Information Sciences Letters
An International Journal

http://dx.doi.org/10.18576/isl/120102

A Predictive Model for Student Performance in Classrooms using Student Interactions with an eTextbook

*Ahmed Abd Elrahman**, *Taysir H. A. Soliman, Ahmed I. Taloba and Mohammed F. Farghally*

Information System Department, Faculty of Computers and Information, Assiut University, Egypt

Received: 5 Apr. 2022, Revised: 12 Jun. 2022, Accepted: 15 Jun. 2022
Published online: 1 Jan. 2023

Abstract: With the rise of online eTextbooks and Massive Open Online Courses (MOOCs), a huge amount of data has been collected related to students' learning. With the careful analysis of this data, educators can gain useful insights into their students' performance and their behavior in learning a particular topic. This paper proposes a new model for predicting student performance based on an analysis of how students interact with an interactive online eTextbook. By being able to predict students' performance early in the course, educators can easily identify students at risk and provide a suitable intervention. We considered two main issues: the prediction of good/bad performance and the prediction of the final exam grade. To build the proposed model, we evaluated the most popular classification and regression algorithms. Random Forest Regression and Multiple Linear Regression have been applied in Regression. While Logistic Regression, decision tree, Random Forest Classifier, K Nearest Neighbors, and Support Vector Machine have been applied in classification. Based on the findings of the experiments, the algorithm with the best result overall in classification was Random Forest Classifier with an accuracy equal to 91.7%, while in the regression it was Random Forest Regression with an R^2 equal to 0.977.

Keywords: Students' Performance, eTextbook, High Risk Students, Drop Out of Course, Classification, Regression, Random Forest Algorithm.

1 Introduction

The worldwide spread of COVID-19 has had a significant impact on human life both in the current era and in the coming years. The education sector has been greatly affected by its spread, as it depends on the regular and continuous presence of all members of the educational process within educational institutions. So, it has become necessary to find alternative learning methods rather than traditional classroom learning methods. Online eTextbooks and Massive Open Online Courses (MOOCs) have become the best alternative to learning ways. There is more potential in Technology-enhanced environments than traditional classroom learning.

When teaching a particular course using an eTextbook, the interaction between students and instructors may be limited and there may be no face-to-face interaction. Therefore, it may be difficult for educators to know the performance of their students, especially low-risk students with learning disabilities, and who may leave the course. So, the prediction of students' performance at an early stage in the course might be helpful in detecting those students with a high risk of failing the course. An early prediction may serve as an active tool for changing educators' practices and issuing an awareness to assist students to back on the right track. In the discipline of learning analytics, the early prediction of student performance is a significant task. We think that it will improve academic retention and performance.

An early prophecy enables students to take the required stages to evade poor performance and enhance their own test achievements ahead of time. This early prediction not only alerts students to their poor performance but also gives them plenty of bases to maximize their academic performance [1].

Our study focused on a CS2 course. This course was taught at a large public research institution using the OpenDSA eTextbook infrastructure [2,3]. OpenDSA is an infrastructure to create eTextbooks including

* Corresponding author e-mail: ahmedabdo@aun.edu.eg.

interactive visualizations and exercises with automated feedback. OpenDSA gathers log data for all user interactions that happen on an OpenDSA module.

Computer science students tend to have high drop-out rates as compared to other disciplines. According to an article published in Irish Times [4]

• After their first year of college, in all institutes of technology, about one-third of computer science students drop out.

According to [5]:-

• 30-50 % of CS1 students at the Helsinki University of Technology are dropping out of the course.

• The difficulty of the subject was one of the compelling reasons.

According to this article [6], there are various reasons why students drop out of courses, such as expensive tuition, not being prepared scholastically, being unhappy with the college, dissuading surroundings, selecting the wrong topic, academic inadequacy, and conflict with work and family responsibilities. There may be some students who are affected by one or more of these reasons, but it is clear that those students require extra effort to thrive in their courses. Instructors should be able to identify these students early in the course to assist them in improving their performance.

In [7], A questionnaire was sent to computer science educators asking them to list topics that they believe are important for students to learn as well as topics that are difficult to learn (for students) and difficult to teach (for instructors). Based on the findings of this questionnaire, the educators found that the five most important topics which are hard to learn are pointers, recursion, polymorphism, memory allocation, and parameter passing. While they found that Recursion, pointers, error handling techniques, and polymorphism were the most difficult topics to teach. Most of these topics were taught in the CS2 course.

According to the conclusions of the questionnaire of [7], many of the topics in the CS2 course are difficult to learn and teach. Many students may find difficulty in learning and understanding these topics, and they may not report their difficulties in learning these topics to their educators. Due to the difficulty of most topics of the CS2 course and based on the paper [5], which mentioned that the difficulty of the course was one of the compelling reasons that made students drop out from the CS1 course, many students may resort to dropping out of this course. Therefore, a method must be devised to aid educators in learning about each student's performance in this course, and it is preferable that this knowledge become at an early stage of the course. Our study intends to build a model for predicting student performance based on their interactions with an eTextbook. This predictive approach has the advantage that it can be used at the beginning of the semester, if necessary, by instructors to communicate their concerns to students when there are signs of hazard. As a result, those students graduate on time and without having to repeat a semester, and they are well-prepared to succeed in their subsequent studies. This early prediction may be helpful to instructors to know early feedback to each student in the course, particularly low-risk, students with learning disabilities who need special attention, and maybe consider dropping out of this course. This feedback may assist instructors in providing appropriate warnings or advising to these students providing more attention to them to enhance their performance and trying to keep students from dropping out of the course. It may allow educators to help these students, and they may succeed in helping students to increase their academic performance and in reducing the falling ratio.

The structure of this paper is as follows: In Section 2, previous works relating to students' performance is reviewed. Section 3 presented the data set description. Section 4 discusses the outcomes of various data mining approaches. Section 5 presented the Conclusion. Section 6 presented the future work.

2 Related Work

There are various approaches to forecasting student performance, but data mining techniques are one of the most well-known and significant. The most significant techniques in data mining are classification and regression. The majority of researchers utilized them to forecast student performance.

In [8], a study was conducted to predict students' grades in their work and their results (pass/fail). To predict the students results, a classification model was utilized, while a regression model was used to predict the grades. For classification, decision trees and SVM were used, and for regression analysis, SVM Random Forest and AdaBoost.R2 were used. In this study, the classification model was shown to be capable of extracting useful patterns, while regression methods failed to prevail over a simple baseline.

A study has been conducted depending on the performance of students by choosing students from various institutes of Dr.R.M.L. Awadh University, Faizabad, India by using Bayes Classification on Category, Language, and Background Qualification. The goal of this study is to determine if incoming students will perform or not, as well as to determine which students who require special attention in order to lower the falling rate [9].

According to a study published in [10], a decision tree model was used to determine the final mark for students enrolled in a C++ course at Yarmouk University in Jordan. Three classification methods were used: ID3, C4.5, and Naive Bayesian. The results showed that the decision tree model outperformed the other models in terms of prediction.

A case study has been conducted in [11] the students' data was used to analyze their learning in order to forecast their outcomes and warn students who could be in danger before their final exams.

In [12], a machine learning method has been used to enhance the prognosis results of academic achievements in real-world case studies. Three methods have been used to resolve the problem of class imbalance all of which gave positive results. After balancing the datasets, both cost-sensitive and insensitive learning algorithms were used, including SVM for small datasets and a Decision tree for large datasets.

In [13], a design for student careers is developed. This paper described various methods based on sequential and clustering pattern algorithms to propose solutions for enhancing student performance and exam schedule.

According to a study published in [14], data mining techniques have been used to investigate students' intellectual performance. The primary objective of this study is to assess the student's performance across a variety of measurement categories.

In [15], classification-based techniques have been utilized to predict students who are slow learners. Five classification techniques have been used: Multilayer Perceptron, Naive Bayes, SMO, J48, and REP Tree to analyze and test the output dataset. According to the findings of this study, it was found that Multilayer Perception outperforms all other classifiers.

To improve students' performance at an early stage, a study is made in [16], which may predict student's performance at an early stage and provide early warning to these students. Three well-known single classification algorithms, C4.5, CART, and LGR have been used to design this system.

In [17], a model is developed for predicting student achievement based on students' personal, pre-university, and university functional properties, with the neural network obtaining the highest accuracy, followed by the decision tree classifier and the kNN model.

The authors of the paper [18], presented a study in which they applied data mining techniques on educational data by analyzing students' academic performance. They used the decision tree approach to identify dropouts and students who require further assistance, making it easier for instructors to issue warnings or advice.

The following sections will describe the OpenDSA system in brief detail, the Data set that was used in our work with their descriptions, and Finally, the data mining algorithms with the results of their experiments were mentioned.

3 Materials and Data Set Description

During fall 2020, OpenDSA was used as the main eTextbook to teach CS2 courses in a large public research institution. A module in OpenDSA represents a single topic or part of a typical lecture, such as a single sorting algorithm and it is considered the most elementary functional unit for OpenDSA materials [19].

Every module is a full unit of instruction that usually contains algorithm visualizations (AVs), interactive assessment activities with automated feedback, and textbook quality text. Modules are organized into chapters in the same way that traditional paper books are organized. One of the most important OpenDSA exercises is "Algorithm simulations". These exercises ask the students to handle a data structure in order to demonstrate how an algorithm works, such as clicking on proper nodes in a tree or clicking to swap elements in an array. The JavaScript Algorithm Visualization (JSAV) [20] library is used to build this type of exercise.

OpenDSA contains multiple types of exercises and events. The next section describes exercise types and events.

3.1 OpenDSA's events and exercises types

In this work, about 200 different types of events are recorded by the OpenDSA system. The following is a summary of them:

- Interactions with sign-up and sign-in .
- Interactions with Static Content (such that when a module page is loaded by a student, when a student uses the navigation menu to navigate to another page).
- Reactive Activities Interactions (when a student needs to go ahead a slideshow).
- Interactions between Assessment Activities (such that when an exercise is loaded or when the answer is submitted by students, etc.).
- Interactions in the grade book (when a student need to know his score so he loads the gradebook page).

There is a timestamp for each event. To determine the relationship between the use of OpenDSA and student performance, log data and performance data (student grades written "etest") was utilized. The visualization was frequently used in the classroom as a lecture assist by the instructor, who utilized OpenDSA as the primary course material.

OpenDSA provides three types of exercises: proficiency exercises, simple questions, and programming exercises. For Proficiency exercises, this type of exercise is an algorithm simulation exercise. It requires students to simulate the behavior of a given algorithm in order to ensure that they understand how it works. This type of exercise was pioneered in the TRAKLA2 system [21].

Simple Questions are made up of a variety of OpenDSA system question categories, including true/false, multiple-choice, and fill-in-the-blank, and they don't typically take a long time to finish. To store and present the simple exercises, OpenDSA used the Khan Academy framework [22].

All exercises are assessed automatically and comments were provided to the students. Students can work on exercises more than once until they obtain credits.

Every exercise contains a different number of questions. Each student can interact with any exercise in any module at any time during the course of study. There

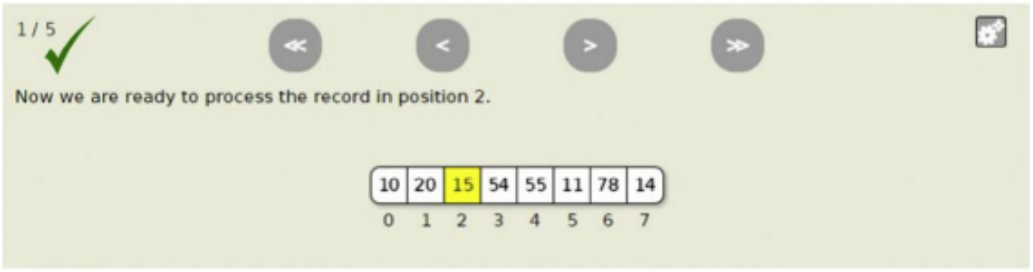

Fig. 1: An example of an OpenDSA Slideshow.

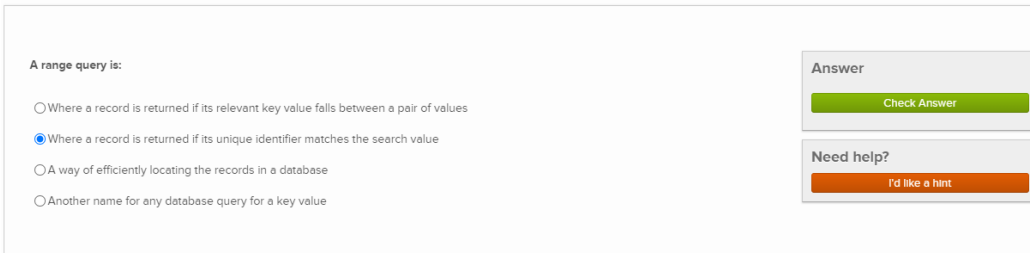

Fig. 2: Example of an OpenDSA Simple Question.

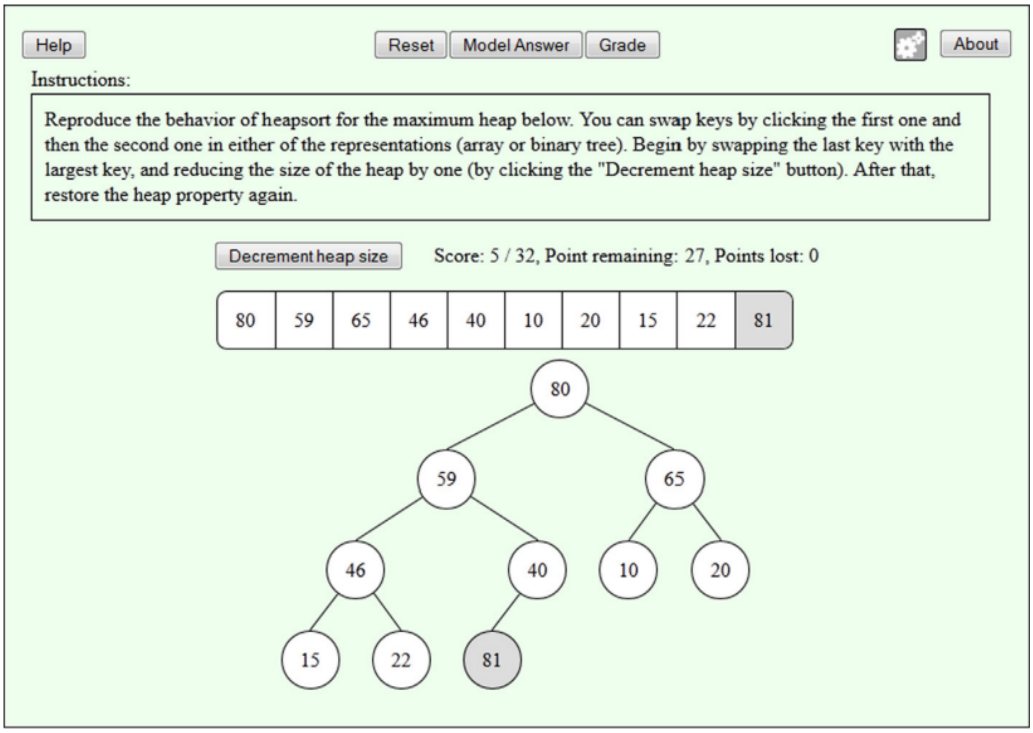

Fig. 3: Example of JSAV proficiency exercises for Heap Sort.

is a Submit button for each question. If the student completes the question, he presses on it, but if he does not click on it, an attempt will be counted for him only for this question. Each time the student interacts with a question, he can solve it, whether in a right or wrong way. In this case, a grade is scored for him, and it is possible for the student to attempt only, and in this case, no points will be scored. In both cases, he can ask for a hint.

Slideshows is a type of interactive content provided by OpenDSA. Slideshows show a sequence of stages that animate the behavior of an algorithm. Slideshows were created using JSAV: the JavaScript Algorithm Visualization Library [20]. **Fig 1** shows an example of an OpenDSA slideshow. Standard controls, as illustrated in **Fig 1**, allow the user to advance the slideshow by one slide, back up one slide, go back to the beginning, or leap to the finish of the slideshow.

As shown in **Fig 2** and **Fig 3**, for each Proficiency exercise, the student can ask for a model answer or reset the exercise. For each Simple exercise, the student can ask for a hint or check his or her answer for this question. A student can attempt to complete each exercise category many times. So, we took the maximum number of attempts for each question.

3.2 Data set Description

We collected data from 200 students and analyzed it. For each student, we obtained the following characteristics. These characteristics were utilized to create the model.

1. **PE-total-time**: Total time in seconds which a student spent in solving or trying to solve proficiency exercises.
2. **PE-total-attempts**: The total number of attempts a student made to complete proficiency exercises.
3. **PE-reset**: How many times did a student reset their proficiency exercises.
4. **PE-model**: For proficiency exercises, the total number of times a student showed the model answer.
5. **PE-exercise**: The total number of proficiency exercises did each student solve .
6. **SS-total-time**: The Total time in seconds which a student was taken in viewing slideshows.
7. **SS-total-visit**: A student's total number of slideshows seen; a student can watch the same slideshow many times.
8. **Slide**: The total number of unique slideshows which a student has viewed.
9. **Interaction**: The total number of interactions that a student did.
10. **Total-time**: The total time in seconds has been spent by the student in dealing with the whole eTextbook.
11. **Total-attempts**: A total number of times did a student try to solve a Simple exercise.
12. **Total-hints**: The total number of hints a student uses during solving Simple exercises.
13. **gaming**: The total number of pages reloads for each student.
14. **exercise**: The Total number of exercises correctly completed.
15. **etest**: The final exam degree for the student.

4 Data Mining Techniques and Results

Many application domains have used data mining techniques, including banking, fraud detection, and telecommunications [23]. Data mining approaches have recently been utilized to improve and evaluate higher education problems.

In learning environments, the capacity to determine a student's performance is critical. The utilization of Data Mining is a very promising approach for achieving this goal [24]. Data mining techniques are applied to large amounts of data to uncover hidden patterns and relationships that aid decision-making.

The goal of our study is to predict (before the final exam) the performance (good/bad) and the final exam grade of students in one of the undergraduate data structures courses at a large public research institution at an early stage of course. This early prediction will aid in the identification of low-risk students at any point during the course, not only at the end. It aims to help these students to overcome the challenges they face in the learning process. It helps students work on their weaknesses to improve their performance and obtain good grades in their final exams. Furthermore, the findings will aid teachers in revising their instructional practices to improve student learning and prevent them from dropping out of this course.

To achieve our goal, we address two problems: the first is the prediction of student performance and the second is the prediction of students' final score in the final exam. We used two data mining approaches to achieve our goal. The first one is regression analysis which has the purpose of predicting the final degree for the exam. While the second one is Classification which has the purpose of predicting the performance for every student whether the performance will be good or bad. **Fig 4** shows the steps of data mining approaches that used in this study.

4.1 Data Preprocessing

Prior to implementing a data mining technique, data preprocessing turns the original data into a form that can be utilized by a specific data mining algorithm. Data preprocessing entails a variety of tasks. Data cleaning, feature selection, and data transformation are all steps in the data preprocessing process[25].

4.2 Data Cleaning

It is regarded as one of the most crucial data preprocessing steps. Data Cleaning, often known as

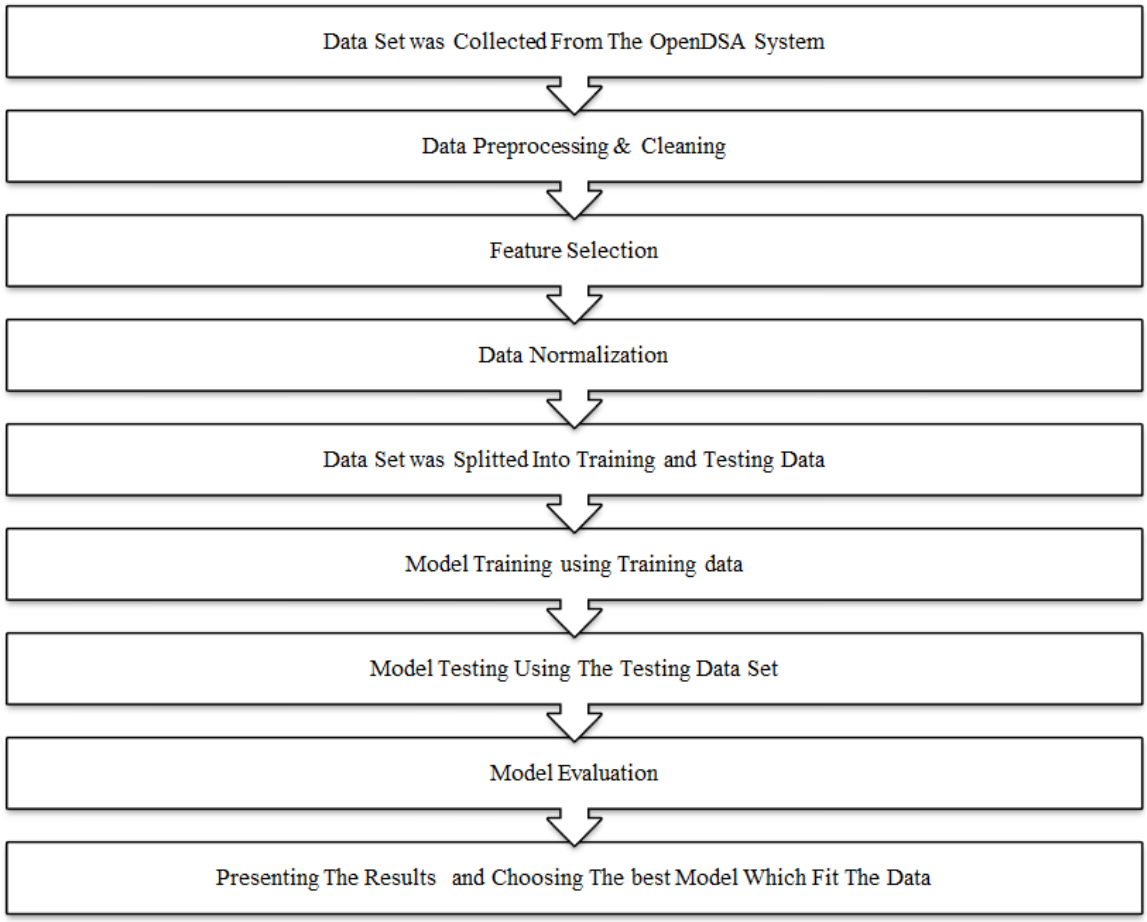

Fig. 4: Steps of data mining approaches.

scrubbing, is the process of discovering and removing errors and inconsistencies from data in order to enhance data quality. The purpose of data cleaning is to clean up the data by removing irrelevant items and missing values. After removing the missing records from the dataset, the dataset was reduced to 194 records. We have applied data cleaning in both regression and classification In the followings Sections, we will take about regression and classification algorithms in details.

4.3 Regression

Regression analysis is a group of statistical processes for appreciating the relationships between a dependent variable (often called the 'outcome') and one or more independent variables (often called features or predictors). As a result of the regression, the predictor is a continuous variable. In our study, the attribute "etest" is the dependent variable while the other attributes are the predictors. Linear regression is the most prevalent regressor in educational data mining [26]. However, regression trees are also very common. There are fewer regressors such as support vector machines and neural networks utilized in educational data mining than in other disciplines [26]. This is thought that more conservative algorithms are more successful in educating domains because of the high levels of noise and various explanatory elements [26]. Multiple linear regression and random forest regression have been utilized as regression techniques in our study.

4.3.1 Feature Selection

The main task in a data preprocessing area is feature selection. While eliminating redundant and irrelevant data is the goal of feature selection, selecting an adequate subset of features that can proficiently describe input data minimizes the feature space's size [27]. As a result, this process can have a considerable impact on the learning algorithm's efficiency.

Wrapper-based and filter-based techniques are the two types of feature selection methods used in supervised learning. Filter-based approaches are used to evaluate the

relationship between input variables and the target variable, and the scores from these evaluations are used to select (filter) which input variables will be included and will be used in the model.

For filter feature selection methods, there are many methods such as Pearson's correlation coefficient feature selection and Mutual information feature selection. About Pearson's correlation coefficient Feature Selection, A measure of how two variables change together is called correlation. For numeric predictors, the sample correlation statistic is the classical approach used to quantify each relationship with the outcome. To determine the correlation between two random variables, there are two broad categories to consider. The first, which is based on a linear correlation, the second is based on an information theory. Among these two measures, the linear coefficient of correlation is the most familiar. In general, linear correlation scores range from -1 to 1, with 0 representing no relationship between the two characteristics being analyzed. We're generally looking for a positive score for feature selection, the higher the positive value, the stronger the relationship and the more likely the feature will be used in modeling. Pearson's correlation coefficient "r" for two variables (X,Y) is given by:

$$r = \frac{\sum_i^n (X_i - \overline{X_i})(Y_i - \overline{Y_i})}{\sqrt{\sum_i^n (X_i - \overline{X_i})^2}\sqrt{(Y_i - \overline{Y_i})^2}} \quad (1)$$

A Pearson's correlation coefficient feature selection method was applied in this study in order to determine which features were most important while building a model that would predict students' final exam degrees. Pearson's Correlation coefficient Feature Selection scores are shown in **Fig 5**. Features were scored using this method. It has been determined that the first eleven features have a high influence on the outcome of the regression algorithms. As a result, those features were chosen, while others were not.

As shown in **Fig 5**, the PE-excercise feature got the highest score, then followed by total-attempts, PE-total-attempts, gaming, total-hints, PE-total-time, total-time, SS-total-visit, interaction, slide, PE-model, PE-reset, SS-total-time, and exercise, we selected the first eleven features which have the highest score while other ones are excluded.

4.3.2 Methodologies and Experiments Results

After the feature selection procedure, regression approaches were used. the following section introduced the two regression algorithms utilized in this work, followed by a discussion of the performance of regression models. Finally, we ended with the Regression models Results.

A. Multiple Linear Regression (MLR)

Multiple linear regression is a technique for understanding the relationship between two or more independent variables (or predictors) and one continuous dependent variable. In our study, the dependent (or outcome) variable is the 'etest' variable while the other features are the independent variables. Linear regression models generally have the form:

$$Y = a + \sum_{i=1}^{n} (b_i * x_i) + \in_i \quad (2)$$

In this case, "y" is the dependent variable (etest), "x" is the predictors, a is y-intercept (constant term), is the model's error term, and "n" is the number of independent variables. Based on the independent variables that are available, the regression equation was used to forecast students' final grades.

B. Random Forest Regression (RFR)

RFR is considered one of the ensemble machine learning approaches.RFR predicts an outcome from a set of predictors by creating multiple Decision Trees (DTs) and aggregating their results. By utilizing a unique bootstrap sample of the training data, each tree in a forest is created independently. Instead of using the best split among all predictors (as in bagging and bootstrapping [28]) for node splitting, RF chooses the best split from a randomly selected subset of predictors. The addition of this randomization reduces the association between trees in the forest so this operation will increase accuracy [29].

C. Regression assessments

The Regression Metrics approaches used to determine the effectiveness of the two regression algorithms utilized here are presented in this section. To evaluate the performance of the regression model, we used three different metrics: The root mean squared error, the mean absolute percentage error, and the coefficient of determination(R^2).

C1. Coefficient of Determination or R Squared (R^2)

(R^2 or r-squared) is a statistical measure in a regression model that predicts the proportion of the difference in the dependent variable that can be described by the independent variable. In other words, the coefficient of determination shows how well the data fit the model (goodness of fit). The most typical interpretation of R^2 is how well the regression model fits the observed data. A higher coefficient indicates a better model fit for the model. A model with an (R^2) score of 1

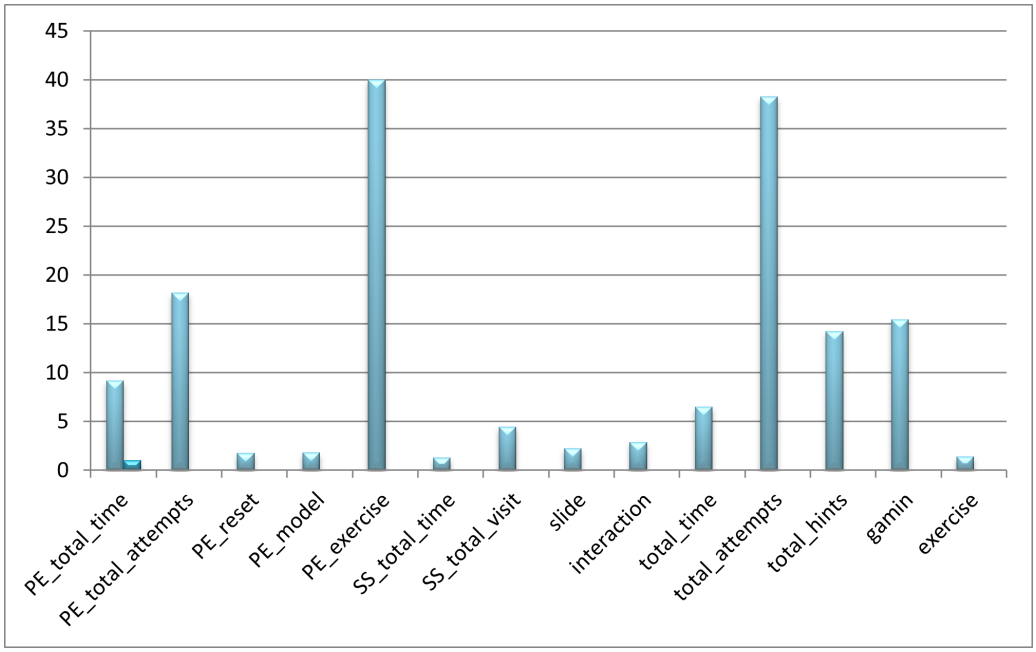

Fig. 5: The Features' scores based on Pearson's correlation coefficient selection method.

is perfect, whereas a score of 0 indicates that it will perform badly on an unknown dataset. This also means that the closer the r squared score is to 1, the better the model has been trained. The following formula is used to compute (R^2):

$$R^2 = 1 - \frac{SSE}{SST} \tag{3}$$

Where:

SSE is called the Total sum of squares, and

$$SSE = \sum_{i=1}^{n} (Yactual_i - Ymean)^2 \tag{4}$$

Where Yactual_i the original or observed y-value, Ymean is the mean of y-value.

SST is called the sum of squares due to regression, and

$$SST = \sum_{i=1}^{n} (Ypredicted_i - Ymean)^2 \tag{5}$$

Where Ypredicted_i is the y-value of regression, Ymean is the mean of y-value. The variation in the observed data is measured using the SSE. We can determine how well the model represents the data that was utilized in the model by using the SST

C2. Mean Absolute Percentage Error (MAPE)

MAPE is the mean or average of the absolute percentage errors of forecasts. The difference between the actual or observed value and the predicted value is called an error. And it is defined as a measure of predictive accuracy of a prediction technique in statistics. The better the forecast, the less the MAPE. The following formula shows how to compute the MAPE value.

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{A_i - P_i}{A_i} \right| \tag{6}$$

Where N is the number of items ,A_i is the actual value and P_i is the predicted value.

C3. Root Mean Squared Error (RMSE)

RMSE is a repeatedly utilized measure of the differences between values that the model predicted and the values observed. RMSE represents the square root of the discrepancies between anticipated and observed values, or the quadratic mean of these differences. We can compute RMSE using the following formula:

$$RMSE = \sqrt{\frac{\sum_i^n (predicted_i - actual_i)^2}{n}} \tag{7}$$

Where n = number of items, actual_i is the original or observed y-value, predicted_i is the predicted value.

D. Regression algorithms Results

As the data was divided in [30]. The data set was divided into ratios of 80% to 20%, having trained data

about 80% and testing data about 20%. Random student data was also utilized to predict grades to evaluate the model.

Table 1 Shows the results which we obtained after applying MLR and RFR to our dataset using selected features and with all features.

Table 1: R^2, MSE, MAPE for two regression algorithms used.

Algorithm	R^2	MSE	MAPE
MLR with All features	0.918	8.937	5.01
MLR with 11 selected features	0.922	8.759	4.89
RFR with All features	0.977	4.821	2.47
RFR with 11 selected features	0.976	4.799	2.27

As shown in **Table 1**, RFR has the highest R^2 value, as well as the lowest RMSE and MAPE values, followed by MLR. The R^2 for the two algorithms is very good when using all features or only the eleven selected features but both of them give better results when we use only the eleven selected features. As we said earlier , feature selection improves the algorithm's efficiency. results proved this statement.

4.4 Classification

Classification is one of the most often utilized and studied data mining techniques. Classification is a technique for predicting the class or category of a data object based on previously learned classes from a training dataset with known classes. As shown in **Table 2**, we divided the students into two groups based on their final exam's degree, which are:

Label: "good" for grades above 65% from the final exam degree.

Label: "bad" for grades less than or equal to 65% from the final exam degree.

Table 2: Selecting 2 class label according to student's grades.

Class	Grade	Number Of Students	Percentage
good	¿65%	181	93%
bad	¡=65%	14	7%

4.4.1 Dealing with Imbalanced Data

One of the most common preferred approaches to dealing with an imbalanced dataset is to resample the data. Undersampling and oversampling are the two most common ways for this. Oversampling techniques are preferred over undersampling techniques in most cases. The reason for this is that when undersampling the data, a lot of instances from data will be removed, and these removed instances may contain some important information.

4.4.1.1 SMOTE: Synthetic Minority Oversampling Technique

The method of oversampling (SMOTE) is used to create artificial samples of minorities [31]. SMOTE works by selecting the examples in the feature space that are close together, a line between the examples in the feature space is drawing, and drawing a new sample at a point along that line. We applied SMOTE to balance our data.' bad' class is the minority class in our dataset.

4.4.2 Feature Selection

We have noticed that Students' performance is impacted by all the features. All features were selected as parameters. So, in the classification algorithms, we used all features in building the predictive model.

4.4.3 Data Normalization

All features must be normalized after presuming that they are normally distributed in Bayesian and Parzen-window classifiers. As a result, in the decision-making process, each feature is given equal weight. The mean and standard deviation of the training data are used to normalize the data, assuming the data is Gaussian distributed. To normalize the training data, initially, we computed the mean and the standard deviations to each attribute, or column. Second, we normalized the training data using equation (8) .

$$x_scaled = \frac{X - mean}{sd} \tag{8}$$

Where mean is the average of x values, sd is the standard deviation of x values, X_scaled is the normalized value. The normalization has the benefits that ensure that each feature of the training dataset has a normal distribution with a mean of zero and a standard deviation of one.

4.4.4 Patterns Identification and Experiments Results

This section describes five classification algorithms which was used in this study ,then we discussed the Classifiers Evaluations for classification algorithms .finally we ended with the experiments results.

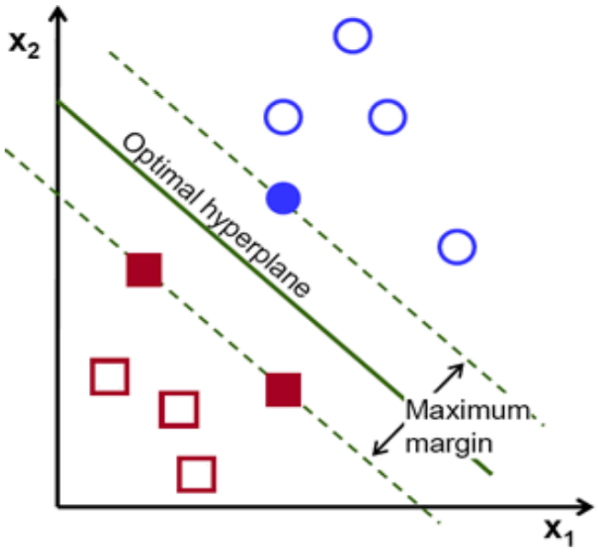

Fig. 6: Support Vector machine [32]

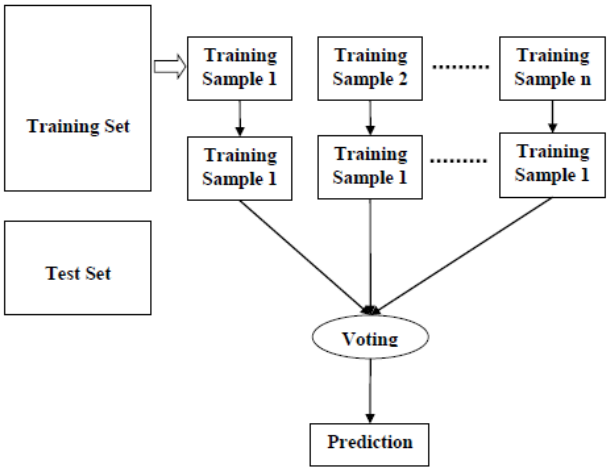

Fig. 7: Random Forest classifier

4.4.4.1 Pattern Identification

This process includes model training, pattern discovery, testing, and evaluation findings. After dividing the data set into testing and training sets, the prior data set is now ready for use. The classification approaches are used to build the model in the training set. The model is evaluated using a testing set. The results will then be evaluated. We have tested five classification algorithms in order to determine which one of them will work best for the prediction.

A. Support Vector Machine (SVM)

SVM is a Supervised Learning algorithm that can be used in classification and regression algorithms, SVM is a discriminative classifier officially defined by a separating hyperplane. In other words, given labeled training data, the algorithm outputs an optimal hyperplane that classifies new examples.

B. Logistic Regression (LR)

For many categorization problems, LR is one of the most basic machine learning algorithms and it is a supervised learning algorithm, utilized to predict the probability of a target variable. A dichotomous variable is considered the nature of the target or dependent variable, a dichotomous variable has only two possible classes. The nature of a binary dependent variable is that its value is single where the data is either 1 (which is for ”good” or ”yes”) or 0 (which is for ”bad” or ”no”). Mathematically, a logistic regression model predicts P(Y=1) as a function of X.

C. Random Forest Classifier (RF)

It is an efficient algorithm that provides more accurate results. Random forest is more efficient because it is the collection of several decision trees. It also prevents the issue of overfitting which is a major issue with decision trees. On the training data, this classifier used the bootstrap sampling method to create a large number of unpruned classification trees. Final predictions are made by using a randomized feature and arithmetic mean of all unpruned classification trees [33]. The following stages describe how Random Forest Algorithm works.

- –Stage 1 Start by selecting randomized samples from the dataset.
- –Stage 2 the second step is that a decision tree for each sample will be created. After that, the result of the prediction of each decision tree will be computed.
- –Stage 3 The third stage consists in voting for each expected result.
- –stage 4 at last, pick the result of the most voted prediction as to the result of the final prediction.

D. Decision Tree (DT)

A Decision tree is a flowchart-like tree structure, in which each internal node designates a test on a characteristic, each branch characterizes the test’s result, and each leaf node holds a class label. The CART algorithm or the C4.5 algorithms can be used to model the decision tree. The items are classified into predefined classes using the decision tree. When it is used to classify

data, it is referred to as a classification tree. The decision tree can induce the "If, then" rule to understand the data well and classify them into respective classes correctly [34].

E. K nearest Neighbors (KNN)

It is a supervised learning algorithm and one of the simplest machine learning techniques. The KNN algorithm assumes that the new case/data and available cases are similar and places the new case in the category that is most similar to available categories. The KNN algorithm stores all available data and classifies a new data point based on how similar they are to the existing data. This means when new data appears then KNN algorithm can readily classify it into a good group category.

4.4.4.2 Classifiers Evaluations

To assess the classification algorithms quality, four different measures were utilized: Accuracy, Precision, Recall, and F-Score [35,36]. measures were derived using Table 3, which shows the classification confusion matrix based on Equations(9),(10),(11),(12) respectively.

Table 3: Confusion Matrix.

		Predicted	
		Positive	Negative
Actual	Positive	True Positive(TP)	False Negative(FN)
	Negative	False Positive(FP)	True Negative(TN)

1-Accuracy : is the proportion of correct predictions made to the total number of forecasts made

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \tag{9}$$

2-Recall is the ratio of positive predictions that are right compared to the total number of positive examples.

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

3-Precision is the ratio of positive predictions that are right compared to the total number of predicted positives.

$$Precision = \frac{TP}{TP+FP} \tag{11}$$

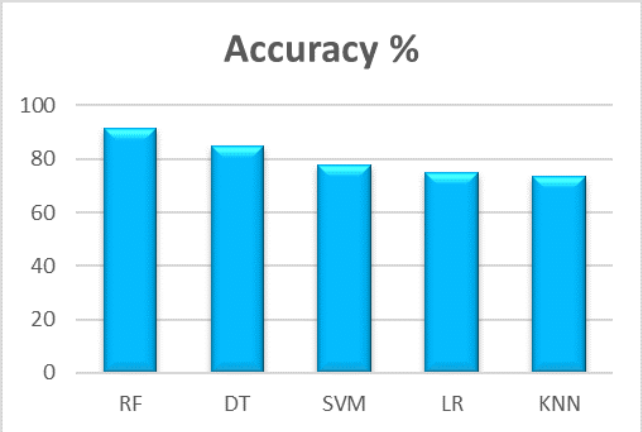

Fig. 8: The accuracy of different classifiers

4-F-score is a harmonic mean of the model's precision and recall.

$$F-score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{12}$$

4.4.4.2 Experiments Results for the classification algorithms

The data set was separated into ratios of 80% to 20%, having trained data about 80% and testing data about 20%. To predict the performance of each student, five different classification algorithms have been applied to our data. The next Figures and tables show the comparisons between different algorithms.

Table 4: Comparison of different classifiers based on accuracy

Algorithm	accuracy
RF	91.7%
DT	84.9%
SVM	78%
LR	75.7%
KNN	73.9%

Based on the accuracy of the five classifiers, it is evident that the Random Forest classifier beats the others, as shown in **Fig 8** and **Table 4**. It has a 91.7 % accuracy rate, followed by a decision tree with an accuracy rate of 84.9 %, a support vector machine with a 78 % accuracy rate, logistic regression with a 75 % accuracy rate, and lastly K Nearest Neighbors with a 73.9 % accuracy rate.

Fig 9 shows that, based on the precision of the five classifiers, the Random Forest classifier and Support Vector Machine have the highest precision among others classifiers.

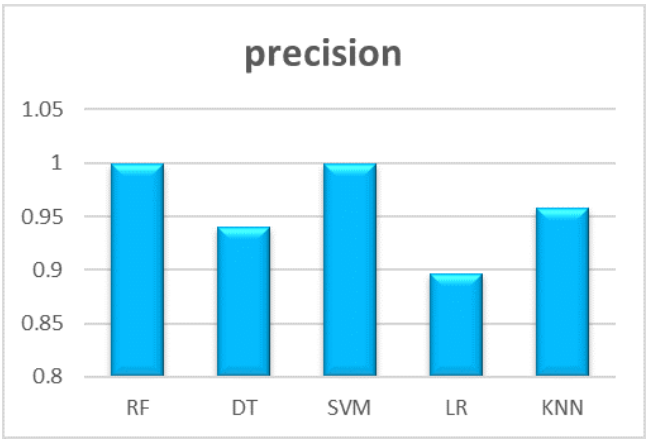

Fig. 9: Comparison of different classifiers based on precision

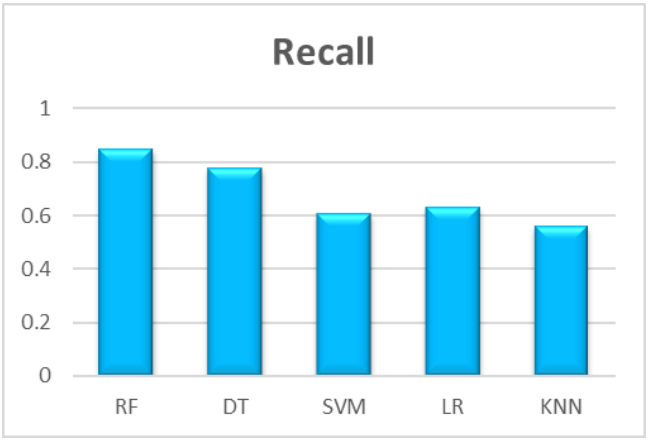

Fig. 10: Comparison of different classifiers based on Recall

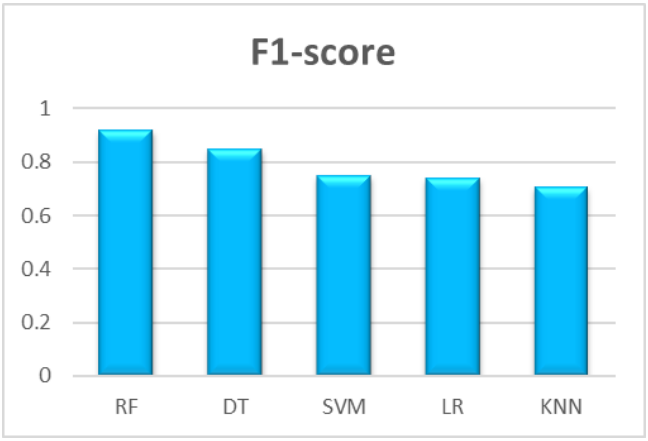

Fig. 11: Comparison of different classifiers based on F-score

Table 5: Comparison of different classifiers based on precision, Recall, F-score.

Algorithm	precision	Recall	F-score
RF	1.0	0.85	0.92
DT	0.94	0.78	0.85
SVM	1.0	0.6097	0.75
LR	0.8965	0.6341	0.7428
KNN	0.958	0.56	0.7076

Fig 10 shows that, based on the recall of the five classifiers, the Random Forest classifier has the highest recall among others classifiers. As shown in **Fig 11**, we observed that based on the f-score of the five classifiers it can be clear that the Random Forest classifier has the highest f-score among other classifiers.

4.4.5 Discussion of classification Results

Random Forest classifier, K-nearest neighbors, Support Vector Machine, decision tree, and logistic regression are the five classification methods that were applied and tested. Each one has its own features for classifying the data set. The Random Forest classifier was found to be the best classifier for building our model, outperforming other classifiers in accuracy, precision, recall, and f-score.

5 Conclusion

Due to the spread of the COVID-19 worldwide, and with the invasion of Technology-enhanced environments, there became an increase in the amount of data in the education sector, notably the data from online eTextbooks., which can be utilized to predict the performance of the student through teaching a particular course. Our study focused on the data structure and algorithms (CS2) course which was taught using an eTextbook at a large public research institution.

The purpose of our study is to build an early predictive model for students' performance. Students who are at risk of failure may be identified early using this prediction model, and they can be guided in the right direction for better results in the future. this predictive model may be helpful in reducing the drop-out ratio and will help improve students' performance. it may be helpful in looking for students who demand particular attention to minimize the failure rate and take required measures to improve their performance.

We addressed two problems: the prediction of good/bad performance and the prediction of the final exam grade. Both of them aim at the early prediction of Students who are at risk of failure.

The students' performance was predicted using classification, and the students' final exam grades were predicted using regression techniques. We evaluated

multiple Linear Regression and Random Forest Regression in regression analysis. We evaluated the Random Forest classifier, Logistic regression, Support Vector Machine, K Nearest Neighbors, and Decision Tree for classification analysis. To find the best features for the regression models, we applied a correlation coefficient feature selection approach. during classification, We discovered that the student's performance is influenced by all features, hence all features were chosen as parameters in the classification models.

All of These different approaches were compared based on their accuracy and error statistics. Based on experiments results, we found that the algorithm with the best result overall in classification was Random Forest Classifier with an accuracy equal to 91.7% while in regression it was Random Forest Regression with R^2 equal to 0.977.

6 Future Work

In the future, Experiments can be broadened to incorporate additional distinguishing features in order to acquire more accurate data that can be utilized to improve student learning outcomes. Data mining experiments can be undertaken to gain a broader perspective and produce more valuable results. More datasets will be collected, and various data mining techniques such as clustering and association will be used to compare and analyze them.

Conflict of Interest

All authors declare that there is no conflict regarding the publication of this paper.

References

[1] Kavipriya, P. "A review on predicting students' academic performance earlier, using data mining techniques." ,*International Journal of Advanced Research in Computer Science and Software Engineering* ,**6.12** (2016): 101-105.

[2] https://opendsa-server.cs.vt.edu /ODSA/Books/CS2/html/index.html ,Date of last visit Tue, May 2022.

[3] SHAFFER, Clifford A., et al. Opendsa: beginning a community active-ebook project , *P*roceedings of the 11th Koli Calling International Conference on computing education research, (2011) 112-117.

[4] https://www.irishtimes.com/news/education/more-than-6-000-students-drop-out-of-college-in-first-year-1.3062362, Date of last visit Tue, May 2022.

[5] Kinnunen, P., & Malmi, LẆhy students drop out CS1 course?, *In Proceedings of the second international workshop on Computing education research*, (2006) 97-108.

[6] https://www.creatrixcampus.com/blog/7-reasons-why-students-drop-out , Date of last visit: Date of last visit Tue, April 2022.

[7] Brusilovsky, Peter, et al. "What should be visualized? Faculty perception of priority topics for program visualization." ,*A*CM SIGCSE Bulletin , **38.2** (2006): 44-48.

[8] Strecht, Pedro, et al. "A Comparative Study of Classification and Regression Algorithms for Modelling Students' Academic Performance." , *I*nternational Educational Data Mining Society ,(2015).

[9] Pandey, Umesh Kumar, and Saurabh Pal. "Data Mining: A prediction of performer or underperformer using classification.", arXiv preprint arXiv:1104.4163 ,(2011).

[10] Al-Radaideh, Qasem A., Emad M. Al-Shawakfa, and Mustafa I. Al-Najjar. "Mining student data using decision trees." ,International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan (2006).

[11] Ben-Zadok, Galit, et al. "Examining online learning processes based on log files analysis: A case study.", 5th International Conference on Multimedia and ICT in Education (m-ICTE'09), 2009.

[12] Cortez, Paulo, and Alice Maria Gonçalves Silva. "Using data mining to predict secondary school student performance." ,(2008).

[13] Archana, S. & Elangovan, K. Survey of classification techniques in data mining, International Journal of Computer Science and Mobile Applications, **2.2** (2014) 65-71.

[14] Kalpana, JK Jothi, and K. Venkatalakshmi. "Intellectual performance analysis of students' by using data mining techniques.", International Journal of Innovative Research in Science, Engineering and Technology ,(2014) 1-8.

[15] Kaur, Parneet, Manpreet Singh, and Gurpreet Singh Josan. "Classification and prediction-based data mining algorithms to predict slow learners in the education sector." ,Procedia Computer Science ,**57** (2015) 500-508.

[16] Hu, Ya-Han, Chia-Lun Lo, and Sheng-Pao Shih. "Developing early warning systems to predict students' online learning performance." ,Computers in Human Behavior ,**36** (2014): 469-478.

[17] Kabakchieva, Dorina. "Student performance prediction by using data mining classification algorithms." ,International journal of computer science and management research, **1.4** (2012) 686-690.

[18] Baradwaj, Brijesh Kumar, and Saurabh Pal. "Mining educational data to analyze students' performance.", arXiv preprint arXiv: 1201.3417 ,(2012).

[19] Fouh, Eric, et al. "Exploring students learning behavior with an interactive etextbook in computer science courses." ,Computers in Human Behavior, ,**41** (2014) 478-485.

[20] KARAVIRTA, Ville; SHAFFER, Clifford A. JSAV: the JavaScript algorithm visualization library., In: Proceedings of the 18th ACM conference on Innovation and technology in computer science education. ,(2013) 159-164.

[21] Malmi, Lauri, et al. "Visual algorithm simulation exercise system with automatic assessment: TRAKLA2.", Informatics in education, **3.2** (2004) 267-288.

[22] http://github.com/Khan/khan-exercises, Date of last visit Tue, April 2022.

[23] Han, Jiawei, Jian Pei, and Micheline Kamber. Data mining: concepts and techniques. Elsevier, (2011).

[24] Bhardwaj, Brijesh Kumar, and Saurabh Pal. "Data Mining: A prediction for performance improvement using classification." ,arXiv preprint arXiv:1201.3418, (2012).

[25] Romero, Cristobal, José Raúl Romero, and Sebastián Ventura. ”A survey on pre-processing educational data.”, Educational data mining. Springer, Cham, ,(2014) 29-64.
[26] Baker, Ryan Shaun, and Paul Salvador Inventado. ”Educational data mining and learning analytics.” ,Learning analytics. Springer, New York, NY, (2014) 61-75.
[27] Karegowda, Asha Gowda, A. S. Manjunath, and M. A. Jayaram. ”Comparative study of attribute selection using gain ratio and correlation based feature selection.”, International Journal of Information Technology and Knowledge Management ,**2.2**,(2010)271-277.
[28] TSYMBAL, Alexey; PUURONEN, Seppo. Bagging and boosting with dynamic integration of classifiers. , European Conference on Principles of Data Mining and Knowledge Discovery. Springer, Berlin, Heidelberg, (2000) 116-125.
[29] Gislason, Pall Oskar, Jon Atli Benediktsson, and Johannes R. Sveinsson. ”Random forests for land cover classification.”, Pattern recognition letters, **27.4** (2006) 294-300.
[30] GULL, Hina, et al. Improving learning experience of students by early prediction of student performance using machine learning, IEEE International Conference for Innovation in Technology (INOCON), (2020) 1-4.
[31] Chawla, Nitesh V., et al. ”SMOTE: synthetic minority over-sampling technique.”, Journal of artificial intelligence research ,**16** (2002) 321-357.
[32] Karthikeyan, K., and P. Kavipriya. ”On Improving student performance prediction in education systems using enhanced data mining techniques.”, International Journal of Advanced Research in Computer Science and Software Engineering , **7.5** (2017).
[33] Agrawal, Surbhi, Santosh K. Vishwakarma, and Akhilesh K. Sharma. ”Using data mining classifier for predicting student’s performance in UG level.”, International Journal of Computer Applications, **172.8** (2017) 39-44.
[34] Al-Radaideh, Qasem A., Emad M. Al-Shawakfa, and Mustafa I. Al-Najjar. ”Mining student data using decision trees.” ,International Arab Conference on Information Technology (ACIT’2006), Yarmouk University, Jordan (2006).
[35] Powers, David MW. ”Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.” ,arXiv preprint arXiv:2010.16061 ,(2020).
[36] CHEN, Tsong Yueh; KUO, Fei-Ching; MERKEL, Robert. On the statistical properties of the f-measure, Fourth International Conference onQuality Software, 2004. QSIC 2004. Proceedings. IEEE, (2004) 146-153.