


Article

# A Learning Analytics Approach to Identify Students at Risk of Dropout: A Case Study with a Technical Distance Education Course

Emanuel Marques Queiroga <sup>1,2,\*</sup> , João Ladislau Lopes <sup>2</sup> , Kristofer Kappel <sup>1</sup> ,  
Marilton Aguiar <sup>1</sup> , Ricardo Matsumura Araújo <sup>1</sup> , Roberto Munoz <sup>3,\*</sup> ,  
Rodolfo Villarroel <sup>4</sup>  and Cristian Cechinel <sup>5</sup> 

<sup>1</sup> Centro de Desenvolvimento Tecnológico (CDTEC), Universidade Federal de Pelotas (UFPel), Pelotas 96010610, Brazil; kristofer.kappel@gmail.com (K.K.); marilton@inf.ufpel.edu.br (M.A.); ricardo@inf.ufpel.edu.br (R.M.A.)

<sup>2</sup> Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-Grandense (IFSul), Pelotas 96015560, Brazil; joao.lblopes@gmail.com

<sup>3</sup> Escuela de Ingeniería Informática, Universidad de Valparaíso, Valparaíso 2362735, Chile

<sup>4</sup> Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Valparaíso 2362807, Chile; rodolfo.villarroel@ucv.cl

<sup>5</sup> Centro de Ciências, Tecnologias e Saúde (CTS), Universidade Federal de Santa Catarina (UFSC), Araranguá 88906072, Brazil; contato@cristiancechinel.pro.br

\* Correspondence: emanuelmqueiroga@gmail.com (E.M.Q.); roberto.munoz@uv.cl (R.M.)

Received: 1 May 2020; Accepted: 28 May 2020; Published: 9 June 2020



**Abstract:** Contemporary education is a vast field that is concerned with the performance of education systems. In a formal e-learning context, student dropout is considered one of the main problems and has received much attention from the learning analytics research community, which has reported several approaches to the development of models for the early prediction of at-risk students. However, maximizing the results obtained by predictions is a considerable challenge. In this work, we developed a solution using only students' interactions with the virtual learning environment and its derivative features for early predict at-risk students in a Brazilian distance technical high school course that is 103 weeks in duration. To maximize results, we developed an elitist genetic algorithm based on Darwin's theory of natural selection for hyperparameter tuning. With the application of the proposed technique, we predicted the student at risk with an Area Under the Receiver Operating Characteristic Curve (AUROC) above 0.75 in the initial weeks of a course. The results demonstrate the viability of applying interaction count and derivative features to generate prediction models in contexts where access to demographic data is restricted. The application of a genetic algorithm to the tuning of hyperparameters classifiers can increase their performance in comparison with other techniques.

**Keywords:** at-risk students; genetic algorithm; learning analytics; educational data mining

## 1. Introduction

Learning analytics (LA) approaches have emerged in the context of the increasing use of digital information and communication technologies in education [1]. LA provides information and knowledge so that institutions can overcome core challenges with the qualification of their teaching and learning processes [2,3]. Student dropout is one of the main problems in e-learning that has received considerable attention from the research community. Early detection of students at risk of dropout plays an essential role in reducing the problem, enabling targeted actions aimed at specific situations [4–6].

According to OECD [7], contemporary education is vast, and there are many concerns about the performance of education systems. Among the various important challenges faced in education, one of the most difficult to tackle is the low completion rates observed in many institutions [8], being the final representation of the high dropout rates and low student performance in courses. These problems are related to many factors other than teaching methodologies, such as the profile of the students and their ability to self-manage time [7,9,10].

Dropout rates in e-learning are generally higher compared with face-to-face education [8]. According to the European Commission on Education and Culture, countries like Poland, Sweden, and Hungary have dropout rates in higher education of 38%, 47%, and 47%, respectively [11]. In Spain, the dropout rate is 50% at the Spanish National Distance Education University (UNED) [12]. In Brazil, the enrolment numbers significantly increased in the last few years, but student dropout rates simultaneously increased. The last census of distance education in Brazil [13] reported dropout rates of 50% in the distance courses offered by the Ministry of Education.

Studies have established that student success in distance courses is directly correlated with their engagement inside virtual learning environments (VLEs). Distance learning technology allows tutors to measure the engagement of students by looking into system logs and evaluating the intensity of students' interactions in the different activities available inside virtual classrooms [9,14,15].

In the educational context, access to data is a considerable challenge [16]. The distribution of institutional and academic data across numerous systems creates challenges for accessing social-demographic and previous academic data. This occurs typically because VLEs are usually unprepared for the storage of this kind of data and because several educational institutions apply different learning modalities, such as face-to-face learning, hybrid learning, and distance learning, thus requiring a central academic system. Data are usually concentrated in a central academic system that has no direct connection with the virtual environment. This situation restricts the automated retrieval of data external to VLE, for example, to use in dashboards for data visualization or in the generation of predictive models. In many institutions, the access to use this kind of data is restricted either due to internal policies or data access legislation [16–18].

One of the main advantages of distance learning courses is the large amount of data generated by interactions between students and the system, which provides new possibilities for studying and understanding the data. In e-learning courses, the interaction between students and teachers is usually mediated by a VLE. Thus, VLEs generate a large volume of data that can be consumed by machine learning models [19]. Machine learning algorithms have been used to build successful classifiers using diverse student attributes [10]. While these models showed promising results in several settings, these results are usually attained using attributes that are not immediately transferable to other courses or platforms.

In machine learning, the parameters are defined by the model generated by the algorithms, unlike regular programming, where the term *parameter* is used to refer to the entry of a given function. The final accuracy of models is directly linked to the quality of the fine-tuning of their hyperparameters on the algorithm input [20]. Thus, more adjusted hyperparameters result in more accurate models [21,22]. The control variables of the classifiers are called hyperparameters, which aim to define relevant issues regarding the model to be trained, such as the number of estimators in a random forest algorithm or the number of layers hidden in a neural network [21]. In a neural network context, parameters are adjusted during the training phase using weights. Hyperparameters are variables set before the training, such as the network topology or learning information [22].

In this context, we previously proposed exploiting students' interaction counts solely over time (and other attributes derived from the counts) to predict at-risk students [23–25]. This approach was tested and produced good results, allowing the early prediction of students at risk of dropout and achieving overall accuracies varying from 65% to 90% in the first eight weeks of a two-year distance courses. These studies produced results comparable to those in the literature. For instance,

Jayaprakash et al. [26] obtained general accuracies varying from 73% to 94% and Manhães et al. [4] reported accuracies from 62.22% to 67.77%.

Maximizing the results obtained by predictions is a considerable challenge [27], as the different algorithms commonly present a wide variation in the performance rates that depend on the combination of several characteristics (e.g., balance among classes, amount of data, input variables, and others) and algorithm hyperparameters [28]. Evolutionary computation, and especially genetic algorithms (GAs), are used for optimization problems and tuning classifiers in several areas such as medicine [20] and emotion recognition [29], producing significant results. Here, we propose the use of an evolutionary GA to tune the hyperparameters of the classifiers, thereby optimizing the performance of the models for the early detection of students at risk of dropping out.

This paper is a continuation of these previous works, now aiming to enhance the results by applying an approach that uses GAs to tune machine learning algorithms' hyperparameters. This paper contrasts the results of two methods for hyperparameter optimization applied on models to detect at-risk students in technical e-learning courses based on the counting of students' interactions inside the VLE. The first method for hyperparameter optimization is based on a GA created by the authors, and the second is the traditional widely used method called grid search [21]. During this study, we aimed to answer the following research questions:

- RQ1.** Does the GA approach to hyperparameter optimization outperform traditional techniques?
- RQ2.** Does the resulting predictive models generated by the use of the GA approach for hyperparameter optimization perform better than models with default hyperparameters?

The remainder of this paper is organized as follows: Section 2 presents the theoretical background and related work about the problem of predicting at-risk students and the use of GAs in this context. Section 3 presents the case study conducted to test the proposed solution, detailing the data gathered, the methodology, the proposed GA for fine-tuning, and the experiments. Section 4 discusses the results, and Section 5 concludes the paper and proposes future work.

## 2. Theoretical Background

This section presents works focused on predicting at-risk students in different scenarios and the use of hyperparameter techniques to improve results. Several works in the field of learning analytics and educational data mining deal with the problem of early predicting at-risk students. The works usually differ according to several aspects, such as (1) the sources of data used to generate the models for prediction (demographic, VLEs, surveys, exams); (2) the level of education of the courses (high school, secondary education); (3) the goal of the predictive models (e.g., to predict performance or evasion); (4) the scope of the prediction focused on an entire program or a specific course or discipline; (5) the modality of the course (formal or informal, face-to-face, blended, or distance learning); and (6) whether or not to use tuning techniques for classifiers.

According to Liz-Domínguez et al. [30], data analysis is the set of techniques used to transform data into information and knowledge, thus revealing correlations and hidden patterns. The data resulting from this process can be used to create early warning systems to predict future events. This process mainly aims to support learning and mitigate some of the problems, such as academic performance, retention, and dropout. The reliability of the predictions by the predictor is one of the main factors established by Liz-Domínguez et al. [30] and Herodotou et al. [31] for their application on a large scale.

According to Liz-Domínguez et al. [30], researchers have experimented with methodologies in different scenarios. However, according to Hilliger et al. [32] and Cechinel et al. [33], in Latin America, these studies are mainly concentrated in the university context, so more applications in other contexts are necessary.

González et al. [34] demonstrated that information and communication technologies have a greater impact on the teaching and education process. González et al. [34], de Pablo González [35]

demonstrated the significant impact of the use of VLEs by teachers on student learning. This impact can be maximized using intervention methods based on machine learning, as proposed by Herodotou et al. [36]. Herodotou et al. [31] demonstrated that the classes where teachers used predictive methods produced a performance at least 15% higher than the classes without that use. This improvement was also observed in comparison with classes with the same teachers but from previous years.

In the educational context, traditional research usually uses data from educational systems and virtual environments. The research by Zohair [37] proposed only using data from the academic system (e.g., extracurricular courses, grades, and age) to predict performance in graduate students. Some of the extracted data were extracurricular courses taken and the respective grades, initial training course, and descriptive data about the grades and the age of the student. This study demonstrated that for small groups of students, this is a logical approach that produces good results with few pre-processing steps and a limited set of data. The author focused on the use of algorithms that perform well with low amounts of data, such as support vector machines and multilayer perceptrons (MLP), that produce results with accuracy above 76%.

The search for methods that can be generalized and therefore replicable for other courses represents a significant portion of the research. Thus, studies such as [38] proposed an architecture that is not dependent on a single type of datum, working with the flow of clicks that academics make in a Massive Open Online Course (MOOC). To do so, data are captured from a course and different prediction models are trained and tested in other courses and environments. The experiments showed 87% accuracy when testing in different courses and 90% when tested in the same course, not varying significantly according to the environment.

In [39], several techniques for pre-processing data were compared in terms of interactions with the virtual environment Moodle in risk prediction. Data from the plugin Virtual Programming Laboratory (VPL) were used for risk prediction in algorithm and programming disciplines in undergraduate courses. Data such as weekly interaction count, an average of interactions, median, number of weeks without interactions, standard deviation, and commitment factor are generated based on a previously proposed technique [25,40]. Data added included the teacher interaction count, social count, and cognitive count based on a proposed theory Swan [41]. With naturally unbalanced data, the synthetic minority over-sampling technique (SMOTE) was applied to create balance. Several datasets were generated with different variables to compare the techniques. The results demonstrated that the use of only the interaction count as proposed in [24,25] presented results superior to the other techniques, including their union.

For instance, [5] proposed a students' dropout prediction system that combines outcomes from three different algorithms (neural network, support vector machine (SVM), and probabilistic ensemble simplified fuzzy Adaptive Resonance Theory (ARTMAP—PESFAM)). The authors gathered static demographic data, like sex and place of residence; academic data, like performance and scholar degree; and dynamic data, such as the number of interactions in the virtual environment, grades, and even delivery dates of activities. After applying the algorithms, three distinct approaches to the dropout prediction were generated: (1) A student is considered a dropout case if at least one method classified them as such, (2) a student is considered as a dropout if at least two methods indicated the student to be a dropout and, (3) the student is only presumed as a dropout if all three techniques classified them as a dropout. The accuracy of the results obtained ranged from 75% to 85%, and the best results were achieved using the less restrictive approach, the first one, which achieved accuracies up to 85% on the first section of a given course.

Jayaprakash et al. [26] proposed a warning system focused on student performance to reduce dropout and retention rates. The system provides the student with updated feedback on their potential scholarly performance. To do so, the system uses several types of data, such as demographic (sex and age), student interactions on the VLE, previous academic performance, time passed since the student entered the university, online time spent on the VLE, and outcomes from the scholastic aptitude

test (SAT) (verbal and math). Different models of prediction were produced using J48, Bayesian networks with naive Bayes, SVM with minimal sequential optimization (SMO), and logistic regression, considering data from 9938 students. These classifiers presented similar results, with the classifier based on logistic regression producing slightly superior outcomes (94.2% general accuracy and 66.7% precision for identifying students at dropout risk).

A classifier able to early predict student dropout using students' interactions inside a VLE was proposed [42]. They used information such as if the student watched all video tutorials, if the student ignored some given material or activity, if the student was delayed in following the virtual classes, and the student performance in the activities. Students were then classified according to three flags: Green (low dropout risk), yellow (medium dropout risk), and red (high dropout risk). The authors did not mention the types of machine learning algorithms used but reported performance (TP accuracy) varying from 40% to 50% to predict dropout students within two weeks in advance.

Genetic algorithms are widely used in data mining and can be implemented as the classifier or as a result of the optimizer, as proposed in this approach. One of the applications of genetic algorithms for optimization is a method combining the predictions generated by classifiers. To this end, Minaei-Bidgoli and Punch [43] proposed the application of machine learning to predicting student performance in an online physics course at Michigan State University. For this, data derived from the tasks performed by the students were used. Ten different variables were extracted, including success rate, success on the first attempt, the number of attempts, the time between task delivery and deadline, the time involved in solving, and the number of interactions with colleagues and instructors. A principal component analysis (PCA) method was applied to transform the variables, and three different sets with two, three, or nine components were generated. After this, the Bayes classifier, I-nearest neighbor (I-NN), k-nearest neighbor (k-NN), Parzen-window, multilayer perceptron (MLP), and decision tree classifiers were applied. Then, the predictions obtained by the classifiers were combined with the genetic algorithm using 200 individuals with 500 generations. The GA proposed by the author achieved optimization of 10% to 12% depending on the number of components in the input.

Márquez-Vera et al. [6] proposed the evolutionary algorithms Interpretable Classification Rule Mining Algorithm (ICRM) [27] and ICRM2 [6] based on grammar-based genetic programming (GBGP). In Márquez-Vera et al. [6], ICRM was used to predict the dropout of high school students in Mexico. The authors proposed a double-approach prediction on the same algorithm, creating two classification rules: One for identifying students who tend to complete the course and the other for students who tend to drop out. The data used included 60 attributes that range from the entrance test to research data distributed to students. As a comparison method, the algorithm proposed by the author was compared with five classifiers: Naive Bayes, decision tree, Instance-based lazy learning (IBK), Repeated Incremental Pruning (JRip), and SVM. Techniques were also used to reduce the dimensionality of the base. Using the accuracy as an evaluation metric, the results obtained by the proposed algorithm showed that it can be a valid approach, especially considering the ease of interpretation of the generated classification rules.

### 3. Proposed Approach

The proposed approach consists of the use of a GA for the classifier (hyperparameter) optimization and selection of the fittest, to predict dropout in distance learning courses. Figure 1 shows the proposed solution. The following machine learning algorithms were selected to test the solution: Classic decision tree (DT), random forest (RF), multilayer perceptron (MLP), logistic regression (LG), and the meta-algorithm AdaBoost (ADA). The proposed approach was compared against the grid search method regarding hyperparameter optimization and the regular solution without hyperparameter optimization. The proposed approach uses a classification method, where several classifiers with different hyperparameters, such as DT, RF, MLP, LG, and ADA, compete against each other. In the end, the classifier and the hyperparameters with the best results are selected by a fitness function.

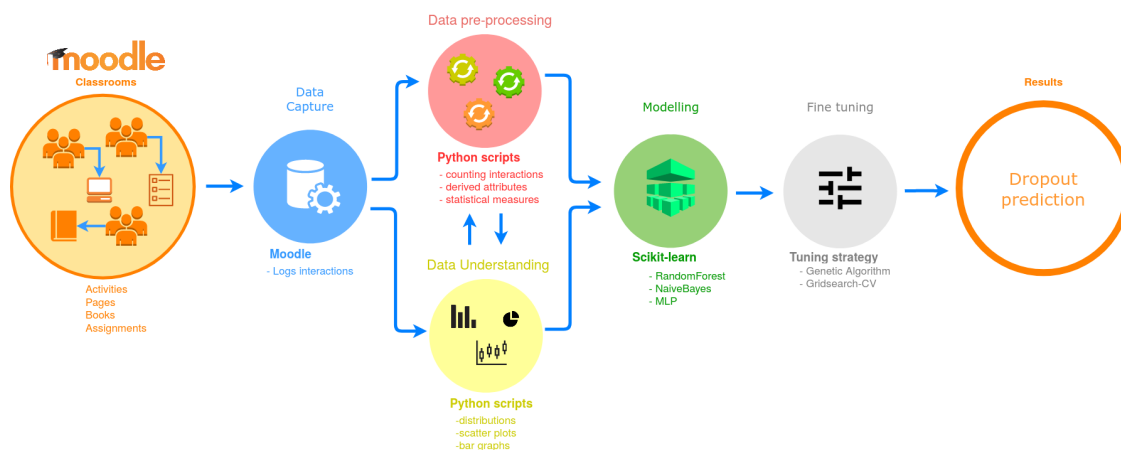


Figure 1. Proposed approach.

### 3.1. Case Study

The case study consisted of the following steps: Data capture, data pre-processing, data understanding, and modelling, according to the solution proposed in the Figure 1. These steps occur in parallel, with tests, implementations, and generation of new features for developing models for the early prediction of at-risk students in a technical distance learning course. The methodology to generate the models relies on the counting of interactions of the students inside the VLE, with the use of the proposed solution described in the previous section.

Data related to the student’s interactions were collected from the logs of the institutional Moodle platform of a given technical distance course of the Instituto Federal Sul Rio-grandense (IFSul) in Brazil. Table 1 shows the number of logs collected, the number of students enrolled in the course, and the percentages of dropout and success. The course is taught in 18 different cities throughout the state of Rio Grande do Sul and involves weekly activities that are posted on the VLE by the teacher. Students have one week to develop the activities with the help of tutors. The course has a maximum completion time of 103 weeks, with a total workload of 1215 h divided into disciplines. The maximum duration is 24 months, with three breaks also called vacations, and the student’s final situation is determined by their performance in the evaluations and their re-enrolment every six months.

The maximum term for completion of the curriculum is four years, and the student may repeat each discipline only once and, therefore, the year. The student has the option of taking up to two subjects for the next year and taking them concurrently with the others. For approval, the student must have a grade of six or higher in each of the disciplines of the curriculum. Students who spend 365 days without interactions with the virtual environment or do not perform their annual re-enrolment are considered absent and are removed from the course. Thus, the student receives a grade from 0 to 10 at the end of a given discipline, and one of two states is associated with the student: Approved or failed. However, we aimed to predict students who drop out during the course. For this, the student will be considered dropped out if they leave, do not perform the activities during the course, and their enrolment in the following semester.

Table 1. Dataset summary.

Number of Log Rows	Number of Students	Dropouts (%)	Success (%)
1,051,012	752	354 (47%)	398 (53%)

The choice to only use data from the counting of interactions was motivated by previous research that achieved satisfactory results using the same approach [23,25]. This choice was also related to limitations on capturing other kinds of data for the present study. In previous works, we sought to create models that are easy to generalize so that they could be applied to other courses. To accomplish

that, we used four courses, where the model created by one was applied to the others, and the models generated with data from three courses were applied to the remaining one. In these experiments, the labeling of the type of interaction was tested and did not show significant results. When testing the models generated with data from one course on data from other courses, this type of labeling negatively impacted the results.

Studies such as Macarini et al. [39] tested the application of different types of interactions and derived data, with their labeling showing no significant differences in performance. Thus, we applied the methodology that presented the best previous results to model other courses in the same educational context, even if the model is derived from data from one course only.

The courses studied here are offered in several cities throughout the interior of Brazil and present a large demographic diversity. Nowadays, the collection of demographic data is a task manually performed by eighteen different teaching centers through a printed questionnaire that is sent to IFSul after completion. This process generates a series of problems, such as lack of data, reading and typing problems, and consequently low diversity and inconsistencies. These factors led to the lack of reliability in these data and their consequent non-use.

Data capture consisted of collecting raw data from student interactions with Moodle VLE. The data initially had the format presented in Table 2. After selection, data were validated. This stage consisted of comparing the student situation data in the VLE to the data on the institutional academic system. Both systems are independent and have no integration. Cases of inconsistency were handled manually by checking other types of internal control.

**Table 2.** Information contained in the log files.

Column	Comment
Course	Name of the virtual classroom accessed
Time	Day and time of the access
IP Address	IP Address of the machine
Full name	User (student) name
Action Event Name	The action represents the type of interaction that the student performed in the classroom. For instance: (1) Visualization and participation on chats; (2) Visualization and inclusion of posts in forums; (3) Visualization of resources; and (4) Visualization of the course.
Description	Detailed description of the event. Example: Download the .pdf file.

The course format analyzed in this project consists of 103 weeks divided over two years. As stated by [5], early identification of a risk situation is a fundamental criterion for its reversal. Thus, for this work, we chose to use the methodology based on [4], which consists of the application of data mining on the data of the first subjects of the course. Using this process, we chose to use data from the 50 weeks that compose the first year of the course. Every two weeks starting from the fourth, a prediction model was generated, so the approaches used in this work created 23 models in the period.

After validation, data were anonymized and preprocessed, and variables were generated (features extraction). Table 3 describes the variables extracted to be used as the input for training and testing the predictive models. The table shows that all variables were based on the counting of students' interactions inside the VLE. Figure 2 exemplifies the behavior of the Weekly interactions variable for some weeks of the course and according to the Student Final Status category.

Exploratory data analysis (EDA) seeks to visualize dataset information to better understand the student's behavior when using the VLE. Table 4 shows how dropout rates evolved after every 10 weeks of the course until week 50. The table also shows the dropout rates for the first and second year of

the course after week 50. We considered a student as dropped out after a period of six weeks without interactions with the VLE. The idea here was to pinpoint the period where the departure occurs.

The evasion rates between the two years of the course are practically the same (182 dropouts for year 1 and 172 dropouts for year 2). However, if we look proportionally at the number of students enrolled at the beginning of each year, the dropout rate is slightly higher in the second year, with 30.06% compared with 24.20% in the first year. These values differ from the average dropout rates known from higher institutions in Brazil [44] as well as from secondary and technical schools [45]. Unfortunately, there are no national data related to the distance learning modality to enable a more precise comparison.

A total of 86.81% of the course dropouts of the first year are concentrated in the first 20 weeks (152 dropouts of the 182 in the first year). This shows a tendency of the students to leave at the very beginning of the course, which could be related to difficulties faced in the initial studies. This tendency is also reported in the literature in relation to face-to-face courses where difficulties in the beginning of the course are reported as the most critical factor leading students to drop out.

Figure 2 presents the bi-weekly total count, the means, and the standard deviations of the students' interactions. In the figure, students identified as dropped out in a given week are not counted in the following weeks. As shown in the figure, dropout students present a higher number of interactions than successful students until week 13. One possible explanation for this behavior is that those students are experiencing difficulties during their learning process, so they interact more with the VLE to obtain assistance. The total count of interactions per group is lower for the dropout group (considering the whole period). Figure 3 presents a boxplot of the counting of interactions for each group of students, which highlights the differences in these groups regarding the use of the VLE.

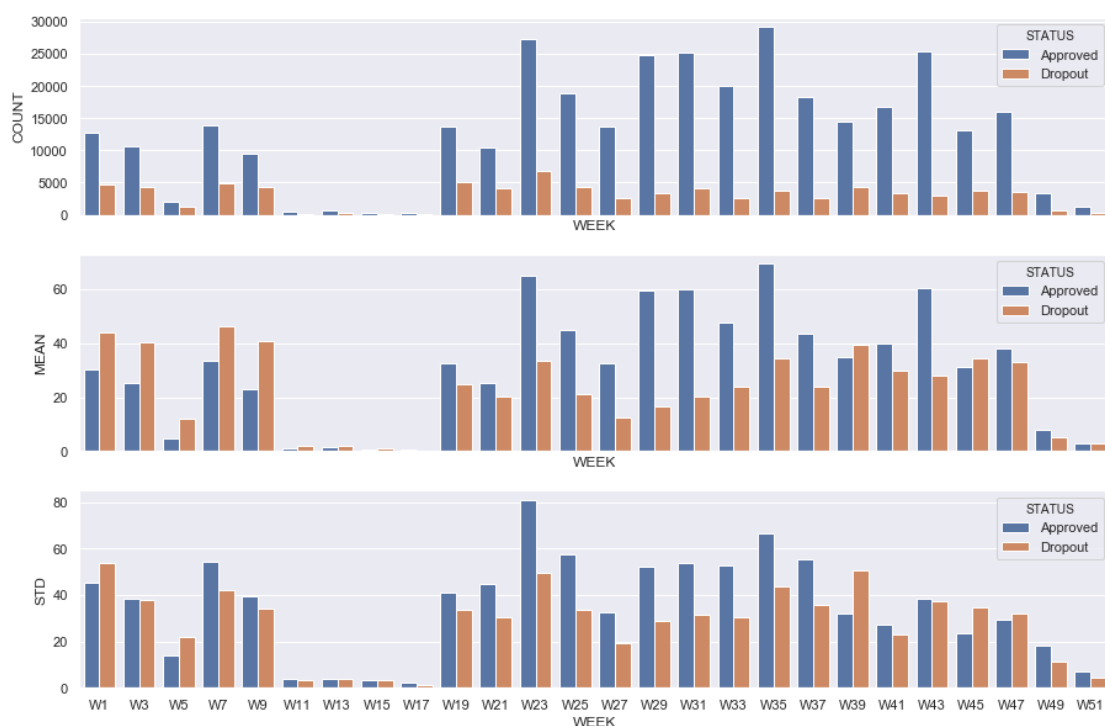


Figure 2. Interactions every two weeks.



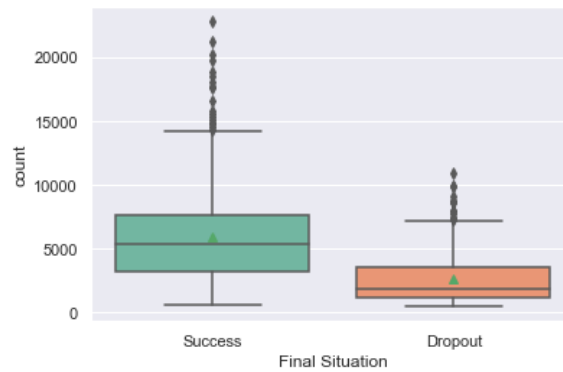


Figure 3. Boxplot success X dropout.

Table 3. Features extracted to be used as input for the models.

Variable	Description
Daily interactions	Count of interactions of a given day (from 1 to 350 days)
Weekly interactions	Count of interactions of a given week (from 1 to 50 weeks)
Mean of the week	Average of the count of interactions of a given week
Standard deviation of the week	Standard deviation of the count of interactions of a given week
Student final status	Dependent variable representing the student final status: Dropout or success

Table 4. Evolution of dropout during the course.

Year	Period	Number of Students in Course	Number of Dropout Students (NDS)	NDS Rate	Accumulated NDS	Accumulated NDS Rate
Year 1	Week 10	752	87	11.56	87	11.56
	Week 20	665	71	10.67	158	21.01
	Week 30	594	21	3.5	179	23.27
	Week 40	573	1	0.17	180	23.4
	Week 50	572	2	0.34	182	24.20
	Total of First 50 Weeks	752	182	24.20	182	24.20
Year 2	Total after 50 Weeks	572	172	22.87	354	47.07
Final Values	Total	752	354	47.07	354	47.07

In Figure 4, the central diagonal presents the density plots of the Weekly Interactions variable for weeks 1, 10, 20, 30, and 40. The two groups of students (dropout and success) initially presented similar behavior at the beginning of the course (weeks 1 and 10), and gradually started to differ after week 20 when the number of weekly interactions of the successful students was slightly higher. The scatterplots help to better visualize the behavior of the interactions and their comparison between the weeks. The scatter plots demonstrate that there is no direct positive correlation between weeks. Students who were successful in the course tended to have more interactions, similar to that observed in Figure 2.

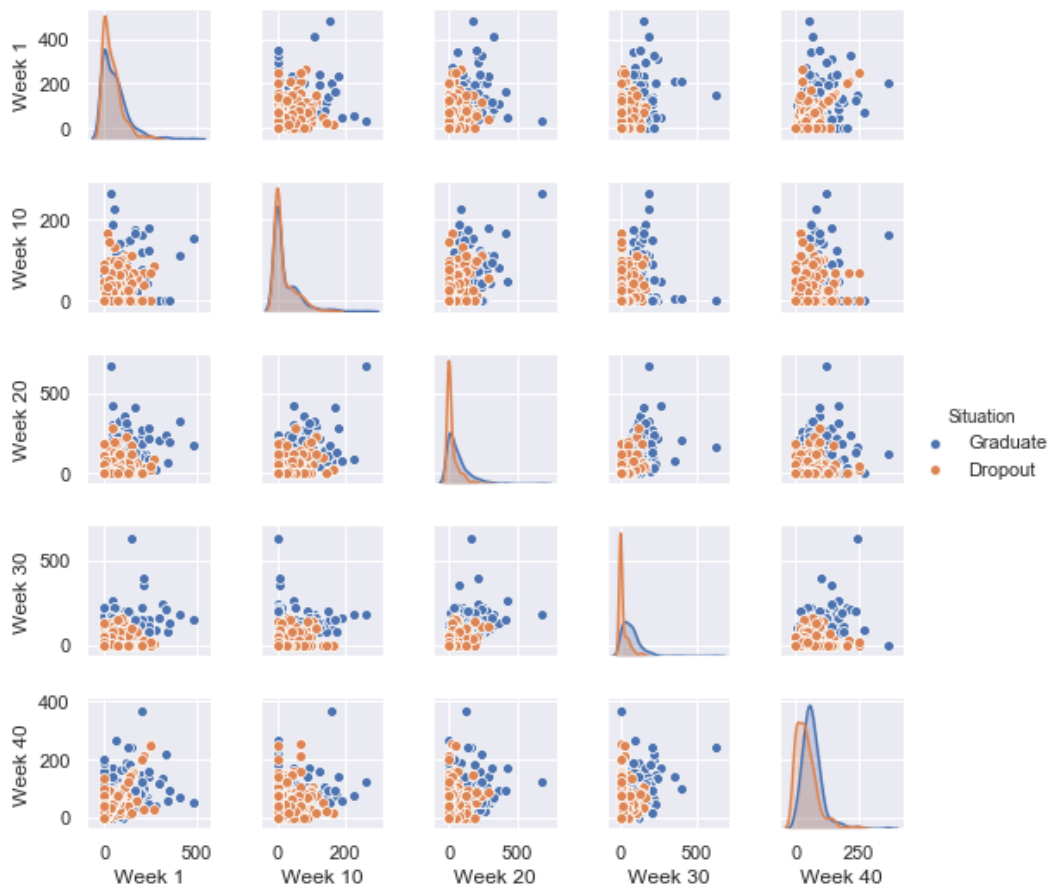


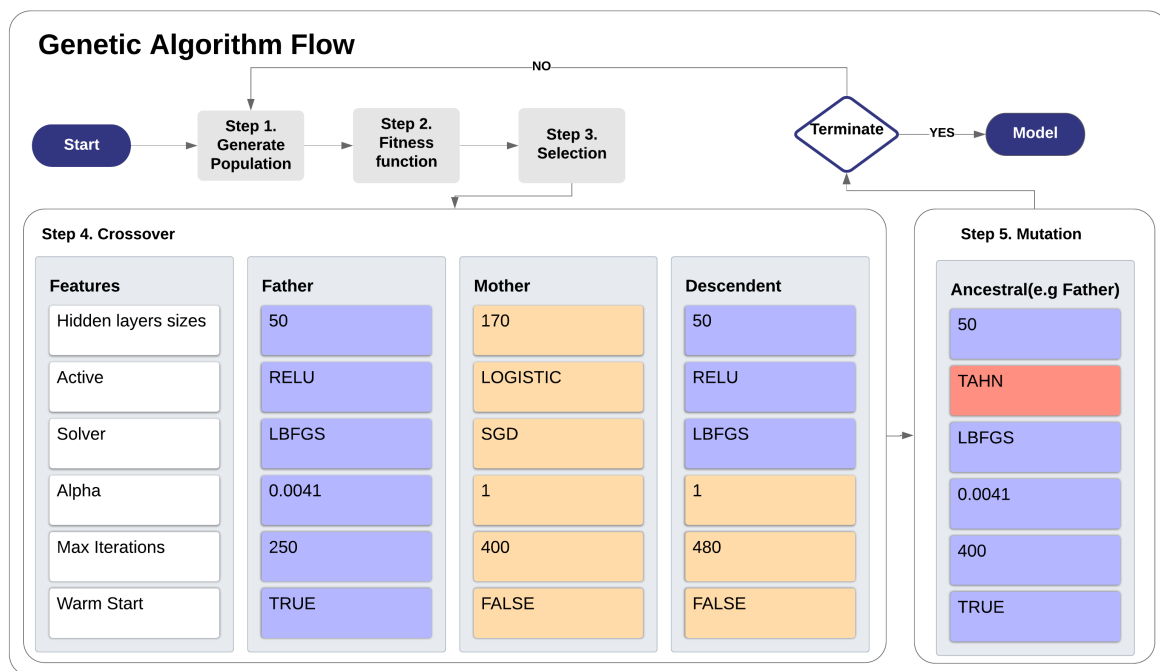
Figure 4. Density and scatter plots of weekly interactions.

### 3.2. Fine Tuning with Proposed Genetic Algorithm

In the GA, the solution set is defined by a space where a search for an optimal solution occurs, which may not be the global best solution [46]. This factor is directly dependent on the problem, the time that can be spent searching, the expected result, and the input dataset, among others. These should be considered when the algorithm is designed [47]. In this work, a time-limited search approach is proposed, so the algorithm creates a number  $N$  of generations, where  $N$  is predefined at the time of configuration. In the end, the algorithm returns a solution with the setting that produced the best performance according to the predefined metric [48]. In this case, a learning machine model together with its hyperparameters were optimized for the prediction of students at risk in technical distance courses. As previously mentioned, this solution can be global or local. The steps of this process are presented in Figure 5.

The proposed approach is executed according to the general steps of classical GA solutions, which are: (1) Generate population, (2) fitness function, (3) selection, (4) crossover, and (5) mutation. For the context of our solution, the following definitions are provided:

- (a) Epoch: One complete cycle execution of the GA (from Steps 1 to 5). The proposed approach works with 50 epochs;
- (b) Individual (or candidate): A machine learning algorithm/classifier (DT, RF, MLP, LG, and ADA) together with its hyperparameters;
- (c) Chromosome: A vector of hyperparameters for a given individual (machine learning algorithm). As different machine learning algorithms have different hyperparameters, the chromosomes in our study have different sizes and meaning according to the machine learning algorithm to which they are referring.



**Figure 5.** Genetic algorithm flow: An example of crossover and mutation with a multilayer perceptrons (MLP) chromosome.

Here, we outline each step of the process in the context of our proposed approach:

- Step 1 (generate population): The GA generates 100 individuals (candidates) for each machine learning algorithm (DT, RF, MLP, LG, and ADA) with hyperparameters (chromosomes) randomly defined considering the available list of options. The classifiers are trained and tested using 10-fold cross-validation and their performances are measured by using the area under the receiver operating characteristic curve metric (AUROC) [49] and as conducted by Gašević et al. [50].
- Step 2 (fitness function): The performance obtained by each of the 100 individuals of each machine learning algorithm are then compared by the fitness function.
- Step 3 (selection): The 25 individuals with the highest AUC for each machine learning algorithm are selected for the next step.
- Step 4 (crossover): The crossover is conducted using the concept based on the genetic inheritance of sexual reproductions, where each descendant receives a part of the genetic code (chromosome) of the father and part of the mother, as exemplified in Figure 5. Thus, the configurations of the fittest individuals of the last step are combined, one being the father and the other the mother. In the implemented algorithm, the individuals who will assign part of their genetic code to form a new member are chosen randomly from among the 25 best placed of that classifier in the last generation. This step results in 25 new individuals for each machine learning algorithm.
- Step 5 (mutation): This step randomly alters the chromosome (hyperparameter) of the 25 best individuals. In other words, a certain characteristic of an individual selected in the previous step receives a randomly generated configuration. As shown in Figure 5, an individual of the MLP type with hyperparameter “Active” set to “RELU” was changed to “TAHN”. The mutation is set to change only one hyperparameter of the chromosome.

After Step 5, if the GA did not run the predefined number of epochs (50 for our experiment), a new population is generated in Step 1. The last important factor in generating a new population is randomness. For each generation, 25 new individuals are randomly generated again, even though they may have already been generated in earlier epochs. This seeks to ensure population diversity by narrowing the hypothesis that the solution reaches a local maximum and has no opportunity to evolve

to the global maximum. The quantitative formation of the population from the second epoch onwards for each machine learning algorithm is:

- 25 individuals selected from the previous generation from the fitness function (Steps 2 and 3);
- 25 individuals formed by crossover (Step 4);
- 25 individuals formed by mutations (Step 5); and
- 25 new individuals randomly generated (Step 1).

The process is repeated for 50 epochs. In the end, for each of the five machine learning algorithms, the individual with the highest aptitude (highest AUC) is selected. With the selection of the fittest for each machine learning algorithm, the five remaining individuals compete against themselves, and the one with the best AUC is selected.

### 3.3. Experiments

This section outlines the experiments with three different approaches to predict students at risk of dropout in the database described earlier. The first is the proposed genetic algorithm, the second is a grid search method called GridSearchCV, implemented using the Scikit-learn package. The third and last is the use of classifiers with their default hyperparameters. The machine learning techniques implemented in this study used the Python programming language with the Scikit-learn, Pandas, and Numpy libraries.

GridsearchCV allowed the testing of different combinations of hyperparameters for classifiers, facilitating choosing the best one. The hyperparameters needed to be explicitly declared and all possible combinations tested. All available combinations in Table 3 were checked with the same algorithms defined in the GA (DT, RF, MLP, LG, and ADA). For each week of the course, we selected the classifier together with its hyperparameters that achieved the best performance for the given week.

The same machine learning algorithms with their default hyperparameters were also implemented for comparison with the GA and GridsearchCV approaches. All experiments were performed with 10-fold cross-validation, and the number of combinations was approximately 5000 individuals created by the GA. Appendix A shows the quantities tested in each of the classifiers in the Evaluations column.

An essential task in machine learning is choosing the performance appraisal metric. For this work, we decided to use the area under the ROC curve, also known as AUC and AUROC. AUC is calculated from the size of the area under the plotted curve where the y-axis is represented by true positive rate (TPR) or sensitivity (Equation (1)), and the x-axis is true negative rate (TNR) or specificity (Equation (2)):

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (1)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (2)$$

According to Gašević et al. [50], the AUC may be interpreted as follows:

- $AUC \leq 0.50$ : Bad discrimination;
- $0.50 < AUC \leq 0.70$ : Acceptable discrimination;
- $0.70 < AUC \leq 0.90$ : Excellent discrimination; and
- $AUC > 0.90$ : Outstanding discrimination.

## 4. Results and Discussion

This section presents the results obtained by the models generated by each of the selected algorithms compared with the application of the GA. Table 5 presents the AUC results for each tested machine learning algorithm without hyperparameter optimization and for the grid search (GRID) and GA approaches.

Table 5. AUC for 50 weeks.

Approach or Machine Learning Algorithm	Hyperparameter Optimization	AUC Mean	AUC Median	AUC Standard Deviation
GA	Yes	0.8454	0.8498	0,0637
GRID		0.7939	0.8288	0.1056
ADA		0.7509	0.8062	0.1342
DT	No	0.6771	0.7065	0.1008
LG		0.6943	0.7198	0.1110
MLP		0.7353	0.7946	0.1277
RF		0.7752	0.8243	0.1150

As can be seen from Table 5, the best AUC results were produced by the GA approach with a mean of 0.8454 and median of 0.8498. GA also produced the lowest AUC standard deviation (0.0637) among all tested approaches. Figure 6 helps visualize the performance of the models for the 50 weeks of the course.

To confirm the research hypothesis in this work (RQ1 and RQ2), two tests of statistical significance were applied. The objective of the tests was to verify if there was a significant difference in the treatments applied and, if so, which method was the most accurate. The central idea involved in the process of statistical significance is to test whether one treatment, in this study GA, presents a significant result concerning the others [51].

The results had a normal distribution, so analysis of variance (ANOVA) was chosen to verify the existence of a significant difference, and Tukey’s test to determine in which treatment it occurred. For this, the *p*-value was set to 0.05; thus, values lower than this indicated that the treatment was significant and higher than not significant. In ANOVA, the *p*-value was 0.0006865, which reflects the existence of significant differences between the approaches. In Tukey’s test, the results produced a *p*-value of 0.0475 for the GridSearch and 0.0003 for standard RF, indicating a statistically significant difference between the performance. Thus, statistically, the results obtained by GA were superior to the other treatments.

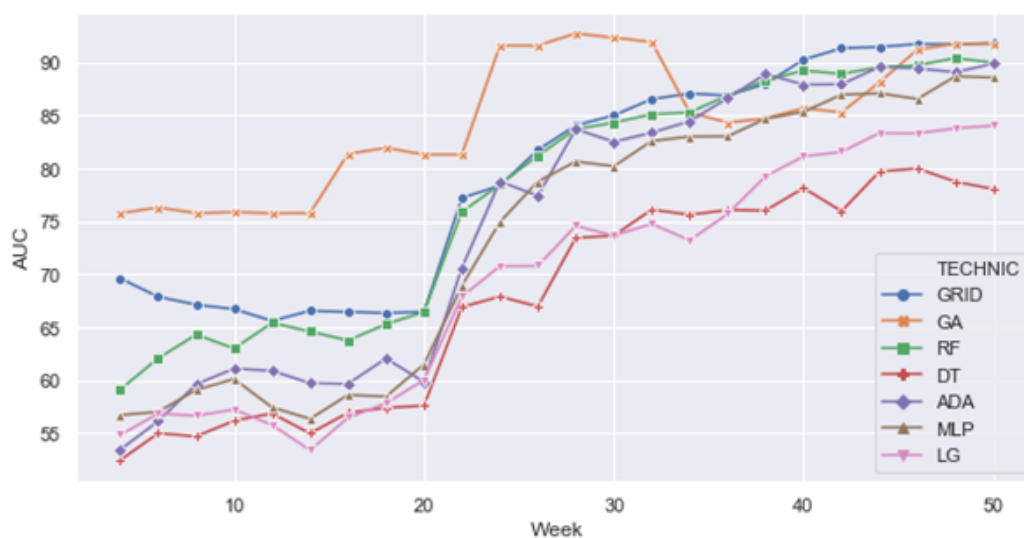


Figure 6. AUC results for each tested technique during the 50 weeks of the course.

The results obtained from the three approaches are presented in Figure 6. The GA achieved excellent discrimination (AUC > 0.7) as early as week 4. This held until week 24, where the GA

provides outstanding discrimination ( $AUC > 0.9$ ). The other approaches still yielded acceptable discrimination results ( $AUC < 0.7$ ) until week 22. However, from week 30, the performance of the GA considerably decreased, with the other approaches progressing. One of the factors determining this drop in GA performance was the increase in the number of input attributes. In this situation, the GA tends to find a local solution quickly and converge on it. However, this solution is probably a plateau and the GA getting stuck. This is a problem specific to genetic algorithms that does not occur in the other approaches tested. In the proposed algorithm, the reinsertion step tries to soften this, but as verified from weeks 32 to 42, the GA is still susceptible to this failure. However, the GA was considerably better in early prediction and with limited data. This demonstrates that the refinement of the GA is essential for tuning hyperparameters.

Table 6 presents the best configuration obtained by the GA approach for week 25 of the course (individual 37, fourth epoch), with an MLP with an AUC of 91.54, in comparison with the configuration for the same algorithm without hyperparameter optimization. From the first weeks of the courses, satisfactory results were already produced in the prediction of students at risk of dropout.

In general, the results of the models generated by GA per the AUC were satisfactory, allowing the prediction of at-risk students in the early stages of the courses. Data were naturally balanced, with similar percentages of dropout and success students. The models developed here produced similar or better results in comparison to some of the works in the literature that focused on the early prediction of dropout students.

**Table 6.** Comparison between configurations of models for week 25.

Hyperparameter Optimization	Hidden Layer Sizes	Activation	Solver	Alpha	max Iter	Warm Start	AUC
yes	30	logistic	sgd	0.2855486101	353	False	0.9154
no	100	relu	adam	0.0001	200	False	0.849

According to [31,52], one of the main factors involved in the acceptance of learning analytics by teachers and students when using prediction models is the correctness rates involved in the process. The GA proposed in this work was able to increase these rates compared to the results obtained in previous works [23,24]. However, direct comparison with these experiments is somewhat complicated, as they used the true positive (TP) and true negative (TN) of the models as metrics, and we used AUCROC.

In these previous experiments, the results obtained in scenarios similar to this experiment showed rates of TP and TN varying between 58 and 82 in the first 25 weeks of the course. However, with the approach proposed in this work, it was possible to reach an initial AUCROC above 0.75, which increased over the first 25 weeks until reaching values above 0.90.

The comparison with prominent works of predictors of educational environments is necessary to situate the results obtained. Some limitations for comparison include the various techniques used to measure the results, such as accuracy, TP, TN, AUC, and AUCROC, among others [32] Cechinel et al. [33] Liz-Domínguez et al. [30]. Still, a significant part of the works on LA are characterized by the exploration of data from disciplines of a specific course or semester, whereas the work presented in this paper is characterized by the use of data from a course of two years in duration [32]. However, even when we compare the results obtained with the related works, the rates are satisfactory. Previous studies Lykourantzou et al. [5], Zohair [37], Whitehill et al. [38] reported rates of 85% and Jayaprakash et al. [26] reported 94% overall accuracy, but only 66.7% dropout prediction.

The results obtained in the optimization with the proposed GA are close to those of the literature Minaei-Bidgoli and Punch [43], which obtained an optimization of 12%. The proposed GA was able to reach values above 10% in the experiments until the 20th week compared to the algorithms in their standard configuration. When compared to the other optimization method, Gridsearch, in that same period, GA obtained values always above 6%, sometimes reaching 15%.

The method followed here is the result of an incremental process of a series of experiments previously performed [23,24]. As such, when comparing the results achieved in this work with the results from previous actions, the hyperparameters generated by the GA allows the generation of more robust models and higher performance. This is also demonstrated in comparison to the other methods tested in this article. Thus, the methodology used both for the development of the GA and for the generation of input data from genetic algorithms demonstrated that it could be used for early prediction of students at risk of dropout. Concerning data modeling, although the use of interaction count is not unprecedented, the methodology used in this study has several attributes that produced the results.

## 5. Final Remarks

This paper presented the results of an approach for the early prediction of students at risk of dropout using the counting of their interactions inside the VLE. This approach uses genetic algorithms for the hyperparameter of classifiers. The methodology of generating a prediction model every two weeks allows every student to be followed throughout the course. This is an approach that differs from the traditional methods [6] that define models that seek to predict dropout using all available data at the end of the course. This difference and, consequently, the results obtained with smaller amounts of data contribute to the early prediction of the risk of dropout.

The proposed approach is based on the premise of allowing greater generalization when replicating the methodology in other courses and platforms, since it only uses the count of interactions within the VLE without distinguishing the types of actions performed and without using information from different data sources (demographic data, questionnaires, curriculum, etc.), the availability of which may vary between e-learning platforms. The results can be considered satisfactory since they allow the identification of students at risk of dropout with reasonable performance rates even before the end of the first semester of the course.

The prediction of academic issues, such as performance and dropout, is concentrated at the university level, with about 70% of the research destined for this purpose [10]. This trend is repeated in Latin America, with few applications considering the context of education at the secondary and technical levels [33]. While not unprecedented, the application of prediction techniques in other contexts, such as technical high school e-learning, is also relevant [10].

**RQ1.** Does the approach for hyperparameter optimization with a GA outperform traditional techniques?

The proposed GA must be evaluated to emphasize that testing different combinations of hyperparameters within the same algorithm is a complicated and time-consuming task that may require a large amount of processing time. However, the accuracy of prediction models is directly linked to the quality of hyperparameter optimization. Thus, the more adjusted they are, the more accurate the rates of the models tend to be. The alternatives to applying exhaustive search methods, such as grid-search, are computationally expensive when searching in large spaces [53]. Thus, the refinement obtained by GA with its mutation and crossover stages produces better results for model generation, surpassing the traditional techniques and grid-search. Compared to standard algorithms, the performance of the proposed method is clearly superior.

**RQ2.** Does the resulting predictive models generated by the use of the GA approach for hyperparameter optimization perform better than models with default hyperparameters?

In comparison with the classifications using the default hyperparameters, the GA produced significantly better results. In the first 20 weeks of the course, the difference between the two methods varies from 10% and 15%. Tukey's test demonstrated that the overall values obtained are significantly different. However, all techniques have advantages and limitations. The drawback of the GA is the lack of assurance that the solution is global; the positive aspect is the number of resultant

hyperparameters accepted without significantly altering the processing cost and the final results. In grid-search, the computational cost is the biggest issue, as previously reported; however, it delivers the best possible combination of hyperparameters. Concerning the standard classifiers, we highlight the cost–benefit factor as the method produces satisfactory results in short processing time, which, depending on the project, can be an essential point.

The main limitation of the proposed methodology presents is that for each course analyzed, the calendar must be studied to identify periods without classes, such as holidays. This causes extra work, which does not occur when socio-demographic data are used. Another limitation concerns generalization; although the methodology may be generalized, the models are unlikely to be suitable for courses that do not follow the same timetable as ETec. Models that seek long-term predictions are more susceptible to failures due to external situations, such as economic and epidemiological crises.

An important point to note is that the GA possibly presents slightly different results for each execution. Thus, it may be interesting to run the GA multiple times (e.g., 10). Analysis of other metrics, such as overall accuracy and true positive (TP) and true negative (TN), may provide different perspectives. The application of other hyperparameter search methods, such as random search, and algorithms, such as XGBOOST, can still be explored. These questions will possibly be studied in the future stages of this project, as well as hybrid choice methods such as the vote theory, for final classification selection.

The results obtained in this work enable the development of an early warning system using the proposed approach. Currently, the development of this system is occurring in the form of a plugin integrated with Moodle. Another future work toward improving the results is the application of survival analysis to increase student retention and consequently reduce dropout.

**Author Contributions:** Author Contributions: E.M.Q.: Experimental data analysis, algorithms development, data pre-processing, experiments conduction, results description, and manuscript writing; J.L.L.: Course lecturer, conceived and designed, and methodology definition; K.K.: Algorithms development and writing review; M.A.: manuscript writing and algorithms development; R.M.A.: Algorithms development and data pre-processing; R.V.: writing—review and editing; R.M.: writing—review and editing. C.C.: methodology definition, algorithm developed, experiments setup, manuscript writing, and writing—review and editing. The final manuscript was written and approved by all authors. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by CNPq (Brazilian National Council for Scientific and Technological Development) [Edital Universal, proc.404369/2016-2][DT-2 Productivity in Technological Development and Innovative Extension scholarship, proc.315445/2018-1]. R.V. and R.M. were funded by Corporación de Fomento de la Producción (CORFO) 14ENI2-26905 “Nueva Ingeniería para el 2030”—Pontificia Universidad Católica de Valparaíso, Chile.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

ADA	ADABOOST
ANOVA	Analysis of Variance
AUC	Area Under the Curve
AUROC	Area Under the Receiver Operating Characteristic Curve
DT	Decision Tree
EDA	Exploratory Data Analysis
EDM	Educational Data Mining
GA	Genetic Algorithm
GBGP	Grammar-Based Genetic Programming
GRID	Grid SearchCV
IBK	Instance-Based Lazy Learning
ICRM	Interpretable Classification Rule Mining Algorithm
IFSul	Instituto Federal Sul Rio-grandense



- INN I-nearest neighbor
- kNN k-Nearest Neighbor
- LA Learning Analytics
- LG Logistic Regression
- LMS Learning Management Systems
- ML Machine Learning
- MAE Mean Absolute Error
- MLP Multilayer Perceptron
- NDS Number of Dropout Students
- PCA Principal Component Analysis
- RQ Research Question
- RF Random Forest
- SAT Scholastic Aptitude Test
- SMOTE Synthetic Minority Over-Sampling Technique
- SVM Support Vector Machine
- TNR True Negative Rate
- TPR True Positive Rate
- VLE Virtual Learning Environment

### Appendix A

**Table A1.** Classifiers, Hyperparameters and Number of Evaluations.

Alg.	Hyperparameters	Possibi-Lities	Number of Ind.	Grid	Eval.
DT	criterion: [gini, entropy], max_depth: range (0, 32), min_samples_split: range (1, 15), min_samples_leaf: range (1, 20)	19.200	5.100	criterion: [gini,entropy], max_depth: [0, 1, 2, 3, 5, 7, 10, 12, 15, 17, 20, 23, 25, 30], min_samples_split: [0, 1, 2, 3, 5, 7, 10, 12, 15] min_samples_leaf: [0, 1, 2, 3, 4, 5, 7, 9, 10, 12, 15, 17, 20]	3.726
RF	n_estimators: range (1,200), criterion: [gini, entropy], max_features [1, 2, 3,4], min_samples_split: range (2, 21), min_samples_leaf: range (1, 2), bootstrap: [True, False]	128.000	5.100	n_estimators: [1, 10, 20, 30, 40, 50, 70, 100, 120, 130, 150, 170, 190, 200], criterion: [gini, entropy], max_features [1, 2, 3, 4], min_samples_split: [2, 3, 4, 5, 7, 9, 10, 12, 15, 17, 20], min_samples_leaf: [1, 2], bootstrap: [True, False]	4.928
ADA	algorithm: [SAMME, SAMME.R], n_estimators: range (1, 200), random_state: range (None, 50), learning_rate: range (1e-2,1)	2 KK	5.100	algorithm: [SAMME, SAMME.R], n_estimators: [1, 10, 20, 30, 40, 50, 70, 100, 120, 130, 150, 170, 190, 200], random_state: [None, 1, 5, 10, 15, 20, 25, 30, 40, 50 ], learning_rate: [1e-2, 5e-2, 7e-2, 1e-1, 3e-1, 5e-1, 7e-1, 1]	2.240
MLP	hidden_layer_sizes: range (1,200), activation: [identity, logistic, tanh, relu], solver: [lbfgs, sgd, adam], max_iter: range (50, 200), alpha: range (1e-4, 1e-1), warm_start: [True, False]	720 KK	5.100	hidden_layer_sizes: [(50,50,50), (50,100,50), (100,),(50,),(10,),( 1),(5)], activation: [identity, logistic, tanh, relu], solver: [lbfgs, sgd, adam], max_iter: [1, 2, 5, 10, 30, 50], alpha: [1e-4, 1e-3, 1e-2, 5e-2, 1e-1], warm_start: [True, False]	5.040
RL	penalty: [l1, l2, elasticnet], C: [1e-4, 1e-3, 1e-2, 1e-1, 5e-1, 1, 5, 10, 15, 20, 25], dual: [True, False], solver: [newton-cg, lbfgs, lbfgs, sag, saga], multi_class: [ovr, auto], max_iter: range (50,200)	99.000	5.100	penalty: [l1, l2, elasticnet], C: [1e-4, 1e-1, 5e-1, 1, 5, 15, 25], dual: [True, False], solver: [newton-cg, lbfgs, lbfgs, sag, saga], multi_class: [ovr, auto], max_iter: [1, 10, 20, 30, 40, 50, 70, 100, 120, 130, 150, 170, 190, 200]	5.800

## References

1. Chatti, M.A.; Dyckhoff, A.L.; Schroeder, U.; Thüs, H. A reference model for learning analytics. *Int. J. Technol. Enhanc. Learn.* **2013**, *4*, 318–331. [[CrossRef](#)]
2. Siemens, G. Learning analytics: The emergence of a discipline. *Am. Behav. Sci.* **2013**, *57*, 1380–1400. [[CrossRef](#)]
3. Sheehan, M.; Park, Y. pGPA: A personalized grade prediction tool to aid student success. In Proceedings of the Sixth ACM Conference on Recommender Systems, Dublin City, Ireland, 9–13 September 2012; pp. 309–310.
4. Manhães, L.M.B.; Cruz, S.d.; Costa, R.J.M.; Zavaleta, J.; Zimbrão, G. Previsão de Estudantes com Risco de Evasão Utilizando Técnicas de Mineração de Dados. In Proceedings of the Anais do XXII SBIE-XVII WIE, Aracaju, Brazil, 21–25 November 2011.
5. Lykourantzou, I.; Giannoukos, I.; Nikolopoulos, V.; Mpardis, G.; Loumos, V. Dropout prediction in e-learning courses through the combination of machine learning techniques. *Comput. Educ.* **2009**, *53*, 950–965. [[CrossRef](#)]
6. Márquez-Vera, C.; Cano, A.; Romero, C.; Noaman, A.Y.M.; Mousa Fardoun, H.; Ventura, S. Early dropout prediction using data mining: A case study with high school students. *Expert Syst.* **2016**, *33*, 107–124. [[CrossRef](#)]
7. OECD. *Benchmarking Higher Education System Performance*; OECD: Paris, France, 2019; p. 644. [[CrossRef](#)]
8. Yukselturk, E. Predicting Dropout Student: An Application of Data Mining Methods in an Online Education Program. *Comput. Educ.* **2014**, *17*, 118–133. [[CrossRef](#)]
9. Li, Q.; Baker, R.; Warschauer, M. Using clickstream data to measure, understand, and support self-regulated learning in online courses. *Internet High. Educ.* **2020**, 100727. [[CrossRef](#)]
10. Rastrollo-Guerrero, J.L.; Gómez-Pulido, J.A.; Durán-Domínguez, A. Analyzing and Predicting Students' Performance by Means of Machine Learning: A Review. *Appl. Sci.* **2020**, *10*, 1042. [[CrossRef](#)]
11. Vossensteyn, J.J.; Kottmann, A.; Jongbloed, B.W.; Kaiser, F.; Cremonini, L.; Stensaker, B.; Hovdhaugen, E.; Wollscheid, S. Dropout and Completion in Higher Education in Europe: Main Report. In *European Commission; Center for Higher Education Policy Studies and Nordic Institute for Studies in Innovation Research and Education*: Enschede, The Netherlands, 2015; doi:10.2766/826962. 2015. [[CrossRef](#)]
12. Gregori, E.B.; Zhang, J.; Galván-Fernández, C.; Fernández-Navarro, F.d.A. Learner support in MOOCs: Identifying variables linked to completion. *Comput. Educ.* **2018**, *122*, 153–168. [[CrossRef](#)]
13. Censo, E. BR 2018-Relatório Analítico da Aprendizagem a Distância no Brasil. *Acesso Em* **2018**, 16.
14. Dickson, W.P. Toward a deeper understanding of student performance in virtual high school courses: Using quantitative analyses and data visualization to inform decision making. In *A Synthesis of New Research in K–12 Online Learning*; Michigan Virtual University: Lansing, MI, USA, 2005; pp. 21–23.
15. Murray, M.; Pérez, J.; Geist, D.; Hedrick, A. Student interaction with content in online and hybrid courses: Leading horses to the proverbial water. In Proceedings of the Informing Science and Information Technology Education Conference, Santa Rosa, CA, USA, 30 June–6 July 2013; Informing Science Institute: Santa Rosa, CA, USA, 2013; pp. 99–115.
16. Leitner, P.; Ebner, M.; Ebner, M. Learning Analytics Challenges to Overcome in Higher Education Institutions. In *Utilizing Learning Analytics to Support Study Success*; Springer: Berlin, Germany, 2019; pp. 91–104.
17. Gursoy, M.E.; Inan, A.; Nergiz, M.E.; Saygin, Y. Privacy-preserving learning analytics: Challenges and techniques. *IEEE Trans. Learn. Technol.* **2016**, *10*, 68–81. [[CrossRef](#)]
18. Drachsler, H.; Greller, W. Privacy and analytics: It's a DELICATE issue a checklist for trusted learning analytics. In Proceedings of the Sixth International Conference on Learning Analytics & Knowledge, Edinburgh, Scotland, 25–29 April 2016; pp. 89–98.
19. Baker, R.S.; Inventado, P.S. Educational data mining and learning analytics. In *Learning Analytics*; Springer: Berlin, Germany, 2014; pp. 61–75.
20. Olivares, R.; Munoz, R.; Soto, R.; Crawford, B.; Cárdenas, D.; Ponce, A.; Taramasco, C. An Optimized Brain-Based Algorithm for Classifying Parkinson's Disease. *Appl. Sci.* **2020**, *10*, 1827. [[CrossRef](#)]
21. Bergstra, J.S.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Granada, Spain, 2011; pp. 2546–2554, ISBN 9781618395993.

22. Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.* **2017**, *18*, 6765–6816.
23. Queiroga, E.; Cechinel, C.; Araújo, R. Predição de estudantes com risco de evasão em cursos técnicos a distância. In Proceedings of the Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE), Recife, Brazil, 30 October–2 November 2017; p. 1547.
24. Queiroga, E.; Cechinel, C.; Araújo, R.; da Costa Bretanha, G. Generating models to predict at-risk students in technical e-learning courses. In Proceedings of the IEEE Latin American Conference on Learning Objects and Technology (LACLO), San Carlos, CA, USA, 3–7 October 2016; pp. 1–8.
25. Detoni, D.; Cechinel, C.; Matsumura Araújo, R. Modelagem e Predição de Reprovação de Acadêmicos de Cursos de Educação a Distância a partir da Contagem de Interações. *Revista Brasileira de Informática na Educação* **2015**, *23*, 1.
26. Jayaprakash, S.M.; Moody, E.W.; Lauria, E.J.M.; Regan, J.R.; Baron, J.D. Early Alert of Academically At-Risk Students: An Open Source Analytics Initiative. *J. Learn. Anal.* **2014**, *1*, 6–47. [[CrossRef](#)]
27. Márquez-Vera, C.; Cano, A.; Romero, C.; Ventura, S. Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. *Appl. Intell.* **2013**, *38*, 315–330. [[CrossRef](#)]
28. Xing, W.; Guo, R.; Petakovic, E.; Goggins, S. Participation-based student final performance prediction model through interpretable Genetic Programming: Integrating learning analytics, educational data mining and theory. *Comput. Hum. Behav.* **2015**, *47*, 168–181. [[CrossRef](#)]
29. Munoz, R.; Olivares, R.; Taramasco, C.; Villarroel, R.; Soto, R.; Barcelos, T.S.; Merino, E.; Alonso-Sánchez, M.F. Using black hole algorithm to improve eeg-based emotion recognition. *Comput. Intell. Neurosci.* **2018**, *2018*, 22. [[CrossRef](#)]
30. Liz-Domínguez, M.; Caeiro-Rodríguez, M.; Llamas-Nistal, M.; Mikic-Fonte, F.A. Systematic Literature Review of Predictive Analysis Tools in Higher Education. *Appl. Sci.* **2019**, *9*, 5569. [[CrossRef](#)]
31. Herodotou, C.; Rienties, B.; Verdin, B.; Boroowa, A. Predictive learning analytics ‘at scale’: Towards guidelines to successful implementation in Higher Education based on the case of the Open University UK. *J. Learn. Anal.* **2019**. [[CrossRef](#)]
32. Hilliger, I.; Ortiz-Rojas, M.; Pesántez-Cabrera, P.; Scheihing, E.; Tsai, Y.S.; Muñoz-Merino, P.J.; Broos, T.; Whitelock-Wainwright, A.; Pérez-Sanagustín, M. Identifying needs for learning analytics adoption in Latin American universities: A mixed-methods approach. *Internet High. Educ.* **2020**, *45*, 100726. [[CrossRef](#)]
33. Cechinel, C.; Ochoa, X.; Lemos dos Santos, H.; Carvalho Nunes, J.B.; Rodés, V.; Marques Queiroga, E. Mapping Learning Analytics initiatives in Latin America. *Br. J. Educ. Technol.* **2020**, doi:10.1111/bjet.12941. [[CrossRef](#)]
34. González, P. Factores que favorecen las presencia docente en entornos virtuales de aprendizaje. *Tendencias Pedagógicas* **2017**, *29*, 43–58.
35. de Pablo González, G. *La Importancia de la Presencia Docente en Entornos Virtuales de Aprendizaje*; Universidad Autónoma de Madrid: Madrid, Spain, 2016.
36. Herodotou, C.; Rienties, B.; Boroowa, A.; Zdrahal, Z.; Hlosta, M.; Naydenova, G. Implementing predictive learning analytics on a large scale: The teacher’s perspective. In Proceedings of the Seventh International Learning Analytics & Knowledge Conference, Vancouver, BC, Canada, 13–17 March 2017; pp. 267–271.
37. Zohair, L.M.A. Prediction of Student’s performance by modelling small dataset size. *Int. J. Educ. Technol. High. Educ.* **2019**, *16*, 27. [[CrossRef](#)]
38. Whitehill, J.; Mohan, K.; Seaton, D.; Rosen, Y.; Tingley, D. Delving deeper into MOOC student dropout prediction. *arXiv* **2017**, arXiv:1702.06404.
39. Macarini, B.; Antonio, L.; Cechinel, C.; Batista Machado, M.F.; Faria Culmant Ramos, V.; Munoz, R. Predicting Students Success in Blended Learning—Evaluating Different Interactions Inside Learning Management Systems. *Appl. Sci.* **2019**, *9*, 5523. [[CrossRef](#)]
40. Queiroga, E.; Cechinel, C.; Araújo, R. Um Estudo do Uso de Contagem de Interações Semanais para Predição Precoce de Evasão em Educação a Distância. In Proceedings of the Anais dos Workshops do Congresso Brasileiro de Informática na Educação, Maceio, Brazil, 26–30 October 2015; p. 1074.
41. Swan, K. Learning effectiveness online: What the research tells us. *Elem. Qual. Online Educ. Pract. Dir.* **2003**, *4*, 13–47.

42. Halawa, S.; Greene, D.; Mitchell, J. Dropout Prediction in MOOCs using Learner Activity Features. *Eur. MOOC Summit EMOOCs* **2014**, *37*, 1–10.
43. Minaei-Bidgoli, B.; Punch, W.F. Using genetic algorithms for data mining optimization in an educational web-based system. In Proceedings of the Genetic and eEvolutionary Computation Conference, Chicago, IL, USA, 12–16 July 2003; Springer: Berlin, Germany, 2003; pp. 2252–2263.
44. Silva Filho, R.L.L.; Motejunas, P.R.; Hipólito, O.; Lobo, M.B.d.C.M. A evasão no ensino superior brasileiro. *Cadernos de Pesquisa* **2007**, *37*, 641–659. [[CrossRef](#)]
45. Resende, M.L.d.A. *Evasão Escolar No Primeiro Ano Do Ensino médio Integrado Do Ifsuldeminas-Campus Machado*; Encontro Anual da ANPOCS: Caxambu, Brazil, 2012.
46. Fonseca, C.M.; Fleming, P.J. Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization. In Proceedings of the ICGA, San Mateo, CA, USA, 17–22 June 1993; pp. 416–423.
47. Hartmann, S. A competitive genetic algorithm for resource-constrained project scheduling. *Nav. Res. Logist. (NRL)* **1998**, *45*, 733–750. [[CrossRef](#)]
48. Sebastiani, F. Machine learning in automated text categorization. *ACM Comput. Surv. (CSUR)* **2002**, *34*, 1–47. [[CrossRef](#)]
49. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [[CrossRef](#)]
50. Gašević, D.; Dawson, S.; Rogers, T.; Gasevic, D. Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success. *Internet High. Educ.* **2016**, *28*, 68–84. [[CrossRef](#)]
51. Bruce, P.; Bruce, A. *Practical Statistics for Data Scientists: 50 Essential Concepts*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2017.
52. Larrabee Sønderlund, A.; Hughes, E.; Smith, J. The efficacy of learning analytics interventions in higher education: A systematic review. *Br. J. Educ. Technol.* **2019**, *50*, 2594–2618. [[CrossRef](#)]
53. Zöllner, M.A.; Huber, M.F. Survey on automated machine learning. *arXiv* **2019**, arXiv:1904.12054.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).