Early Forecasting of At-Risk Students of Failing or Dropping Out of a Bachelor's Course Given Their Academic History - The Case Study of Numerical Methods

Isaac Caicedo-Castro[123], Mario Macea-Anaya[234], Samir Castaño-Rivera[13]
[1]*Socrates Research Team*
[2]*Research Team: Development, Education, and Healthcare*
[3]*Faculty of Engineering*
[4]*CINTIA, Center of INnovation in Technology of Information to support the Academia*
University of Córdoba
Carrera 6 No. 76-103, 230002, Montería, Colombia
emails: {isacaic, mariomacea, sacastano}@correo.unicordoba.edu.co

***Abstract*—In this work, we ponder the following research question: Is it possible to predict if a given student might either fail or drop out of an undergraduate course taking into account its performance in prerequisite courses? Therefore, we study the case of forecasting the risk faced by students of failing or dropping out of the course of numerical methods in an engineering bachelor's program. To this end, the prediction is based on the student's academic history, which consists of the grades the student has obtained in previous prerequisite courses, whose concepts and skills are required to succeed in the studied case of the numerical methods course. Additionally, the admission test results are also used for forecasting purposes. Moreover, we adopt machine learning, where supervised methods for classification are fitted using the academic history of students enrolled in the Engineering bachelor's program with a major in Systems Engineering at the University of Córdoba in Colombia. We collected the academic history of 56 anonymized students and carried out 10-fold cross-validation. The results of this study reveal that a support vector machines method predicts if a given student is at risk of failing or withdrawing from the numerical methods course with mean values for accuracy, precision, recall, and harmonic mean (F_1) of 76.67%, 71.67%, 51.67%, and 57.67%, respectively. This method outperforms the others studied in this work.**

Keywords*—Machine learning; educational data mining; classification algorithm; dropout and failure forecasting; student long-term retention.

I. INTRODUCTION

This work is part of a broader project called Course Prophet, whose goal is to design and implement an intelligence system that predicts if a student is at risk of failing or dropping a bachelor's course, that belongs to the scientific computing area in engineering, such as, e.g., numerical methods, linear programming, and so forth. Therefore, in this work, we have studied the case of forecasting if a given student might fail or withdraw from the numerical methods course in the context of a bachelor's program with a major in systems engineering at the University of Córdoba in Colombia.

The problem coped in this work shall be defined in Section I-A, whilst we shall discuss the motivation to solve it in Section I-B. In Section I-C, we shall present the key assumptions taken into account in this study, and its scope. Furthermore, we shall outline the contributions and organization of the remainder of this paper in Section I-D.

A. Problem statement

In a bachelor's degree, courses are organized and grouped into each semester to train students. The foundation of each subject is typically covered in the first semesters, with more advanced topics introduced in the later semesters. This gradual progression enables students to acquire skills and knowledge progressively, starting with the basic concepts and building towards more elaborated theories. However, students may still find the coursework challenging at times, and instructors need to ensure that they do not become overwhelmed, to ensure a positive learning experience. So, some courses are prerequisites of more advanced ones, for instance, differential calculus is required to understand linear and non-linear programming. Therefore, it is expected that the student's performance in a given course is influenced by their performance in prerequisite courses. For example, a student who struggled to pass differential calculus might have poor performance in differential equations and numerical methods.

Considering the relationship between courses, we ponder the following research question: Can an artificial intelligence system learn the regular patterns in a student's academic history to predict whether the student is at risk of failing or withdrawing from a course, based on their academic performance in the prerequisite courses?

To answer this question, we studied the case of the numerical methods course, which builds on concepts taught in prerequisite courses like calculus, physics, and computer programming. This case study focused on the context of the engineering students at the University of Córdoba in Colombia, a public university.

To determine a student's risk of failing or withdrawing from the numerical methods course, we analysed their performance in these prerequisite courses. In addition, we also considered their performance in the Saber 11 test, which is the standardized test used for bachelor program admission in Colombia. In the United States, a similar test called the Scholastic

Copyright (c) IARIA, 2023. ISBN: 978-1-68558-049-0

Assessment Test (SAT) is used for the same purpose. The University of Córdoba admits or rejects candidates based on their Saber 11 test scores.

The performance in prerequisite courses is measured through the student's grades, whilst the admission test scores achieved by the student, measure their proficiency in every subject evaluated in the Saber 11 test. Therefore the student's grades and Saber 11 scores are the independent variables or the student's features, whereas the failure or dropout risk is the target variable (a.k.a., dependent variable), hence, the problem is finding the functional dependency between independent variables and the target variable. In the particular context of this study, the artificial intelligence system must infer the regular patterns between the risk of failing or withdrawing from the numerical methods course and the grades and Saber 11 scores achieved by the students in the past.

B. Motivation

Bachelor students at Colombian universities are graded in the range from 0 up to 5. To maintain their student status, bachelor students at Colombian universities must achieve a minimum global average grade. At the University of Córdoba, students are required to maintain a global average grade of at least 3.3, as specified in Article 16 of the university's student code [1]. Article 28-th of the university's code states that if a student's global average grade is between 3 and 3.3, they must increase their grade to at least 3.3 in the next semester or risk being dropped out. If a student's grade falls below 3, they are automatically withdrawn from the university (cf., article 16-th in the student's code [1]). The student who fails courses might lose their student status according to university rules. This problem is commonly referred to as *student dropout*.

On the other hand, those students who dropout courses might take longer to fulfil the requirements to receive their bachelor's degree. This problem is known as *long-term retention*. Both issues might cause students psychological issues, frustration, and financial loss.

Identifying students at risk in advance, allows professors and lecturers to run plans of action and strategies to handle previously mentioned issues. Moreover, precautions may be taken to prevent those students from failing or withdrawing from their courses. Some strategies include psychological support for students, or professors might suggest books, papers, or websites, amongst other educative resources, which students at risk might consult to review the material required to succeed in the course.

Thus, eventually, students' dropout and long-term retention rates might decrease, considering that both problems are a serious concern in the higher education systems and for policy-making stakeholders at universities [2].

C. Key Assumptions and Limitations

In this study we have taken into account the following assumptions:

- In the context of this study, we assumed that to succeed in Numerical Methods course, the prerequisites are Linear Algebra, Calculus I, II, III, Physics I, II, III, Introduction to Computer Programming, Computer Programming I, II, and III. The subjects included in the Numerical Methods course are as follows:
    (i) Approximations and computer arithmetic: the concepts to understands these subjects are taught in Introduction to Computer Programming.
    (ii) Non-linear equations: students must have a working knowledge of integral calculus (taught in Calculus II), be able to program computers using iterative and selective control structures (skills taught in both Introduction to Computer Programming and Computer Programming I), and understand Taylor series, which is the foundation of the secant method, a numerical method used to solve non-linear equations.
    (iii) Systems of linear equations: the student must be familiar with matrix and vector operations taught in linear algebra in order to understand numerical methods such as, e.g., Gauss-Seidel or Jacobi. Besides, programming such methods are subjects dealt in computer programming II course.
    (iv) Interpolation: the student must know the topics taught in calculus II to understand the background of the Taylor polynomial interpolation, and the subjects taught in courses such as linear algebra, computer programming I, and II to implement the other numerical methods for interpolation.
    (v) Numerical integration: in this subject, algorithms are used for computing integrals which cannot be solved through analytic methods, hence, the student must know what integration is (taught in calculus II course), and how to calculate some integrals to understand this subject.
    (vi) Ordinary differential equations: In this subject, the student must know concepts from all prior mathematics courses. It would be appropriate if the student would have attended a differential equation course, however, in the context of this study, this course is simultaneously scheduled with numerical methods, so students attend both in the same semester.
    (vii) Numerical optimization: this subject is an introduction for more advanced courses such as, e.g., statistics, linear and non-linear programming, stochastic methods courses, and machine learning. To understand this subject, the student must have mastered topics taught in courses, such as computer programming II and III, linear algebra, basic calculus, and vector calculus (which is taught in calculus III course).
- We assumed that a given students is at risk as long as they might either fail or dropout the numerical methods course.
- We assumed the admission test called Saber 11 might

be useful to forecast the failure or dropout risk. As a consequence, the score in each evaluated area is an input variable for the prediction.

- We assumed that the student's grades in prerequisite courses and the score in the admission test are sufficient input variables for forecasting the failure or dropout risk.

The scope of the research is limited as follows:

- We did not aim at designing an artificial intelligence system that predicts the dropout rate nor the failure rate of a given course.
- We did not consider additional input variables for the prediction, such as, e.g., gender, ethnicity, or economic variables, because the students who took the survey are alike regarding these features. Thus, these features do not help to differentiate students contributing little information to the forecasting process. For instance, Figure 1 shows that 83.95% of the students in the sample are male. Figure 2 indicates that over 90% of the students do not consider themselves part of an ethnic group. Additionally, Figure 3 reveals that more than 80% of the students belong to the first economical stratum. This aligns with the information presented in Figure 4, where over 90% of the students' family incomes are lower than two monthly minimum wages.

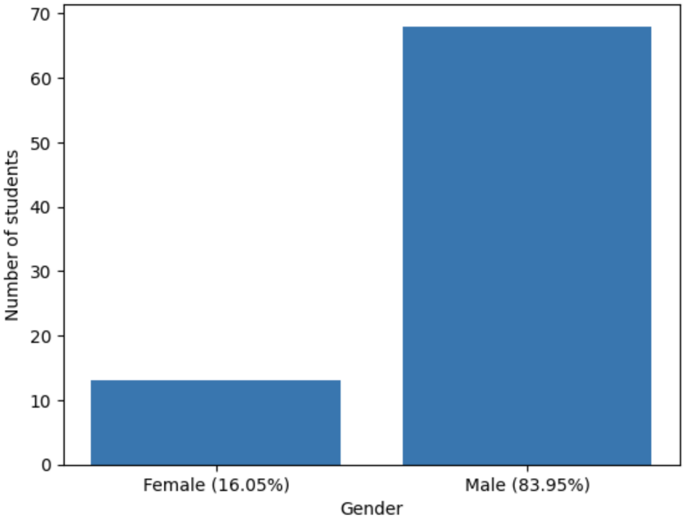

Figure 1. Sample distribution according to the students' gender.

D. Contributions and Paper Outline

The contributions of this work are as follows:

(i) A dataset with 56 records, each one with 38 attributes corresponding to the independent variables, and another attribute, which is the target variable. These students have attended courses from the fifth semester up to the ninth semester during the second half of 2022.

(ii) The prototype of an intelligence system that learns regular patterns from the students' academic history to predict if a student might fail or dropout the numerical methods course.

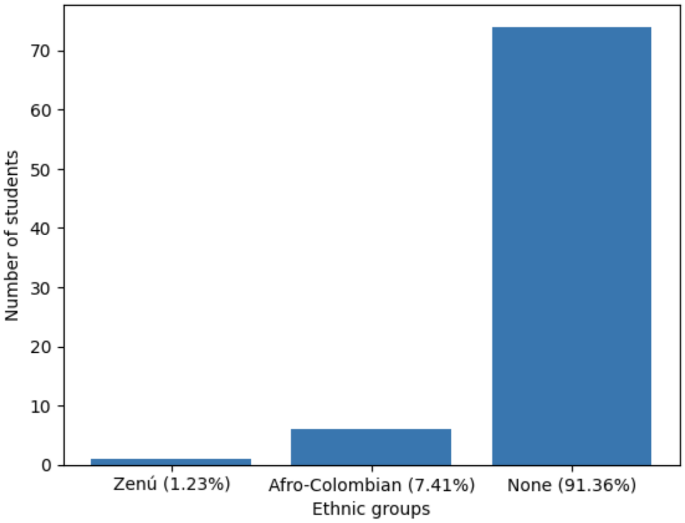

Figure 2. Sample distribution according to the students' ethnic group.

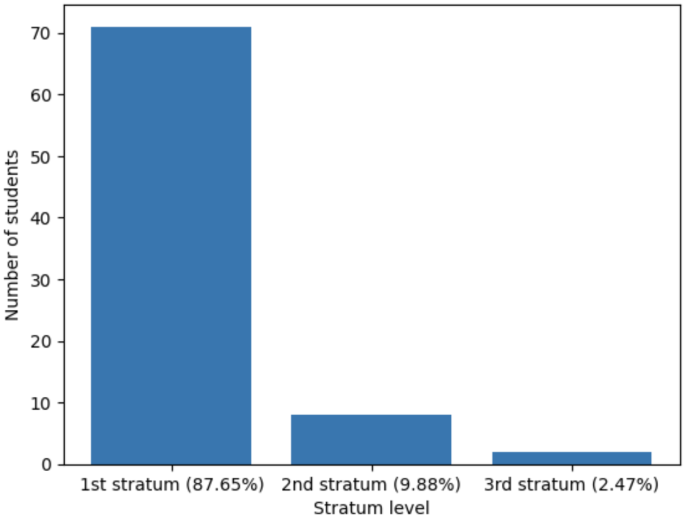

Figure 3. Sample distribution according to the students' economic stratification.

(iii) An empirical study that reveals the support vector machine outperforms decision trees, Gaussian processes, artificial neural networks, amongst other machine learning methods. During the evaluation, the support vector machines achieved the mean values for accuracy, precision, recall, and harmonic mean (F_1) of 76.67%, 71.67%, 51.67%, and 57.67%, respectively.

The rest of this paper is outlined as follows: in Section II, we shall discuss the prior research on the problem addressed in this work, whilst we present the methods adopted in this study in Section III. In Section IV, we shall delve into de details of experimental setting, present and analyse the results. Finally, we shall draw the conclusions of this study and describe directions for further work in Section V.

ISBN: 978-1-68558-049-0

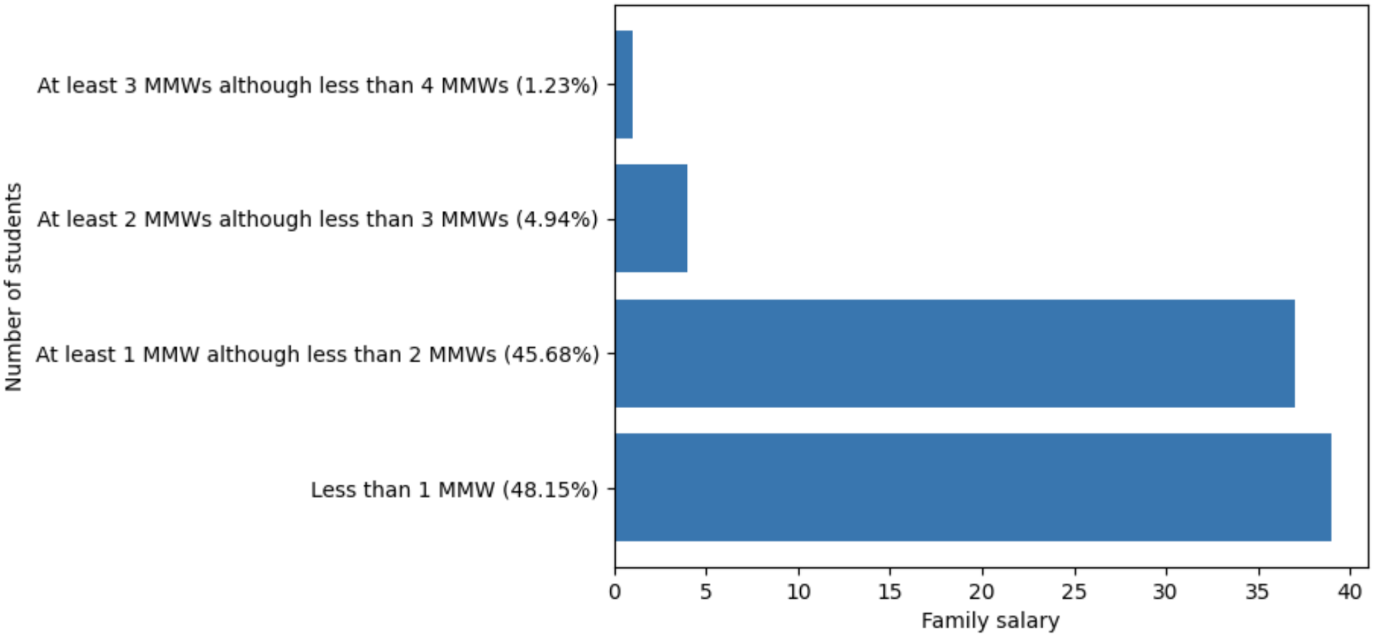

Figure 4. Sample distribution according to the students' family incomes in terms of Monthly Minimum Wages (MMW).

II. Related Work

This work falls within the domain of educational data mining, which aims to apply machine learning methods to large educational datasets to gain insights into students' learning behaviour. This includes analysing educational data, studying pedagogical theories using data mining, understanding students' domain knowledge, and evaluating their engagement in learning tasks.

Although course failure and dropout are general education problems, they have been mainly studied in the context of online education where predicting student dropout is a concerning issue. Previous research has used machine learning methods to forecast whether a given student will drop out of specific online courses, such as Computer Networks and Communications, and Web Design [3]. In contrast, our study aims not only to predict dropout but also to forecast the risk of failing a course. Additionally, our system aims to make predictions before the student begins the course, whilst the previously mentioned study has focused on predicting dropout during the course development.

Other studies have also focused on predicting failure and dropout risk, but they have used different independent variables compared to those in our study. For example, some studies have used the number of course views and scores achieved in assignments, tests, and projects as independent variables [4]. Other studies have used variables such as academic year, in addition to the aforementioned variables and others [5].

Only academic data, such as, e.g., students' age and grades have been used for predicting students dropout as well. However, grades have been considered ordinal data in lieu of quantitative information in some studies [6].

More recent studies conducted in the context of online courses have focused on the prediction of course dropout risk in STEM (Science, Technology, Engineering, and Mathematics) oriented courses [7] and mathematics course [8]. These studies use various independent variables, including the number of content downloads, scores obtained on weekly quizzes, video lesson views, and overall student activity in the course. The target variable is the course dropout risk.

It is important to note that in both of the latter related works, the prediction of the risk of not completing the course is primarily based on the student's behaviour during the learning process. Nevertheless, our study takes a different approach by focusing on the performance of the student in previous courses.

Additionally, other study is focused on predicting dropout from bachelor's degree instead of a specific course [9].

Furthermore, in the above-mentioned related works, the following machine learning classifiers have been adopted: artificial neural networks or multilayer perceptrons [3]–[6], [8], support vector machines [3], [4], [9], logistic regression [4], [7], [9], decision trees [4], [7], [9], ensemble methods with different kind of classifiers [3], [5], random forest [4]–[6], gradient boosting [6], XGBoost [5], [6], and variants of gradient boosting [6], [9], namely CatBoost [10] and LightGBM [11].

Finally, as far as we know, no related work has focused on the independent variables that we consider in this work, which are based on prerequisite student performance. Our aim is to predict the risk of failure or dropout for a specific course.

III. Methods

In this work, we adopted a quantitative approach, using students' grades that measure the performance during their academic history, including their scores achieved in the admission test known as Saber 11. Moreover, due to we used machine learning methods for forecasting, this work is also experimental regarding the nature of this approach, i.e., machine learning is an empirical discipline. With the experimental work, our goal is measuring the quality of the forecasting that depends on the capability of the machine learning methods to generalize properly with new input data.

In order to fit the machine learning methods, it is necessary to collect a dataset that include the history of students who have failed, dropped out and succeeded the numerical methods course, including their performance in the prerequisite courses and admission test. The machine learning methods capture the regular patterns that let the intelligence system predict the target variable given new input variables corresponding to future students.

The remainder of this section is organized as follows: we shall explain the procedure carried out to collect the dataset in Section III-A. In Section III-B, we shall discuss about the machine learning methods adopted in this study. Finally, in Section III-C, we shall describe the evaluation approach conducted in this work.

A. The Dataset Collection

In 2022, we conducted a survey on 81 students pursuing the bachelor's degree of engineering with major in systems engineering at the University of Córdoba in Colombia. These students have attended courses from the fifth semester up to the ninth semester.

In 2018, the curriculum structure of the above-mentioned bachelor's changed, thereby, we dropped 25 records corresponding to those students who started to pursue the bachelor's degree with the previous curriculum structure. Therefore the resulting dataset contains 56 out of 81 original records.

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^D \wedge y_i \in \{0, 1\} \forall i = 1, 2, \ldots n\}$ be the complete dataset, where D and n are the number of independent variables and records, respectively. If the ith student either failed or dropped out the numerical method course, the target variable is equal to one, i.e., $y_i = 1$, otherwise it is equal to zero, i.e., $y_i = 0$. The real-valued $\mathbf{x}_i$ represents the ith student's record, and its components represent the independent variables. The first five components are scores achieved by a given student in each subject evaluated through the admission test called Saber 11, and each score is in the range of 0 up to 100, i.e., $x_{ij} \in \mathbb{Z}$, where $0 \leq x_{ij} \leq 100$ for $j = 1, 2, \ldots, 5$. On the other hand, for each prerequisite course, there are components whose values are the student's highest and lowest grade, including the number of semesters the student has attended the course. Each grade is a real-valued number between 0 and 5. There are eleven prerequisite courses, hence, there are thirty three components, besides the previous five ones, thereby, there is a total of 38 components in every vector, i.e., $D = 38$. The meaning of each component is explained as follows:

- x_{i1} is the score achieved by the ith student in the mathematics subject of the admission test.
- x_{i2} is the score achieved by the ith student in the natural science subject of the admission test.
- x_{i3} is the score achieved by the ith student in the social science subject of the admission test.
- x_{i4} is the score achieved by the ith student in the critical reading subject of the admission test.
- x_{i5} is the score achieved by the ith student in the social English proficiency evaluation of the admission test.
- x_{i6} is the best grade that a given student achieved in Calculus I course.
- x_{i7} is the number of semester a given student has attended the Calculus I course.
- x_{i8} is the worst that a given student achieved in Calculus I course.
- x_{i9} is the best grade that a given student achieved in Calculus II course.
- $x_{i,10}$ is the number of semester a given student has attended the Calculus II course.
- $x_{i,11}$ is the worst that a given student achieved in Calculus II course.
- $x_{i,12}$ is the best grade that a given student achieved in Calculus III course.
- $x_{i,13}$ is the number of semester a given student has attended the calculus III course.
- $x_{i,14}$ is the worst that a given student achieved in Calculus III course.
- $x_{i,15}$ is the best grade that a given student achieved in Linear Algebra course.
- $x_{i,16}$ is the number of semester a given student has attended the Linear Algebra course.
- $x_{i,17}$ is the worst that a given student achieved in Linear Algebra course.
- $x_{i,18}$ is the best grade that a given student achieved in Physics I course.
- $x_{i,19}$ is the number of semester a given student has attended the Physics I course.
- $x_{i,20}$ is the worst that a given student achieved in Physics I course.
- $x_{i,21}$ is the best grade that a given student achieved in Physics II course.
- $x_{i,22}$ is the number of semester a given student has attended the Physics II course.
- $x_{i,23}$ is the worst that a given student achieved in Physics II course.
- $x_{i,24}$ is the best grade that a given student achieved in Physics III course.
- $x_{i,25}$ is the number of semester a given student has attended the Physics III course.
- $x_{i,26}$ is the worst that a given student achieved in Physics III course.
- $x_{i,27}$ is the best grade that a given student achieved in Introduction to Computer Programming course.
- $x_{i,28}$ is the number of semester a given student has attended the Introduction to Computer Programming course.
- $x_{i,29}$ is the worst that a given student achieved in Introduction to Computer Programming course.
- $x_{i,30}$ is the best grade that a given student achieved in

Computer Programming I course.

- $x_{i,31}$ is the number of semester a given student has attended the Computer Programming I course.
- $x_{i,32}$ is the worst that a given student achieved in Computer Programming I course.
- $x_{i,33}$ is the best grade that a given student achieved in Computer Programming II course.
- $x_{i,34}$ is the number of semester a given student has attended the Computer Programming II course.
- $x_{i,35}$ is the worst that a given student achieved in Computer Programming II course.
- $x_{i,36}$ is the best grade that a given student achieved in Computer Programming III course.
- $x_{i,37}$ is the number of semester a given student has attended the Computer Programming III course.
- $x_{i,38}$ is the worst that a given student achieved in Computer Programming III course.

The dataset is not utterly balanced, nevertheless, Figure 5 shows it contains enough positive examples, namely those where the students have either failed or dropped out courses.

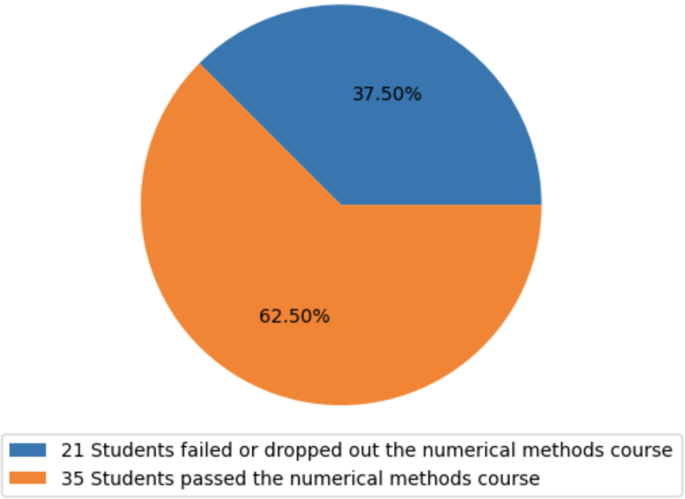

Figure 5. Distribution of students who have failed or dropped out the numerical course in compared to those ones who passed it

B. Classification Methods

The problem addressed in this work is finding the functional dependency between the independent variables, $\mathbf{x} \in \mathbb{R}^D$, and the target variable, $y \in \{0, 1\}$, in other words, fitting the function $f : \mathbb{R}^D \to \{0, 1\}$ given the training dataset (which is a portion of the whole dataset). To cope this problem we adopted supervised learning approach, specifically, classification methods.

We used logistic regression as our first classification method to predict the probability between two possible outcomes. Logistic regression utilizes the sigmoid function of the linear combination between input variables and weights, which are fitted by maximizing the objective function based on the log-likelihood of the training data given the binary outcome [12]. We fitted the logistic regression classifier through Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm [13], [14].

With the logistic regression method, it is assumed there exists a discriminant hyperplane to separate the examples in two classes, which might be a reasonable assumption regarding the high dimensionality of the dataset used in this study. Nevertheless, we also adopted other classifiers that are far better suited for non-linear classification problems, such as the Gaussian process classifier. This is a probabilistic method based on Bayesian inference. In the Gaussian process, the probability distribution of the target variable is Gaussian or normal, this explains the name of the method [15], [16]. The main advantage of the Gaussian process classifier is the possibility of incorporating prior knowledge about the problem, improving its forecasting even when the training dataset is small. However, the computational cost of fitting and making predictions with this method can become an issue in domains with large-scaled datasets. In the context of this study, the dataset is rather small, which makes the Gaussian process classifier a suitable choice.

So far, the support vector machines (SVM) method is the best theoretical motivated and one of the most successful methods in the practice of modern machine learning [17, pg. 79]. It is based on convex optimization, allowing for a global maximum solution to be found, which is its main advantage. However, SVM method is not well-suited for interpretation in data mining and is better suited for training accurate intelligent systems. A broader description of this algorithm can be found in the work by Cortes and Vapnik [18].

Both SVM and logistic regression are linear classification methods that assume the input vector space can be separated by a linear decision boundary (or a hyperplane in the case of a multidimensional space). However, when this assumption is not satisfied, SVM can be used with kernel methods to handle non-linear decision boundaries (see Cortes and Vapnik [18] for further details).

Although SVM method is considered one of the most successful methods in the practice of modern machine learning, multilayer perceptrons and their variants, which are artificial neural networks, are the most successful methods in the practice of deep learning and big data, particularly in tasks such as speech recognition, computer vision, natural language processing, and so forth. [19, pg. 3]. In this research, we have adopted the multilayer perceptrons fitted through back-propagated cross-entropy error [20], and the optimization algorithm known as Adam [21]. We used multilayer perceptrons with one and five hidden layers.

The multilayer perceptron method is a universal approximator (i.e., it is able to approximate any function for either classification or regression), which is its main advantage. However, its main disadvantage is that the objective function (a.k.a., loss function) based on the cross-entropy error is not convex. Therefore, the synaptic weights obtained through the fitting process might not converge to the most optimum solution because there are several local minima in the objective function. Thus, finding a solution depends on the random initialization of the synaptic weights. Furthermore, multilayer perceptrons have more hyperparameters to be tuned than other learning algorithms (e.g., support vector machines or naive Bayes), which is an additional shortcoming.

Except for the logistic regression method, all the above-mentioned methods are not easily interpretable. Therefore, we adopted decision trees, which are classification algorithms commonly used in data mining and knowledge discovery. In decision tree training, a tree is created using the dataset as input, where each internal node represents a test on an independent variable, each branch represents the result of the test, and leaves represent forecasted classes. The construction of the tree is carried out in a recursive way, beginning with the whole dataset as the root node, and at each iteration, the fitting algorithm selects the next attribute that best separates the data into different classes. The fitting algorithm can be stopped based on several criteria, such as when all the training data is classified or when the accuracy or performance of the classifier cannot be further improved.

Decision trees are fitted through heuristic algorithms, such as greedy algorithms, which may lead to several local optimal solutions at each node. This is one of the reasons why there is no guarantee that the learning algorithm will converge to the most optimal solution, as is also the case with the multilayer perceptrons algorithm. Therefore, this is the main drawback of decision trees, and it can cause completely different tree shapes due to small variations in the training dataset. The decision trees were proposed in 1984, Breiman *et al.* delve into their details (cf., [22]). We also adopted ensemble methods based on multiple decision trees such as, e.g., Adaboost (stands for adaptive boosting) [23], Random forest [24], and XGBoost [25].

C. Evaluation Approach

To evaluate the machine learning methods, we need several pairs of training and test datasets. To this end, we carried out experiments based on K-Fold Cross-Validation (KFCV), hence, from the original dataset, we get K pairs of training and test datasets. We chose $K = 10$, where it is usually 10 or 30. We did not choose $K = 30$ because the dataset is small. Thus, we test each method K times through KFCV. With the test outcomes, we calculate the mean accuracy, mean precision, and mean recall to compare the learning methods, and choosing the best hyper-parameters for each method (e.g., the regularization parameter in the multilayer perceptrons and logistic regression).

IV. Evaluation

Given the no free lunch theorem, there is no universal best machine learning method for the problem at hand. To identify the most effective method, we conducted an experiment using the models described in Section III-B. Details of the experimental setup can be found in Section IV-A, with results and their discussion presented in Sections IV-B and IV-C, respectively.

A. Experimental Setting

The evaluation is conducted through K-fold cross-validation, where K = 10, as it was mentioned in Section III-C. This procedure is preformed in a dataset that contains 56 records or examples, with 38 independent variables for each example (see Section III-A). The dataset is available online to allow the reproduction of our study, and for further research [26].

Finally, we programmed all the experiments with Python, using the Scikit-Learn library [27], in Google Colaboratory [28].

B. Results

The results of the 10-fold cross-validation evaluation are summarized in table I. Support vector machines (SVM) with radial basis function kernel outperforms the other classification methods in terms of accuracy and harmonic mean (F_1). Nonetheless, Random forest classifier achieved the highest precision, whilst the decision tree classification method reached the best recall.

The mean recall value of the SVM with radial basis is in line with the confusion matrix presented in Table II. During all iterations of the 10-fold cross-validation evaluation, the classifier correctly predicted only 11 out of 21 students at risk, resulting in almost half of the actual positive examples being falsely classified as negative (i.e., false negative instances). Conversely, the SVM classifier misclassified only three examples as positive, which aligns with the mean precision achieved during the evaluation. These results indicate that the classifier has a low probability of misclassifying students not at risk as being at risk, which is beneficial in avoiding the wastage of resources for students who do not need them. However, this classifier might miss identifying some students who are actually at risk. The decision tree with entropy index outperformed the recall of others classifiers, nevertheless, the mean recall difference between the decision tree and SVM is not statistically significant (i.e., p-value $>$ 0.05), as it is shown in Table I.

Regarding the size of the dataset, Figure 6 shows the receiver operating characteristics (ROC) curve for the SVM with radial basis function kernel. The area under the curve (AUC) of 0.68 indicates that the classifier performs better than random guessing, or that it provides some level of discrimination between positive and negative examples. However, a larger dataset might improve this performance. It is worth noting that the AUC of this classifier is lower than the AUC of the classifiers shown in Figure 7. In particular, random forest outperforms the other classification methods in distinguishing between positive and negative examples, with an AUC of 0.77.

In the domain of this study, where we are interesting in predicting students at risk, it might be more important a classification method that forecasts accurately to either avoid spend resources in students who do not require them, or to not help students who actually might fail or dropout the numerical methods course. In other domains, such as, e.g.,

TABLE I
TEN-FOLD CROSS-VALIDATION RESULTS

Machine learning method	*Mean Accuracy (%)*	*p-value*	*Mean Precision (%)*	*p-value*	*Mean Recall (%)*	*p-value*	*Mean* $\mathbf{F_1}$ *(%)*	*p-value*
Support vector machines with the radial basis function kernel	**76.67**		71.67		51.67		**57.67**	
Support vector machines with the sigmoid kernel	71.67	0.45	40	0.15	23.33	0.08	28.33	0.08
Support vector machines with the polynomial kernel (degree = 3)	66.67	2.26	48.33	0.22	48.33	0.84	46.67	0.51
Decision tree with entropy index	68.33	0.34	48.33	0.18	**70**	0.3	56.57	0.94
Decision tree with gini index	62.33	0.11	47.5	0.18	56.67	0.76	50	0.61
Logistic regression	64	0.1	47.5	0.24	28.33	0.09	33.33	0.1
Multilayer perceptron with a single hidden layer	57	**0.005**†	18.33	**0.006**†	25	0.1	20	**0.01**†
Multilayer perceptron with five hidden layers	64.33	**0.04**†	6.67	**0.0003**†	10	**0.01**†	8	**0.001**†
Gaussian process with the rational quadratic kernel	68.67	0.31	50	0.32	28.33	0.14	35	0.17
Gaussian process with the periodic kernel	62.33	**0.01**†	0	$\mathbf{3.57 \times 10^{-5}}$†	0	$\mathbf{1.3 \times 10^{-4}}$†	0	$\mathbf{3.05 \times 10^{-5}}$†
Gaussian process with the dot product kernel	62.33	**0.04**†	43.33	0.12	43.33	0.61	39.67	0.21
Gaussian process with the Matern kernel	69.67	0.32	65	0.72	38.33	0.3	38.33	0.45
Gaussian process with the radial basis function kernel	73.33	0.66	70	0.93	48.33	0.82	53.67	0.78
Gaussian process with a sum of radial basis function and Matern kernel	70	0.4	63.33	0.68	38.33	0.37	45.67	0.44
Random forest with the entropy index	72	0.56	**75.83**	0.82	48.33	0.8	55.67	0.88
Adaboost with the entropy index	66.33	0.23	46.67	0.15	65	0.43	53.57	0.79
XGBoost	61	0.08	45.83	0.18	38.33	0.37	39.67	0.24

†Student's paired t-test reveals the difference between means is statistically significant

TABLE II
CONFUSION MATRIX FOR SUPPORT VECTOR MACHINES WITH RADIAL BASIS FUNCTION

	Forecasted class		
True class	*Student without risk*	*Student at risk*	*Total*
Student without risk	32	3	35
Student at risk	10	11	21
Total	42	14	56

fraud detection, it might be far more useful a classifier with high AUC to minimize false positives.

Furthermore, the best hyper-parameter setting for each approach, corresponding with the results shown in table I, is presented as follows:

- Gaussian process classifier with radial basis function kernel, the best values for σ and γ are 4 and 8, respectively, where both values are applied to the formula $k_G(\mathbf{x}_i, \mathbf{x}_j) = \gamma \exp(-||\mathbf{x}_j - \mathbf{x}_i||^2/2\sigma^2)$.
- Gaussian process classifier with Matern kernel, the best values for nu, σ and γ are 1.5, 3 and 1.5×10^{-5}, respectively, where these values are applied to the formula $k_M(\mathbf{x}_i, \mathbf{x}_j) = \gamma \frac{1}{\Gamma(\nu)2^{\nu-1}} \big(\frac{\sqrt{2\nu}||\mathbf{x}_j - \mathbf{x}_i||^2}{\sigma}\big)^{\nu} K_{\nu}\big(\frac{\sqrt{2\nu}||\mathbf{x}_j - \mathbf{x}_i||^2}{\sigma}\big)$, where $K_{\nu}(\cdot)$ and $\Gamma(\cdot)$ are the modified Bessel function and the

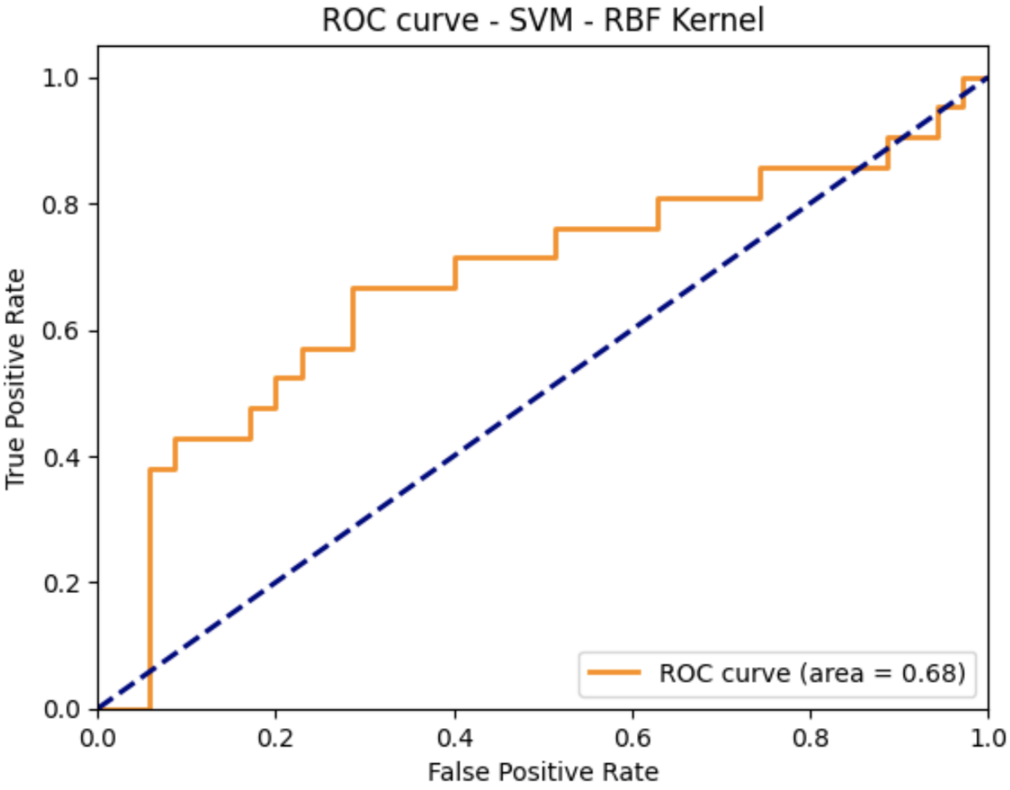

Figure 6. The ROC curve for support vector machine with radial basis function kernel

gamma function, respectively.

- Gaussian process classifier with the combination between radial basis function and Matern kernel as follows: $k(\mathbf{x}_i, \mathbf{x}_j) = \gamma_G k_G(\mathbf{x}_i, \mathbf{x}_j) + \gamma_M k_M(\mathbf{x}_i, \mathbf{x}_j)$, where γ_G and γ_M are 8 and 1.5×10^{-5}, respectively. The hyperparameter values used in the two previous kernels are also used in this combination.
- Gaussian process classifier with dot product kernel, which is defined a follows: $k_d(\mathbf{x}_i, \mathbf{x}_j) = 1 + \langle \mathbf{x}_i, \mathbf{x}_j \rangle$.
- Gaussian process classifier with periodic kernel, where $sigma$ and p (periodicity) are 2^{-16} and 3, respectively. The periodic kernel is defined as follows: $k_p(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\frac{2\sin^2(\pi||\mathbf{x}_j - \mathbf{x}_i||^2/p)}{\sigma^2}\big)$.
- Gaussian process classifier with rational quadratic kernel, where σ and α are both 3×10^5. The kernel is defined as follows: $k_r(\mathbf{x}_i, \mathbf{x}_j) = (1 + ||\mathbf{x}_j - \mathbf{x}_i||^2/(2\alpha\sigma^2))^{-\alpha}$
- SVM with radial basis function kernel, where γ and C are 3.9×10^{-3} and 2, respectively. In this case the kernel is defined as follows: $k_G(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_j - \mathbf{x}_i||^2)$.
- SVM with polynomial kernel, where d (degree) and C are 3 and 4096, respectively. The kernel is defined as follows: $k_p(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle^d$.
- SVM with sigmoid kernel, where γ and C are 4.88×10^{-4} and 32768, respectively. The kernel is defined as follows: $k_s(\mathbf{x}_i, \mathbf{x}_j) = \tanh \gamma \langle \mathbf{x}_i, \mathbf{x}_j \rangle$.
- Multilayer perceptron with a single hidden layer, with 600 neurons in the hidden layer. This was fitted with an initial learning rate and regularization parameter equal to 10^{-4} and 10^{-2}, respectively. The activation function used in the hidden layer is hyperbolic tangent function.
- Multilayer perceptron with five hidden layers. The number of neurons in the first, second, third, fourth, and fifth layer are 600, 300, 100, 300, and 600, respectively. This was fitted with an initial learning rate and regularization parameter equal to 10^{-4} and 10^{-2}, respectively. The activation function used in the hidden layer is hyperbolic tangent function.
- Logistic regression classifier was fitted with a regularization parameter of 10^{-2}.
- The decision trees were fitted using both the Gini and entropy indexes. The parameters used were the given by default in Scikit-Learn API.
- XGBoost algorithm were fitted with a learning rate, maximum depth, and number of estimators equal to 3.13×10^{-2}, 5, and 50, respectively. Besides, we used the entropy index in the trees.
- Adaboost algorithm were fitted with a learning rate and number of estimators equal to 0.13 and 110, respectively. Besides, we used the entropy index in the trees.
- Random forest were fitted with 15 trees (with entropy index), at least one sample per leaf, minimum three samples per split, and a maximum depth of nine levels.

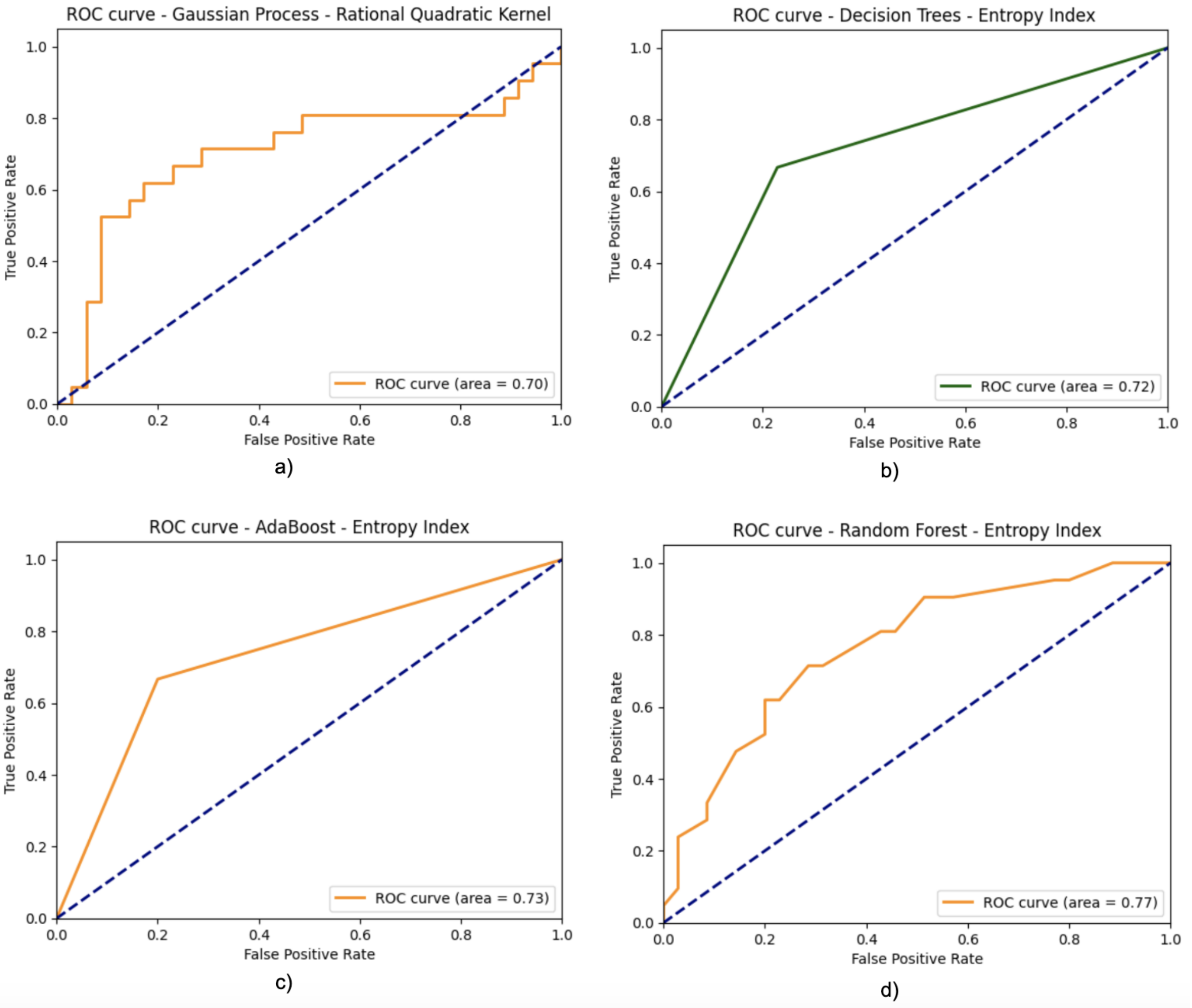

Figure 7. The ROC curve for a) Gaussian process classifier with Rational quadratic kernel, b) decision tree with entropy index, c) Adaboost, and d) random forest

C. Discussion

Based on the evaluation results, we found that the Support Vector Machines (SVM) method with radial basis function kernels was the most accurate classifier tested in this study. However, when we conducted a paired t-test, we found statistically significant evidence that SVM outperformed multilayer perceptrons and Gaussian processes with dot product and periodic kernels, but we did not find strong statistical evidence that SVM was more accurate than all the other classifiers.

In addition, the paired t-test showed that the harmonic mean (F_1) of SVM was significantly better than that of multilayer perceptrons and Gaussian process classifiers with the periodic kernel. However, we did not find strong statistical evidence for a significant difference between the harmonic mean of SVM and the other evaluated classifiers.

Moreover, we observed that the multilayer perceptron and the Gaussian process classifier with the periodic kernel had the poorest performance amongst the classifiers tested in this study. It is possible that increasing the dataset size might improve the performance of the multilayer perceptron. In contrast, the poor performance of the Gaussian process classifier with the periodic kernel might be due to the model's inability to repeat itself exactly.

Whilst the random forest classifier achieved the highest precision amongst the classifiers tested in this work, the paired t-test showed no significant difference between its precision and that of SVM with radial basis function kernels. However, random forest performed better than SVM in distinguishing between positive and negative examples, as evidenced by its higher area under the ROC curve (AUC) of 0.77, compared to SVM's AUC of 0.68. This difference is shown in Figures 6 and 7, and suggests that random forest is better at accurately identifying true positives and true negatives than SVM, even though SVM is slightly more accurate overall.

Furthermore, the decision trees that make up the random forest classifier can be used to extract insights and discover knowledge that can help formulate theories about how a student's performance in prerequisite courses might influence their performance in numerical methods. By analysing the tree structure and the features that lead to high or low performance, we can gain a better understanding of the underlying relationships between these variables and potentially develop new strategies for improving student outcomes.

The Gaussian process classifier with the radial basis function kernel is another strong candidate for classification, as it was the second most accurate method according to our results. One advantage of this classifier over SVM is that it provides a probability estimate of a given student being at risk, which can be useful for making decisions. In contrast, SVM method does not provide such a probability estimate.

To draw a more solid conclusion about the best classifier for this problem, we would need to collect additional data to evaluate the performance and generalization capability of the classifiers.

V. Conclusions and Future Work

In this work, we have studied several machine learning methods for forecasting if a given student is at risk of failing or withdrawing from the Numerical Methods course based on their performance in prerequisite courses.

The aforementioned study was conducted using a dataset with 56 records and 39 variables each (i.e., 38 independent variables and one target variable), collected from students who have attended courses from the fifth semester up to ninth semester during the second half of 2022. The findings of the study conducted in this work are as follows:

- Support vector machine (SVM) with the radial basis function is more accurate than the other studied methods, reaching the mean values for accuracy, precision, recall, and harmonic mean (F_1) of 76.67%, 71.67%, 51.67%, and 57.67%, respectively.
- Whilst there is no strong statistical evidence that SVM is more accurate than the other studied methods, there is solid evidence that SVM with the radial basis function outperforms the multilayer perceptron with one or several hidden layers, as well as the Gaussian process classifier with periodic and dot product kernels.
- SVM performs better with the radial basis function kernel than with polynomial and sigmoid kernels.
- Gaussian process (GP) performs better with the radial basis function kernel than with the other kernels evaluated in this study.
- GP with the radial basis function is the second classifier more accurate according to the evaluation conducted in this study.
- During the evaluation, Random forest (RF) was found to be the third more accurate classifier, with the highest mean recall amongst all methods.
- There is no statistically significant difference between the mean recall of RF and SVM with the radial basis function.

For further work we recommend:

- Collect more data to study the performance of some learning methods such as, e.g., multilayer perceptron, random forest, Adaboost, and so forth.
- Combine more kernels in the Gaussian processes to study their performance.
- Extend this study to other courses besides numerical methods.
- Analyse the decision trees generated by random forest, Adaboost, XGBoost, and other methods in more detail, with the aim of identifying patterns or rules that can help us gain deeper insights into the problem at hand.
- Propose a novel method that surpasses the performance of all previously studied methods in this work, achieving significantly higher accuracy and harmonic mean scores.
- Despite the dataset being sufficiently large, leading to improved performance of most classification methods compared to random guessing, with one of them achieving an area under the ROC curve (AUC) of 0.77, we shall collect a larger dataset for further research. By doing so, we might draw more robust conclusions regarding the performance of the classifiers.
- Investigate the effectiveness of strategies and precautions, such as, e.g., mentoring programs and personalized feedback, in coping with the risks of failure and dropout faced by students.

Acknowledgement

Caicedo-Castro thanks the Lord Jesus Christ for blessing this project. We thank Universidad de Córdoba in Colombia for supporting the Course Prophet Research Project (grant FI-01-22). In special, our deepest appreciation goes to Dr. Jairo Torres-Oviedo, rector of the University of Córdoba, who has always helped us and supported us throughout this research. We thank all students who collaborated with us, answering the survey conducted for collecting the dataset, used to train the learning algorithms adopted in this research. Finally, we thank the anonymous reviewers for their comments that contributed to improve the quality of this article.

References

[1] I. Pacheco-Arrieta *et al.* (2004) Agreement No. 004: Student's code at the University of Córdoba in Colombia. Retrieved on May 24. [Online]. Available: http://www.unicordoba.edu.co/wp-content/uploads/2018/12/reglamento-academico.pdf

[2] C. Demetriou and A. Schmitz-Sciborski, "Integration , Motivation , Strengths and Optimism : Retention Theories Past , Present and Future," in *Proceedings of the 7th National Symposium on Student Retention*, 2011, pp. 300–312.

[3] I. Lykourentzou, I. Giannoukos, V. Nikolopoulos, G. Mpardis, and V. Loumos, "Dropout prediction in e-learning courses through the combination of machine learning techniques," *Computers and Education*, vol. 53, no. 3, pp. 950–965, 2009.

[4] J. Kabathova and M. Drlik, “Towards Predicting Student’s Dropout in University Courses Using Different Machine Learning Techniques,” *Applied Sciences*, vol. 11, p. 3130, 04 2021.

[5] J. Niyogisubizo, L. Liao, E. Nziyumva, E. Murwanashyaka, and P. C. Nshimyumukiza, “Predicting student’s dropout in university classes using two-layer ensemble machine learning approach: A novel stacked generalization,” *Computers and Education: Artificial Intelligence*, vol. 3, p. 100066, 2022.

[6] D. E. M. da Silva, E. J. S. Pires, A. Reis, P. B. de Moura Oliveira, and J. Barroso, “Forecasting Students Dropout: A UTAD University Study,” *Future Internet*, vol. 14, no. 3, pp. 1–14, February 2022.

[7] V. Čotić Poturić, I. Dražić, and S. Čandrlić, “Identification of Predictive Factors for Student Failure in STEM Oriented Course,” in *ICERI2022 Proceedings*, ser. 15th annual International Conference of Education, Research and Innovation. IATED, 2022, pp. 5831–5837.

[8] V. Čotić Poturić, A. Bašić-Šiško, and I. Lulić, “Artificial neural network model for forecasting student failure in math course,” in *ICERI2022 Proceedings*, ser. 15th annual International Conference of Education, Research and Innovation. IATED, 2022, pp. 5872–5878.

[9] S. Zihan, S.-H. Sung, D.-M. Park, and B.-K. Park, “All-Year Dropout Prediction Modeling and Analysis for University Students,” *Applied Sciences*, vol. 13, p. 1143, 01 2023.

[10] A. V. Dorogush, V. Ershov, and A. Gulin, “CatBoost: gradient boosting with categorical features support,” *CoRR*, vol. abs/1810.11363, 2018.

[11] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[12] D. Cox, “The regression analysis of binary sequences,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 20, no. 2, pp. 215–242, 1958.

[13] D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” *Mathematical Programming*, vol. 45, no. 3, (Ser. B), pp. 503–528, 1989.

[14] R. Byrd, P. Lu, J. Nocedal, and C. Zhu, “A Limited Memory Algorithm for Bound Constrained Optimization,” *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[15] C. Williams and C. Rasmussen, “Gaussian Processes for Regression,” in *Advances in Neural Information Processing Systems*, D. Touretzky, M. Mozer, and M. Hasselmo, Eds., vol. 8. MIT Press, 1995, pp. 514–520.

[16] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[17] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, 2nd ed. The MIT Press, 2018.

[18] C. Cortes and V. Vapnik, “Support Vector Networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995.

[19] C. C. Aggarwal, *Neural Networks and Deep Learning*. Springer, 2018.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Representations by Back-propagating Errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[21] D. P. Kingma and J. Ba. (2014) Adam: A method for stochastic optimization. Retrieved on May 25. [Online]. Available: http://arxiv.org/abs/1412.6980

[22] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[23] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” in *ICML*, vol. 96, 1996, pp. 148–156.

[24] L. Breiman, “Random forests,” in *Machine learning*, vol. 45, no. 1. Springer, 2001, pp. 5–32.

[25] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.

[26] I. Caicedo-Castro. (2023) Dataset for Early Forecasting of At-Risk Students of Failing or Dropping Out of a Bachelor’s Course Given Their Academic History - The Case Study of Numerical Methods. Retrieved on May 25. [Online]. Available: https://sites.google.com/correo.unicordoba.edu.co/isacaic/research

[27] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[28] (2017) Google Colaboratory. Retrieved on May 25. [Online]. Available: https://colab.research.google.com/