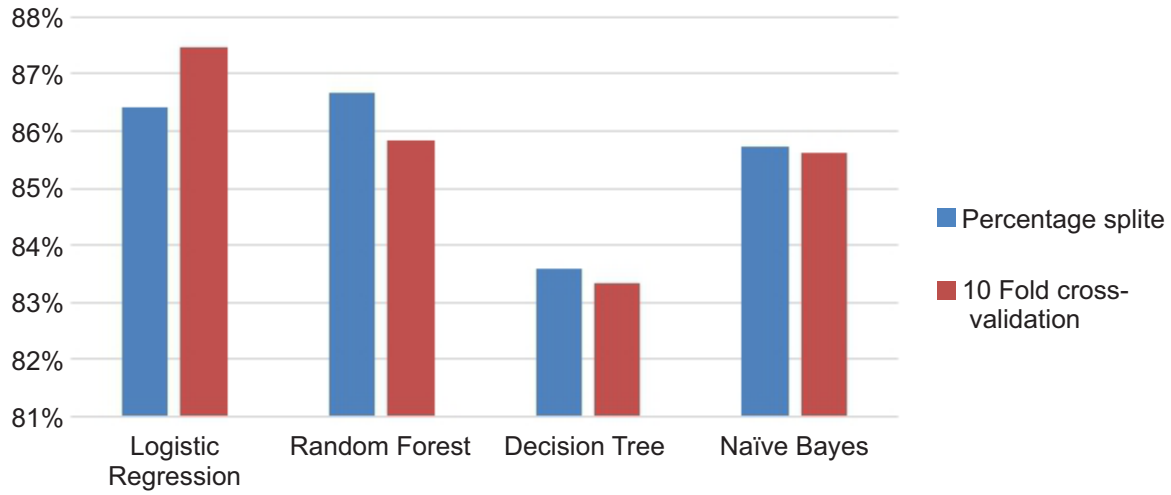

Fig. 4. Accuracy performance as indicated by the percentage split and 10-fold cross-validation.

Furthermore, we looked at our data to find out what causes students to lose their academic standing as a result of poor performance. We discovered that a student's poor performance was caused by their lack of engagement with the online platform.

6. Conclusion

This research aimed to forecast and evaluate students' academic performance using Data Mining techniques. Logistic regression, random forest, Naive Bayes, and decision tree were the four approaches employed in this study. All of these methods were used on student data taken from the Open University Learning Analytics dataset (OULAD), which includes a portion of the OU student data. Both demographic information about the students and information about their VLE interactions are included. Four classification models were created in this study to predict students' academic success. The outcome demonstrates that the Naive Bayes classifier performs better than the other two classifiers by achieving an overall prediction accuracy of 93%. This study helps instructors identify students who are likely to fail the course early on. These students can benefit from additional instructor support, which can help improve their academic performance. Future work will focus on more thorough exploratory data analysis and integrating data preparation techniques with machine learning methods, particularly deep learning algorithms.

[1] Istanbullu A., Karahasan M. A new student performance analysing system using knowledge discovery in higher educational databases. Computers & Education. **55** (1), 247–254 (2010).

[2] Mohamad S. K., Tasir Z. Educational Data Mining: A Review. Procedia – Social and Behavioral Sciences. **97**, 320–324 (2013).

[3] Khoroshchuk D., Liubinskyi B. Machine learning in lung lesion detection caused by certain diseases. Mathematical Modeling and Computing. **10** (4), 1084–1092 (2023).

[4] Al-Radaideh A. Q., Al-Shawakfa M. E., Al-Najjar I. M. Mining Student Data Using Decision Trees. The 2006 International Arab Conference on Information Technology (ACIT'2006). 1–5 (2006).

[5] Hand J. D. Principles of Data Mining. A Bradford Book. The MIT Press (2001).

[6] Mallouk I., Abou el Majd B., Sallez Y. A generic model of the information and decisional chain using Machine Learning based assistance in a manufac-turing context. Mathematical Modeling and Computing. **10** (4), 1023–1036 (2023).

[7] Shahiri M. A., Husain W., Rashid A. N. A Review on Predicting Student's Performance Using Data Mining Techniques. Procedia Computer Science. **72**, 414–422 (2015).

[8] Christian M. T., Ayub M. Exploration of classification using NBTree for predicting students' performance. International Conference on Data and Software Engineering (ICODSE). 1–6 (2014).

[9] Nguyen Thi Ngoc Hien, Haddawy P. A decision support system for evaluating international student applications. 2007 37th Annual Frontiers In Education Conference – Global Engineering: Knowledge Without Borders, Opportunities Without Passports. F2A-1–F2A-6 (2007).

[10] Arsad M. P., Buniyamin N., Manan A. J. A neural network students' perfor-mance prediction model (NNSPPM). IEEE International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA). 1–5 (2013).

[11] Romero C., López I. M., Luna M. J., Ventura S. Predicting students' final performance from participation in on-line discussion forums. Computers and Education. **68**, 458–472 (2013).

[12] Aldowah H., Al-Samarraie H., Fauzy M. W. Educational data mining and learn-ing analytics for 21st century higher education: A review and synthesis. Telematics and Informatics. **37**, 13–49 (2019).

[13] Mishra T., Kumar D., Gupta S. Mining Students' Data for Prediction Performance. 2014 Fourth International Conference on Advanced Computing & Communication Technologies. 255–262 (2014).

[14] Quadri N. M. M. Drop Out Feature of Student Data for Academic Performance Using Decision Tree Techniques. Global Journal of Computer Science and Technology. **10** (2), 2–5 (2010).

[15] Natek S., Zwilling M. Student data mining solution–knowledge management system related to higher education institutions. Expert Systems with Applications. **41** (14), 6400–6407 (2014).

[16] Gray G., McGuinness C., Owende P. An application of classification models to predict learner progression in tertiary education. 2014 IEEE International Advance Computing Conference (IACC). 549–554 (2014).

[17] Kuzilek J., Hlosta M., Zdrahal Z. Open University Learning Analytics dataset. Scientific Data. **4** (1), 170171 (2017).

[18] El-Hafeez A. T., Omar A. Student Performance Prediction Using Machine Learning Techniques. In Review, preprint (2022).

[19] Al-Radaideh A. Q, Al-Shawakfa M. E, Al-Najjar I. M. Mining Student Data Using Decision Trees. The 2006 International Arab Conference on Information Technology (2006).

[20] Jindal R., Borah D. A Survey on Educational Data Mining and Research Trends. International Journal of Database Management Systems. **5** (3), 53–73 (2013).

[21] Marrakchi N., Bergam A., Fakhouri H., Kenza K. A hybrid model for predicting air quality combining Holt–Winters and Deep Learning Approaches: A novel method to identify ozone concentration peaks. Mathematical Modeling and Computing. **10** (4), 1154–1163 (2023).

[22] Dreiseitl S., Ohno-Machado L. Logistic regression and artificial neural network classification models: a methodology review. Journal of Biomedical Informatics. **35** (5–6), 352–359 (2002).

[23] El Naqa I., Murphy J. M., Martin J. What Is Machine Learning? Machine Learning in Radiation Oncology. 3–11 (2015).

[24] Pedregosa I., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay É. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. **12** (85), 2825–2830 (2011).

Прогнозування успішності студентів та моделювання за допомогою методів аналізу даних

Джедідай Ю.[1], Ібріз А.[1], Бенсліман М.[1], Хачмуд А.[1], Тмімі М.[1], Хаджіуі Ю.[1], Рахалі М.[2]

[1]*Innovative Technologies Laboratory, EST, Університет Сіді Мохамеда Бен Абделла, Фез, Марокко*
[2]*ENSA, Університет Сіді Мохамеда Бен Абделлаха, Фез, Марокко*

У навчальних закладах та університетах проблему перерв у навчанні можна вирішити за допомогою електронного навчання. У результаті останнім часом ця галузь привернула значну увагу. У цьому дослідженні ми застосували чотири методи машинного навчання для прогнозування академічного прогресу студентів: логістичну регресію, дерева рішень, випадкові ліси та наївний Баєсів класифікатор. Набір даних Open University Learning Analytics (OULAD), який містить підмножину даних про студентів OU, був джерелом даних студентів для всіх цих методів. Набір даних містить інформацію про взаємодію студентів із віртуальним навчальним середовищем (VLE) та їхні демографічні дані. Сьогодні університети часто використовують методи інтелектуального аналізу даних для аналізу наявних даних і отримання знань, які допомагають у прийнятті рішень. Ми використали відсотковий розподіл і 10-кратну перехресну перевірку, щоб виміряти та порівняти ефективність прогнозування чотирьох класифікаторів. При застосуванні відсоткового розподілу класифікатор Naive Bayes виявився кращим, ніж інші класифікатори, досягнувши загальної точності прогнозування 93%. Це дослідження має на меті допомогти вчителям підвищити успішність учнів.

Ключові слова: *прогнозування успішності студента; великі дані; аналіз даних в освіті (EDM); машинне навчання; класифікатори.*