*data*

MDPI

*Article*

# Predicting Academic Success of College Students Using Machine Learning Techniques

**Jorge Humberto Guanin-Fajardo [1], Javier Guaña-Moya [2,*] and Jorge Casillas [3]**

[1] Facultad de Ciencias de la Ingeniería, Universidad Técnica Estatal de Quevedo, Quevedo 120508, Ecuador; jorgeguanin@uteq.edu.ec
[2] Facultad de Ingeniería, Pontificia Universidad Católica del Ecuador, Quito 170525, Ecuador
[3] Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain; casillas@decsai.ugr.es
* Correspondence: eguana953@puce.edu.ec; Tel.: +593-995000484

**Abstract:** College context and academic performance are important determinants of academic success; using students' prior experience with machine learning techniques to predict academic success before the end of the first year reinforces college self-efficacy. Dropout prediction is related to student retention and has been studied extensively in recent work; however, there is little literature on predicting academic success using educational machine learning. For this reason, CRISP-DM methodology was applied to extract relevant knowledge and features from the data. The dataset examined consists of 6690 records and 21 variables with academic and socioeconomic information. Preprocessing techniques and classification algorithms were analyzed. The area under the curve was used to measure the effectiveness of the algorithm; XGBoost had an AUC = 87.75% and correctly classified eight out of ten cases, while the decision tree improved interpretation with ten rules in seven out of ten cases. Recognizing the gaps in the study and that on-time completion of college consolidates college self-efficacy, creating intervention and support strategies to retain students is a priority for decision makers. Assessing the fairness and discrimination of the algorithms was the main limitation of this work. In the future, we intend to apply the extracted knowledge and learn about its influence of on university management.

**Keywords:** educational data mining; machine learning; educational analysis; higher education; academic success



**Citation:** Guanin-Fajardo, J.H.; Casillas, J.; Guaña-Moya, J. Predicting Academic Success of College Students Using Machine Learning Techniques. *Data* **2024**, *9*, 60. https://doi.org/10.3390/data9040060

Academic Editor: Antonio Sarasa Cabezuelo

Received: 31 January 2024
Revised: 13 April 2024
Accepted: 15 April 2024
Published: 22 April 2024



## 1. Introduction

Higher education has developed a fundamental role due to the versatility and complexity of today's world, which has led to the rapid growth of scientific literature dedicated to predicting academic success or the risk of student dropout [1–7]. Higher education institutions and their traditional role of knowledge dissemination have changed; innovation in new knowledge especially with the irruption of artificial intelligence [8] and the training of qualified professionals make many of them interact in different areas of society. In fact, their missions through teaching, research, and the ability to share and transfer this knowledge constitute central functions of their academic and cultural activity, with the aim of improving the level of knowledge in society. They have the important role of transmitting knowledge, skills, and values to students to create competitive professionals in society. Therefore, channeling students towards academic success is transcendental, as HEIs must continue the work undertaken and further deepen their involvement, significance, and service capacity in relation to the social, cultural, and economic framework [9]. Thus, the prediction of academic success with past information of students who have successfully completed their university studies has become a tool of interest for educational managers since it allows them to strengthen decisions and build improvement alternatives

or educational policies. ICT is one of the most widely used alternatives today, especially machine learning.

Hence, advances in machine learning techniques, along with other areas of study, are precursors to educational data mining. In higher education, the academic success of students is statistically measured by the graduation rate, which is defined as the total number of students graduating among the total number of entering students. In fact, ref. [10] states that it is possible to think about student success more broadly by studying endogenous and exogenous factors in the student environment. Thus, the constant need to be effective in the academic success of students has led to the customization of machine learning, this to achieve specific predictive models that provide useful information.

In the last decade, many studies have focused on investigative works that address the problems of performance, dropout, and academic success in university students. As detailed in [11–14], the authors emphasize that university dropout or failure converges with students from disadvantaged social strata who project university dropout behavior. To sustain university permanence among their findings, the authors are inclined to consider that extra-university activities that guarantee retention should be strengthened. Therefore, early detection has become a tool for solving these problems. Academic history, university context (tangible and intangible resources), and other data were used as the input elements to predict the results [4]. For this purpose, qualitative and quantitative research methods have been used to solve these problems. More recently, multiple studies have been derived that employed data mining or machine learning techniques that, among other things, use algorithms and two well-known techniques to extract useful knowledge from data. The first technique, supervised classification, evaluates the data and predicts the target variable (class). The work of [6,15–17] has shown results related to supervised classification.

Similarly, in [18,19], using another approach based on supervised classification, they used a set of pre-selected algorithms that classify the data by applying the voting technique. Both approaches attempt to predict students' academic success or performance effectively. The second technique, unsupervised classification, is one in which the target variable is unknown and that focuses on finding hidden patterns among the data. In general, association rules are used to discover facts occurring within the data and are composed of two parts: antecedent and consequent; for example, the rule {A, B}⇒{C} means that, when A and B occur, then C occurs. In [20–22], they look for the occurrence of data by focusing on the association rules and evaluating the rules with metrics such as support, confidence, and lift, among others.

In the studies of [23–25], related to machine learning, the convergence of objectives and techniques applied for the data preprocessing stage was observed, both in feature reduction, data transformation, normalization, and instance selection, among others. At the same time, data balancing techniques and "black box" classification algorithms were analyzed. The synergy of the studies lies in the simplification of the predictive models obtained given the high degree of complexity of the extracted knowledge, for which they used decision trees, since this technique simplifies the knowledge by means of the representation of rules of type (X⇒Y). To some extent, the methods applied are part of the KDD process proposed in [26]. However, data asymmetry is a typical problem in any area of study. Duplicity, ambiguity, and missing and overlapping data are frequent, especially in authentic problems. Indeed, in data mining classification techniques, problems are presented as an unequal distribution of examples among classes (target variable), where one or more classes (minority class) are underrepresented compared to the others (majority class) [27]. Commonly, the data balancing method defined by Chawla [28] is used in this type of problem. However, it is intended to fill the existing gap of data balancing with educational data by using different balancing methods for multiclass problems.

The approach of this study is like previous work described in [6,29–31], where similar tasks were performed with predictions in binary and multiclass classes. However, the main difference with our approach focuses on the in-depth analysis of data balancing and feature selection techniques to avoid biases in predictions. Using 53% fewer variables

and improving its accuracy by 10% over the preliminary results with the raw data, we not only built classification models to identify the relevant factors of college students' academic success, but also obtained a general model from the decision tree to obtain a higher readability of the predictive model. In this way, it is intended to provide additional guidance to academic decision makers in decision making. The open license software used for this work was R [32] through a customized library to visualize, preprocess and classify the data. The Python library scikit-learn [33] was used for data balancing.

The core of the work focuses on the study of machine learning techniques that predict academic success. This has allowed us to establish the objective of the work, which is to know in advance the factors that explain the academic success of students at the end of their first year of university. To do this, it has been necessary to pose the research questions since we intend to identify the factors that contribute to the academic success of students during their first year of college. This will allow us to examine the preprocessing techniques, the predictive model, the determinants of academic success and, of course, the visualization techniques to improve its interpretation before and after obtaining the predictive model. In this sense, the following research questions were posed:

- RQ1: Which balancing and feature selection technique is relevant for supervised classification algorithms?
- RQ2: Which predictive model best discriminates students' academic success?
- RQ3: Which factors are determinants of students' academic success?

Most studies on predicting academic success by machine learning have focused solely on finding a predictive model, which is, to some extent, highly effective. In contrast, the work presented, in line with RQ1, seeks the group of features that are most significant for the model and, on the other hand, also seeks a balanced training dataset, using different data balancing techniques and avoiding biases in the prediction. RQ2, on the other hand, aims to find the effective predictive model using different supervised learning algorithms. Finally, RQ3 examines which variables were relevant in the predictive model achieved by the machine learning algorithms to then obtain another model with a better interpretation for the decision maker.

The presented work differs, among other things, by the following contributions: (i) we unveil the effectiveness of educational data mining techniques, to identify academically successful students early enough to act and reduce the failure rate; (ii) the impact of data preprocessing is analyzed; (iii) the important variables underlying the predictive model of better performance are unveiled. Thus, an approach to the presented work is associated with the works of [23,29,34], where the authors have examined the characteristics and impact of the best-performing algorithm. The rest of the paper is organized as follows: in Section 2, a literature review is carried out; in Section 3, the methodology used in this work is explained; in Section 4, the main results obtained by applying machine learning are presented; in Section 5, the discussion is presented; in Section 6, the relevant conclusions, in Section 7, limitations; and finally, in Section 8 future work are described.

## 2. Literature Review

In the cited literature, there are works related to the study of machine learning in higher education and its impact on the prediction of academic performance or success. In prediction, the purpose is to predict the target variable (class) of a dataset. The works cited in Table 1 employ supervised classification algorithms that focus on obtaining the predictive model.

**Table 1.** Summary of papers related to the prediction of academic performance or success of university students.

| Objective | Inst. [1] | Feat. [2] | Class | DPM [3] | Accuracy | Citation | Scope |
|---|---|---|---|---|---|---|---|
| Performance | 6948 | 55 | 2 | Data preprocessing methods | 82% | [35] | Higher Education |
| Performance | 3830 | 27 | 2 | Data transformation, Discretization | 83% | [36] | Higher Education |
| Prediction | 1854 | 4 | 2 | | 75% | [37] | |
| Academic Success Assessment | 731 | 12 | 2 | Extraction Feature, Imbalanced Dataset | 78% | [6] | Higher Education |
| Achievement | 339 | 15 | 3 | Extraction Feature | 69.3% | [23] | Higher Education |
| Performance | 32,593 | 31 | 4 | Extraction Feature, Imbalanced Dataset | 72.73% | [38] | Higher Education |
| Prediction | 9652 | 68 | 2 | Extraction Feature, Imbalanced Dataset | 75.43% | [24] | Higher Education |
| Prediction | 3225 | 57 | 2 | Extraction Feature, Imbalanced Dataset | 79.5% | [28] | Higher Education |
| Prediction | 300 | 18 | 2 | Extraction Feature | 63.33% | [34] | Higher Education |
| Prediction | 1491 | 13 | 2 | Extraction Feature, Imbalanced Dataset | 75.78% | [5] | Higher Education |
| Prediction | 7936 | 29 | 2 | Extraction Feature | 69.3% | [30] | Higher Education |
| Prediction | 4413 | | 2 | | | [18] | Higher Education |
| Prediction | 6690 | 21 | 3 | Selection Feature, Selection Instance, Data imbalanced | 81% | Our proposal | Higher Education |

[1] Number of instances. [2] Number of features. [3] Data preprocessing methods.

Among other works, the use of machine learning techniques to predict the success or failure of university courses or degrees stands out. The use of the recommender system proposed by [35] suggests to computer science students the subjects they can take, in addition to the prediction of success or failure based on the previous experience of other university students. In the work, data preprocessing and example balancing techniques were applied. Then, the preprocessed data were used as input for the classification algorithms to learn and obtain the prediction model from the test data. The results achieved provide guidelines for university administrators to enhance educational quality. In this sense, the early provision of useful information to predict a given event in the student body is valuable. Hence, the study of academic performance is a relevant contribution in higher education. Helal [36], in his work, predicted the academic performance of the student body; the data used in his work were divided into groups, and each subgroup of data was evaluated with different classification algorithms to predict academic performance. Their results suggest that external students and female students performed well in the prediction.

The work of Bertolini [29] set out to examine different classification algorithms to predict final exam grades with reasonable accuracy, considering midterm grades. Similarly, Alyahyan [23] proposed the use of decision trees to predict students' academic performance and generate an early warning when low performance is detected. Different decision tree approaches as well as relevant feature extraction were employed to obtain a simpler model for decision making by academic experts. In line with this, refs. [29,34] also examined high-impact features in the data to fit representative variables with respect to college retention and dropout, to develop interventions to help improve student academic success.

Similarly, in Beaulac [39], the prediction of the academic success of university students has been studied by applying the random forest and decision tree algorithms, the latter being very intuitive for decision making; the authors propose the use of these techniques to know if at the end of the first two semesters the student would achieve the university degree. Their results have indicated that there is a strong relationship between underperforming grades and the likelihood of succeeding in a degree program, although this did not necessarily indicate a causal connection.

Several of the related articles reveal the variety of work linked to improving the educational system. The approach of Guerrero-Higueras [7], which proposes the use of the GIT version control system as an evaluation methodology to observe the frequency and use of the tool to help predict the student's academic success, stands out. The variables studied describe the student's ability with tasks related to the development of the computer science subject. This methodology as introduced differs from the rest given the adaptation of the GIT version control platform and the issues specific to the computer science area.

The literature cited above emphasizes gradualism to achieve features that achieve high accuracy in the algorithms and obtain a simple and readable model. The lack of salient features prevents obtaining an effective prediction model. This is because of the ambiguity or irrelevance of the variables [40]. On the other hand, of significant importance is the reduction of outliers in the data due to duplicate observations or overlapping data [41–43]. It is understood, of course, that all of this leads to the application of each stage suggested in the CRISP-DM [26], methodology that allows obtaining a reliable model at the end. The validity of the model obtained is checked by the performance metrics of the classification algorithms. Based on what has been presented in this section, it was observed in the literature that the work focuses mainly on two fronts: identifying significant attributes to predict student performance, success, or failure in higher education, and finding the best prediction method to improve the accuracy of the predictive model achieved.

## 3. Materials and Methods

### *3.1. Context*

The Institution of Higher Education (IES) is geographically located in the Municipality of Quevedo, Province of Los Ríos, Ecuador. Its coordinates are set at: 1°00′46″ S 79°28′09″ W/−1.012778, −79.469167. According to the policies of the IES and its minimum requirements, each university course is taught in face-to-face mode, and in addition, each academic year of the university course must be passed. In this case, each academic year consists of two academic cycles (semesters). Students must enroll in the university degree program and obtain grades in each subject, with a minimum grade of seven on a scale of zero to ten. As a result of the academic activities performed and their permanence in the university degree, the academic status of the student body is determined (dependent variable/class). Academic statuses are established in three categories. The first is "Passed", when the student has completed and passed all academic courses. The second is "Change", when the student passes courses other than the initial degree. And finally, third is "Dropout", when the student leaves the university completely.

### *3.2. Data Collection*

Data collection was performed using SQL server scripts. The data were extracted from the university's information system database server. The dataset used in this work consisted of two parts: student body and faculty, which were subsequently merged. It should be noted that the criterion for the merger was the classes taught in the first year by the faculty in the teaching process for the university degree. Thus, the first part of the information referring to the students dealt with academic and socioeconomic data, while that relating to the teaching staff referred to degrees obtained, age, and academic experience, among others. Among the diversity of professors in charge of university teaching of first-year students, there were full, associate, and occasional professors, totaling 286 professors selected for this study.

On the other hand, the number of regular students was 6690. Although the number of professors and students does not coincide, it is necessary to clarify that a professor can teach different subjects. The students selected were those who were enrolled and had completed the first year of all university courses. In short, all of the above was framed within a retrospective of six complete academic years of each university degree, that is, ten calendar years. It should also be noted that any identifying reference to both faculty and students was eliminated to obtain an anonymous dataset. Among other things, the

information extracted for this work had the endorsement and permission of the competent authority of the higher education institution detailed in the Institutional Review Board Statement section. The database with the raw data had 21 variables and 6690 records (see Appendix A, Table A1 for a description of the variables used).

So far, one of the main differences in algorithms between machine learning (ML) and traditional statistical methods lies in their purpose, as the former is still focused on the ability to capture complex relationships between features and make predictions as accurate as possible, while the latter, especially linear regression (LR), logistic regression (LOR), generalized mixed models and relevance-based prediction and others, aim at inferring relationships between variables. However, the key difference between traditional statistical approaches and ML is that, in ML, a model learns from examples rather than being programmed with rules. For a given task, examples are provided in the form of inputs (called features or attributes) and outputs (called labels or classes) [44,45].

In this work, we used the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology proposed by [26], which comprises seven phases: understanding the problem, understanding the data, data preparation, modeling, evaluation and implementation; the data preparation or data preprocessing is a stage that gained importance and became a key stage, since its function is related to data preparation. In other words, the objective is to reduce the complexity of the original dataset to obtain a readable predictive model with useful variables. Therefore, the work is based on the best practice for data preprocessing suggested in [46–48]. For this reason, Appendixes B and Cdetail the results of the various methods used for data preprocessing using feature filtering, instance selection, and class balancing. The main advantage of efficient data preprocessing was the transfer of suitable data to classification algorithms for simple and accurate learning. First, the compacted data were cleaned and transformed and then analyzed with visualization techniques that allowed, among other things, the location of trajectories, overlaps and data behavior. Second, the data were stratified into two subsets of data: training and test. Then, the training set was filtered for relevant instances and features to balance the data using different methods. The already balanced dataset was used as input data for the classification algorithms, together with the test data that were used to obtain the predictive model. Finally, this model was evaluated with the metrics proposed in this work. Figure 1 shows the activities that were performed.

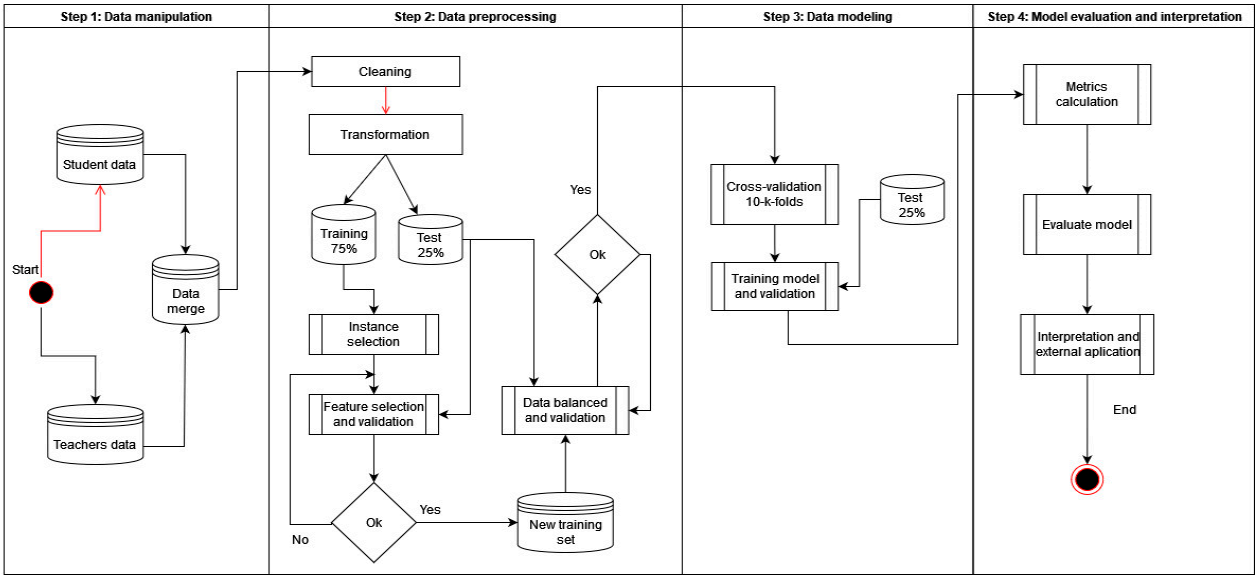



**Figure 1.** Diagram of activities performed. The processes conducted are described in four stages.

### 3.3. Metric Assessment

The metrics referred to in this section are used to evaluate the performance of the set of algorithms used to obtain predictive models. In Equation (4), the term $\alpha$ represents P(Tp) = Sensitivity, and $(1 - \beta)$ represents P(Tn) = Specificity [49].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \tag{1}$$

$$\text{Sensitivity} = \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{3}$$

$$\text{AUC} = \sum \left\{ (1 - \beta_i . \Delta\alpha) + \frac{1}{2}[(1 - \beta).\Delta\alpha] \right\} \tag{4}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

$$\text{Cohen}'\text{s Kappa} = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \tag{6}$$

$$LogLoss = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{i,j} \log(p_{i,j}) \tag{7}$$

### 3.4. Data Exploratory

The importance of data exploration is that it serves to understand the activity and behavior of the data. Visualization techniques have been used that detected significant information in the data; specifically, variables were examined according to each category of the class using graphs (Figure 2).

### 3.5. Data Preprocessing

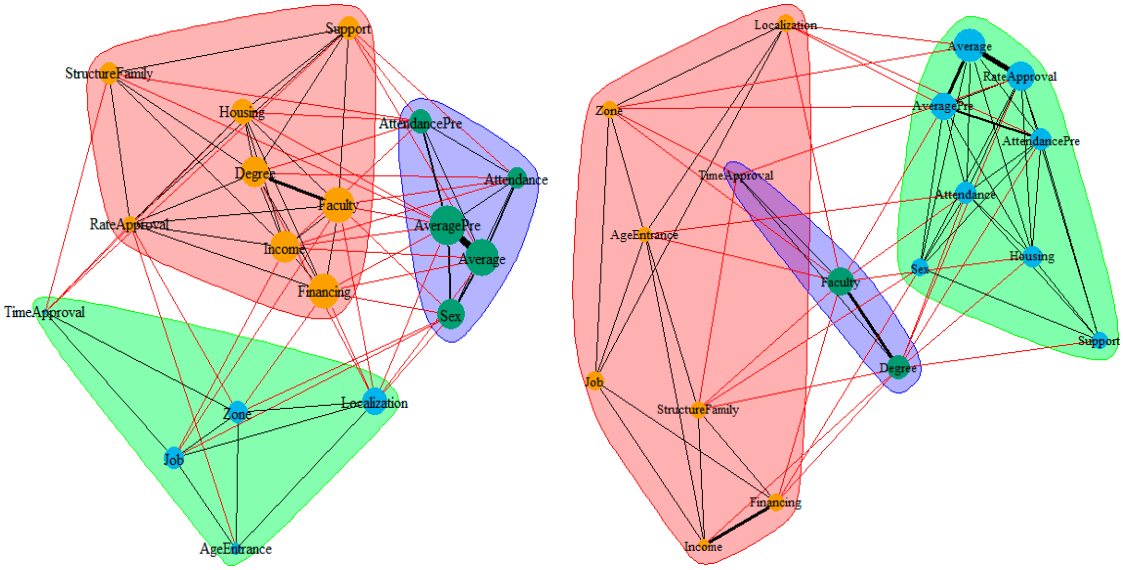
The importance of data preprocessing is to synthesize and achieve expeditious data. This fact has an important consequence for classification algorithms since the integrity of the data is gradually assessed by the hit rate, i.e., the number of true positives that the prediction algorithm can detect. Within this context, the aim is to obtain the set of features and instances that are close to a reasonable hit rate. The problem around which the data preprocessing revolves is the different search strategies such as sequential, random, and complete that are proposed for this task. The evaluation criterion is set with filtering (distance, information, dependency, and consistency), hybrid and wrapper methods [50–54].

The data preprocessing was divided into four phases. First, missing values in the data were replaced using the k-nearest neighbor's algorithm KNN_MV [55]. Second, unrepresentative instances were excluded using the "NoiseFiltersR" algorithm. Third, feature selection was studied with different algorithms and functions that have evaluated feature quality. Finally, data balancing was applied to avoid bias in the prediction model due to the small amount of minority class data.

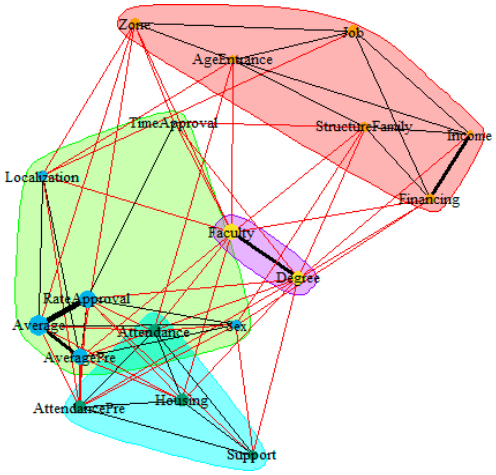
### 3.6. Missing Values

Data in their original form contain inconsistent data and often have missing values. That is, when the value of a variable is not stored, it is considered missing data. Multiple techniques have been developed to replace missing values. In general, statistical techniques of central tendency are usually used; for numerical values, the mean or median is used, while for nominal values, the "mode" is usually used. Another common technique is to remove the entire record from the dataset. Deletion can cause significant loss of information. Frequent techniques are easy to use and solve the problem of missing values, although, in data mining practice, there is a tendency to implement algorithms that solve this problem

by examining the entire dataset. Specifically, in this work, we have used the "rfImpute" function, which replaced missing values by the nearest neighbor technique that takes the class (target variable) as reference.




(**a**) Pass (**b**) Dropout




(**c**) Change

**Figure 2.** Undirected graph calculated from the correlation matrix (Pearson's method). Both the arcs and the adjacency matrix were filtered with cut-off points obtained from the weighted mean of the nodes (Pass = 0.0007804694, Dropout = 0.0061971, Change = 0.01684287). The graphs had weights associated with each of the arcs, and this weight fixed their density. Three groups of subfigures were separated according to the target variable (pass, dropout, change). Subfigure (**a**) showed three subgroups of variables (8, 5, 5) where a common variable overlaps. Cluster (**b**) showed three subgroups of variables (8, 3, 8); this subfigure lacks overlap. Group (**c**) showed four subgroups of variables (6, 7, 4, 2) overlapped by three common variables. On the other hand, red lines indicate a lower degree of association, while black lines and thickness indicate their strength of association.

### *3.7. Instance Selection*

Instance selection was also key in the data preprocessing, since poor-quality examples were eliminated by using the NoiseFiltersR algorithm [41], which filtered out the 5% of examples that were not within the data standard. In other words, when a value is at an unusual distance from the rest of the values in the dataset, it is considered an outlier or noise.

### *3.8. Feature Selection*

There is an important distinction to be made in this section since the generality and accuracy of the predictive model will depend on the quality of the variables. Therefore, it is crucial to decide which variables are relevant to include in the study. For this, we used nine feature selection algorithms among them: "LasVegas-LVF", "Relief" [56], "selectKBest", "hillClimbing", "sequentialBackward", "sequentialFloatingForward", "deepFirst", "geneticAlgorithm", and "antColony". On the other hand, the algorithms used distinct functions to value the attributes. Among the functions, we had "mutualInformation" [57], "MDLC" [58], "determinationCoefficient" [59], "GainRatio" [60], "Gini Index" [61], and "roughsetConsistency" [62,63]. The group of algorithms used for the study of significant characteristics obtained subgroups of variables that have been evaluated and are shown in Table 2 and Appendix C Table A3.

**Table 2.** Feature filtering by the "Relief" algorithm using different k and bestk filters. The lowest feature selection and the highest accuracy achieved by the C4.5 classification algorithm were established with the "bestk" filtering (10 variables).

| Filter | Variable | Value | Accuracy | Kappa | Sensitivity | Specificity | Precision | Recall | F1 |
|---|---|---|---|---|---|---|---|---|---|
| k = 9 | 11 | −0.002 | 0.75 | 0.56 | 0.83 | 0.85 | 0.79 | 0.83 | 0.80 |
| k = 7 | 11 | −0.001 | 0.76 | 0.5 | 0.82 | 0.85 | 0.78 | 0.82 | 0.80 |
| k = 5 | 11 | −0.003 | 0.74 | 0.52 | 0.80 | 0.83 | 0.77 | 0.80 | 0.78 |
| k = 3 | 14 | −0.001 | 0.76 | 0.56 | 0.82 | 0.85 | 0.79 | 0.82 | 0.80 |
| bestk | 10 | 0.062 | 0.79 | 0.62 | 0.85 | 0.87 | 0.81 | 0.85 | 0.83 |

### *3.9. Data Balancing*

Sample balancing is another important step in data preprocessing. Currently, there are several techniques for data balancing or resampling using Python software 3.9 and its scikit-learn library [33]. In this work, the following techniques have been studied: oversampling, combined, undersampling and ensemble. The first used the methods "Smote" [28] and "KMeansSMOTE" (oversampling with SMOTE, followed by undersampling with edited nearest neighbors) [64]. The second used both "Smote-ENN" and "Smote-Tomek" (oversampling with SMOTE) [65]. The third technique used was subsampling with the "RUS" method [66]. Finally, the ensemble technique used "EasyEnsemble" [67] and "Bagging". Specifically, new balanced training datasets were generated. All of this was from the initial training set, in which the different techniques and methods were used to balance the data (See Table 3).

**Table 3.** The table displays the distribution of data per class using different data balancing techniques, along with the corresponding imbalance ratio (IR) between the majority and minority classes. A higher IR indicates a more severe class imbalance problem.

| Algorithms Used | Classes | | | Overall | IR |
|---|---|---|---|---|---|
| | Dropout | Change | Pass | | |
| Origin data (not use algorithm) | 3.346 | 466 | 2.080 | 5.892 | 7.180 |
| Over (SMOTE) | 2.826 | 5.652 | 8.478 | 16.956 | 3 |
| Over (KMeansSMOTE | 5.655 | 8.481 | 2.829 | 16.965 | 2.997 |
| Combined (SMOTE-ENN) | 5.365 | 2.822 | 4.164 | 12.351 | 1.901 |
| Combined (SMOTE-Tomek) | 5.360 | 2.826 | 7.894 | 16.080 | 1.472 |
| Under (RUS) | 355 | 1.065 | 710 | 2.130 | 3 |
| Under (Tomelinks) | 2.439 | 4.229 | 3.874 | 10.542 | 1.733 |
| Ensembles (EasyEnsemble) | 2.826 | 5.017 | 4.662 | 12.505 | 1.775 |
| Ensembles (Bagging) | 2.826 | 5.017 | 4.662 | 12.505 | 1.775 |

### *3.10. Classification Algorithms*

The use of supervised classification techniques aims to achieve a prediction model that is highly accurate. Hence, several algorithms have been created that use different mathematical models to achieve the model. In this section, we detail the types of algorithms and a provide a brief description of how each works.

- Decision Trees: Consists of building a tree structure in which each branch represents a question about an attribute. New branches are created according to the answers to the question until reaching the leaves of the tree (where the structure ends). The leaf nodes indicate the predicted class; see [35].
- Support Vector Machine (SVM): A relatively simple supervised machine learning algorithm used in regression or classification related problems. In many cases, it is used for classification, although it is preferably useful for regression. Basically, SVM creates a hyperplane with boundaries between data types in a two-dimensional space; this hyperplane is nothing more than a line. In SVM, each datum in the dataset is plotted in an N-dimensional space, where N is the number of features/attributes of the data; see [68].
- Neural Network: Multilayer perceptrons (MLP) are the best known and most widely used type of neural network. They consist of neuron-like units, multiple inputs, and an output. Each of these units forms a weighted sum of its inputs, to which a constant term is added. This sum is then passed through a nonlinearity, usually called an activation function. Most of the time, the units are interconnected in such a way that they form no loop; see [69].
- Random Forest: A combination of tree predictors, where each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The use of random feature selection to split each node produces error rates that compare favorably with "Adaboost" but are more robust with respect to noise. The internal estimates control for error, strength, and correlation, and are used to show the response to increasing the number of features used in the split. Internal estimates are also used to measure the importance of variables; see [70].
- Gradient Boosting Machine: Gradient boosting is a machine learning technique used to solve regression or classification problems, which builds a predictive model in the form of decision trees. It develops a general gradient descent "boosting" paradigm for additive expansions based on any fitting criteria. Gradient boosting of regression trees produces competitive, very robust, and interpretable regression and classification procedures, especially suitable for the extraction of not-so-clean data; see [71].
- XGBoost: XGBoost is a distributed and optimized gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides parallel tree boosting (also known as GBDT, GBM) that solves many data science problems in a fast and accurate manner; see [72].
- Bagging: Predictor bagging is a method of generating multiple versions of a predictor and using them to obtain an aggregate predictor. Bagging averages the versions when predicting a numerical outcome and performs plural voting when predicting a class. Multiple versions are formed by making bootstrap replicas of the learning set and using them as new learning sets. Tests on real and simulated datasets show that bagging can provide a substantial increase in accuracy; see [73].
- Naïve Bayes: A probabilistic machine learning model used for classification tasks. The core of the classifier is based on Bayes' theorem: $P(A \mid B) = \frac{P(B|A)P(A)}{P(B)}$, which is the probability of $A$ occurring, given that $B$ has occurred. Here, $B$ is the evidence, and $A$ is the hypothesis. The assumption made here is that the predictors/features are independent. That is, the presence of a particular feature does not affect the other; see [74].

## 4. Results

In response to the research questions posed, different data preprocessing algorithms have been employed to reduce the dimensionality of the dataset, so that the classification algorithms obtain a simple and accurate predictive model. In the following sections, we study data preprocessing for feature selection first. Second, we study data balancing using different data balancing algorithms and, finally, the results using the metrics calculated from the confusion matrix where the performance of the algorithms was evaluated.

### *4.1. Data Preprocessing*

#### 4.1.1. Feature Selection

Prior to preprocessing, the dataset was separated into two parts: 75% of the total was selected for training data, and the other 25% for testing. The latter were used to evaluate the predictive model achieved by the classification algorithms, while the training set was subjected to preprocessing techniques to reduce dimensionality and obtain adequate data. In this sense, the work has focused on achieving simplicity and improving the accuracy of the predictive model, for which different feature and filter selection methods have been configured. Table 2 shows the results of the algorithm that obtained the lowest features; the rest of the runs of other algorithms and their results can be found in Appendix C.

In view of the cited works, in the studies of [15,28], relevant features in the data were examined to improve the predictive model, in line with these. Table 2 presents the results for the pre-selected feature set, where each evaluative filter and method rated the variables according to the performance metric. Specifically, the Relief method together with the "bestk" evaluative filter achieved better efficiency, i.e., higher accuracy with fewer variables. Based on these results, a new dataset with the new characteristics was established and used as input data for the data balancing phase described in the next section.

#### 4.1.2. Data Balancing

The importance of data balancing is fundamental to classification algorithms since the disparity of examples between one class and another can lead to bias in the prediction model. There are two common techniques for data balancing. The first is the oversampling of examples technique, in which the data are balanced to the same number of examples in the majority class. The second is to reduce the other classes to the same number of examples in the minority class. Both techniques, although not very efficient, are useful for obtaining primary results since the redistribution of the data is achieved with the judgment and experience of the data analyst. To some extent, this personalized judgment is avoided by the intervention of algorithms that perform data balancing. The algorithms augment, reduce or equalize the examples depending on the technique applied. From the above, Table 3 shows the data imbalance index according to the algorithms used. Thus, each algorithm generated a new balanced dataset that was used to train the classification algorithms.

### *4.2. Classification Algorithms*

In this section, we examine the effectiveness of the set of classification algorithms proposed for this work, which is related as a multiclass problem, that is, a dependent variable (class) with three types of outputs: Dropout, Change and Passed. For this reason, and as is common in supervised classification problems, two datasets have been used: the first, for the algorithms to learn and obtain a prediction model; and the second, to evaluate the effectiveness of the model obtained. Hence, we worked with two types of analysis: the first with the original data (without data preprocessing) and the second with the different datasets generated from the preprocessing techniques used.

It is difficult not to appreciate the importance of data preprocessing, as it provides classification algorithms with balanced and clean datasets. Obtaining the predictive model requires the algorithm to learn from the provided data (training set), as the effectiveness of the model will depend on it. Therefore, for the algorithm to achieve adequate learning, the cross-validation technique k-fold cross-validation (CV) was applied; this approach

randomly subdivided the training set into 10 folds with approximately equal size, and each fold, in turn, was fragmented into two sections: training and test. This was done so that at the end of training, the mean prediction was obtained from among the folds. On the other hand, to check what was learned by the algorithms, the metrics proposed in the section of methodology were used, which helped to discriminate the most effective predictive models. While it is true that effectiveness is fundamental to evaluate the predictive model, the comprehensibility of the model obtained is also important, since the experts evaluate the simplicity of the model.

Here, we present the best result of the classification algorithms that were achieved using the dataset balanced by the "EasyEnsemble" algorithm and the performance assessment of the classifiers using the ROC curve presented in Figure 3. The rest of the results with different datasets derived from the application of the data balancing algorithms are presented in Appendix B, Table A2.

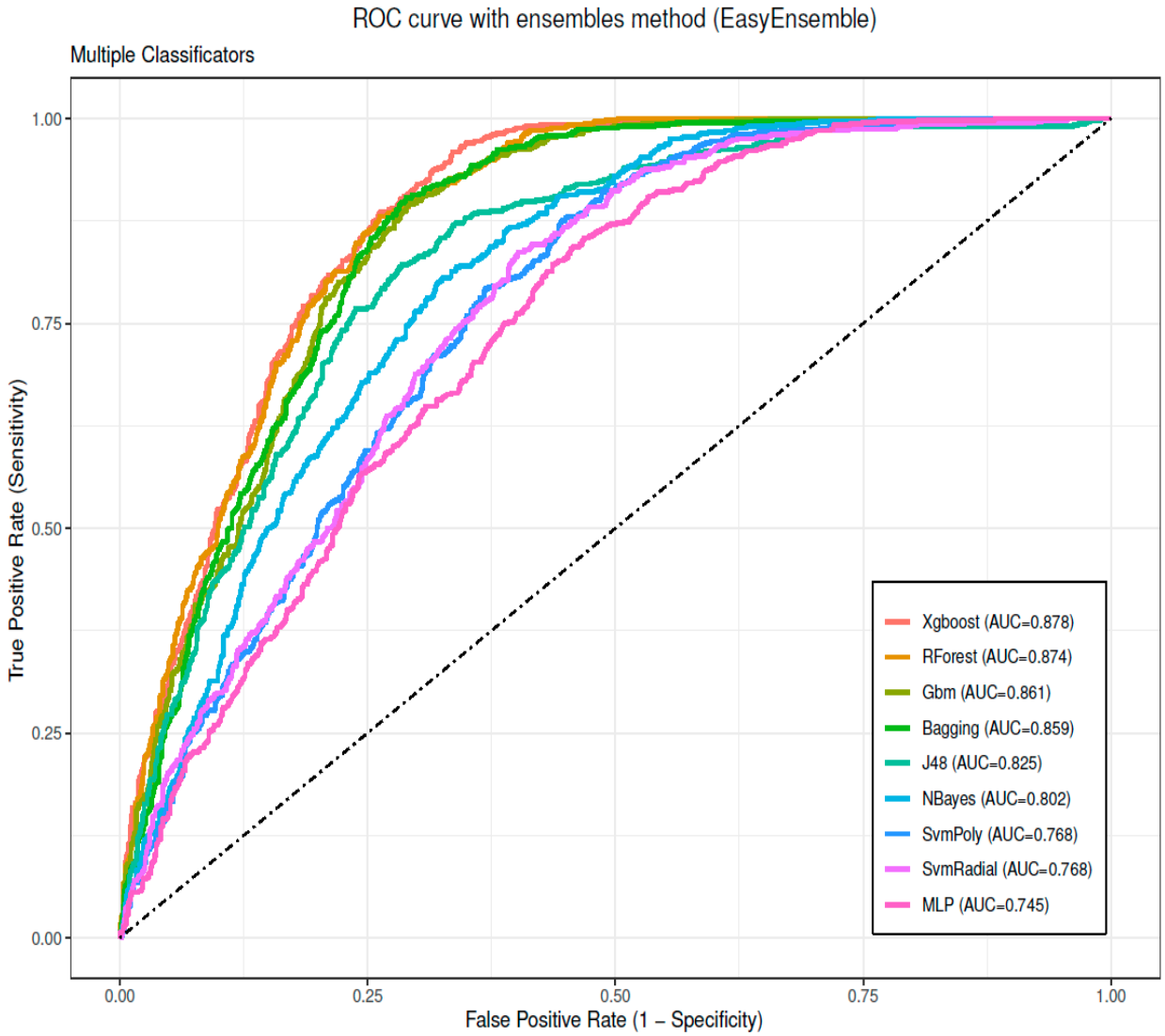



**Figure 3.** Performance of the group of algorithms by plotting the area under the AUC curve. On the ordinate axis is the true positive rate, and on the abscissa axis the false positive rate. The classifier lines above the diagonal (dashed line) represent good classification results (better than random), while those below represent bad results (worse than random). The best performance in classifying the test data examples was obtained by the XGBoost algorithm; two algorithms had an AUC above 0.87, the rest performed below 0.86. This performance clearly indicates the effectiveness of the predictive model against the test set.

In view of the results, Table 4 (raw data) and Table 5 (preprocessed data) show differences in the performance of the algorithms. Negative values −0.0214 and −0.0222 for precision and AUC, respectively, are evident. This negative effect between raw data and preprocessed data is a consequence of preprocessing, so data preprocessing should be interpreted not as a contradictory process but as an improvement of the predictive model by using fewer variables from the original set. Therefore, the advantage of applying data preprocessing has been observed.

**Table 4.** Preliminary results for the original dataset, omitting data preprocessing.

| Algorithms | Accuracy | Kappa | Sensitivity | Specificity | Precision | Recall | F1 | AUC | LogLoss |
|---|---|---|---|---|---|---|---|---|---|
| XGBoost | 0.8133 | 0.6617 | 0.8492 | 0.8861 | 0.8456 | 0.8492 | 0.8462 | 0.8997 | 0.3736 |
| RandomForest | 0.8163 | 0.6664 | 0.8523 | 0.8873 | 0.8428 | 0.8523 | 0.8468 | 0.8978 | NA |
| Gbm | 0.8062 | 0.6473 | 0.8460 | 0.8800 | 0.8352 | 0.8460 | 0.8401 | 0.8930 | 0.3925 |
| Bagging | 0.8008 | 0.6379 | 0.8423 | 0.8769 | 0.8291 | 0.8423 | 0.8351 | 0.8781 | NA |
| C4.5 | 0.7822 | 0.6039 | 0.8378 | 0.8642 | 0.8033 | 0.8378 | 0.8193 | 0.8308 | NA |
| NaiveBayes | 0.6549 | 0.3847 | 0.5215 | 0.8025 | 0.7622 | 0.5215 | 0.5059 | 0.8168 | NA |
| SvmRadial | 0.7284 | 0.4934 | 0.7781 | 0.8218 | 0.7673 | 0.7781 | 0.7709 | 0.7973 | NA |
| SvmPoly | 0.7165 | 0.4687 | 0.7571 | 0.8132 | 0.7685 | 0.7571 | 0.7616 | 0.7754 | 0.5484 |
| MLP | 0.6895 | 0.4501 | 0.7673 | 0.8143 | 0.7471 | 0.7673 | 0.7511 | 0.7621 | 0.5378 |

**Table 5.** Evaluation results of the predictive models obtained by the classification algorithms. The training set was balanced with the "EasyEnsemble" technique. Model validation was performed on the test dataset. The data were sorted according to the AUC column.

| Algorithms | Accuracy | Kappa | Sensitivity | Specificity | Precision | Recall | F1 | AUC | LogLoss |
|---|---|---|---|---|---|---|---|---|---|
| XGBoost | 0.7949 | 0.6299 | 0.8425 | 0.8753 | 0.8214 | 0.8425 | 0.8306 | 0.8775 | 6.3430 |
| RandomForest | 0.7925 | 0.6269 | 0.8444 | 0.8747 | 0.8205 | 0.8444 | 0.8305 | 0.8744 | NA |
| Gbm | 0.7752 | 0.5923 | 0.8318 | 0.8605 | 0.8043 | 0.8318 | 0.8171 | 0.8606 | 5.6340 |
| Bagging | 0.7752 | 0.5933 | 0.8268 | 0.8617 | 0.8088 | 0.8268 | 0.8168 | 0.8591 | NA |
| C4.5 | 0.7644 | 0.5803 | 0.8334 | 0.8594 | 0.7964 | 0.8334 | 0.8110 | 0.8249 | NA |
| SvmPoly | 0.6861 | 0.4094 | 0.7347 | 0.7919 | 0.7466 | 0.7347 | 0.7384 | 0.7679 | 4.1072 |
| SvmRadial | 0.6814 | 0.4073 | 0.7460 | 0.7920 | 0.7321 | 0.7460 | 0.7377 | 0.7676 | NA |
| MLP | 0.6539 | 0.4059 | 0.7620 | 0.8013 | 0.7462 | 0.7620 | 0.7360 | 0.7446 | 3.2832 |
| NaiveBayes | 0.6389 | 0.3850 | 0.6348 | 0.8022 | 0.7879 | 0.6348 | 0.6442 | 0.8018 | 6.3015 |

It should be noted that the logloss was lower with the original data than with the preprocessed data. The increase with the latter was due to the smaller imbalance between classes. That is, the smaller the imbalance between classes, the greater the logloss, due to the smaller proportion of observations in the minority class. Table 3 shows the imbalance index between the original set and the dataset preprocessed with "EasyEnsemble" (column IR: 7.18 and 1.775 respectively).

In Table 6, the confusion matrix of the best-scoring algorithm (XGBoost) aimed to explain the predicted values of the test dataset, and the prediction model obtained by the algorithm was established. First, the type II error or β type error was analyzed, where (a) the "Dropout" class had predicted values of 868 cases, of which 741 were correct, and 127 cases were classified as "Pass"; (b) the "Change" class had 126 cases, of which 115 were correct and 11 were classified as "Pass"; (c) the "Pass" class of the 679 predicted cases had 474 that were correct, four cases were classified as "Change", and 201 were classified as "Dropout". Secondly, the type I error or type α error was analyzed, where (a) the class "Dropout" had 942 cases, of which 741 were correct and 201 "Pass"; (b) the class "Change" had 119 cases, of which 115 were correct and four were classified as "Pass"; (c) the class "Pass" had 612 cases, of which 474 were correct, 11 were classified as "Change", and 127 were classified as "Dropout".

**Table 6.** Confusion matrix of the XGBoost algorithm. Here, the actual values (rows) are shown versus the values predicted by the classifier (columns).

| Actual \ Prediction | Dropout | Change | Pass | Total | Error Type II (β) |
|---|---|---|---|---|---|
| Dropout | 741 | 0 | 127 | 868 | 0.8536 |
| Change | 0 | 115 | 11 | 126 | 0.9126 |
| Pass | 201 | 4 | 474 | 679 | 0.6980 |
| Total | 942 | 119 | 612 | 1673 | μ = 0.8214 |
| Error Type I (α) | 0.7866 | 0.9663 | 0.7745 | μ = 0.8431 | |

Overall, a more efficient predictive model was obtained with the XGBoost classification algorithm. In the work of [75], they highlight that the random forest algorithm obtained a better result in accuracy (ACC: 0.81) using only 10 features of the original dataset, pointing out the importance of improving academic performance and increasing the graduation rate of the students of the educational center. Consequently, it is necessary to consider that the accuracy of the model increases, and its complexity needs to be explainable as well. In this context, we looked for a way to apply a simple and readable method. The decision tree provides a simple rule-based model that improves comprehensibility. The use of the decision tree, although less efficient, is very easy to interpret. Figure 4 shows the decision tree generated from the training data and Figure 5 shows the important variables.

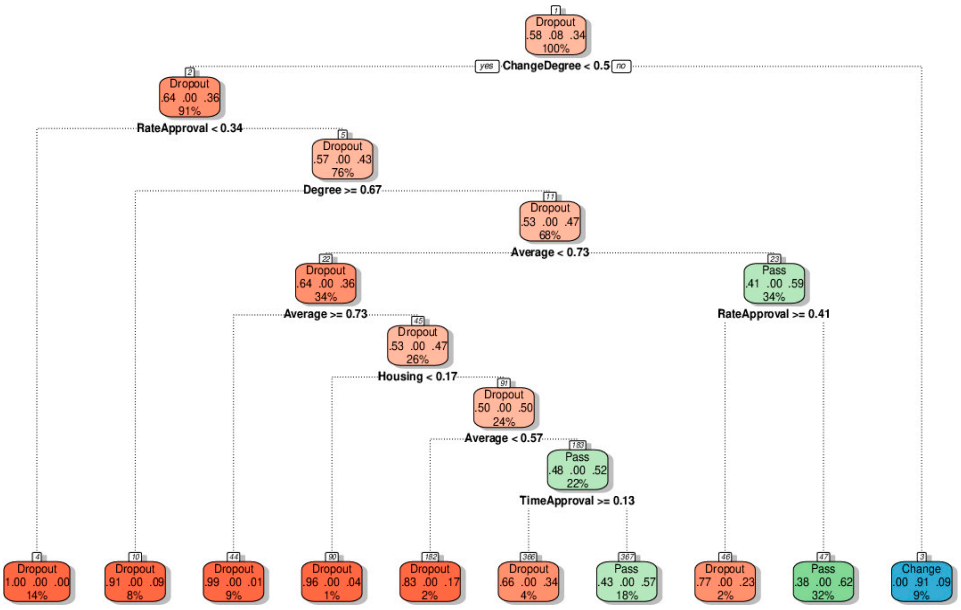



**Figure 4.** The decision tree drawn is based on the rules obtained. The nodes represent the class. The three decimal values within the node represent the probability of each class with respect to the evaluation of the rule. In turn, the total percentage of cases for the rule (cover) is shown. Below the node, the condition of the rule is displayed.

### *4.3. Static Comparison of Several Classifiers*

Formally, statistical significance is defined as a probability measure to assess experiments or studies. Ronald Fisher promoted the use of the null hypothesis [76], establishing a significance threshold of 0.05 (1/20) to determine the validity of the results obtained in empirical tests. In this way, it is guaranteed that the provenance of their results is not due to chance coincidences. In the work of Demšar [77], the statistical significance of different classification algorithms and real-world datasets was validated by different empirical tests. In this context, the nonparametric Friedman and Wilcoxon tests were used, which are suitable for this type of analysis because they both do not skimp on the normal distribution

of the data or on the homogeneity of variances, making them suitable for studies with data of a real or unmanipulated nature.

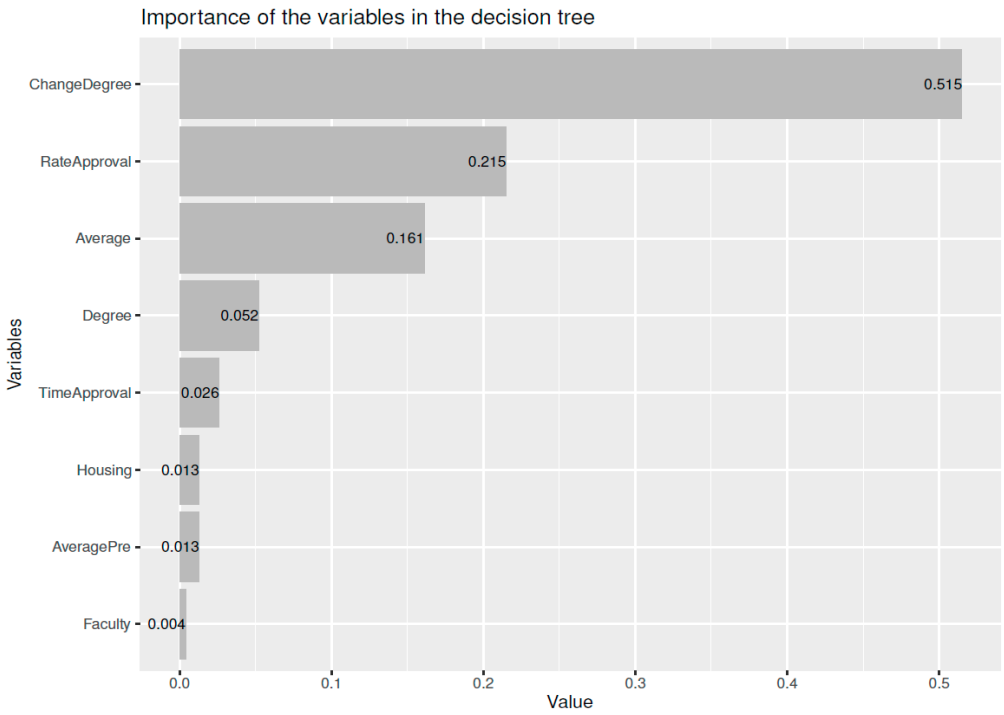



**Figure 5.** The importance of the variable is calculated by summing the decrease in error when divided by a variable. Thus, the higher the value, the more the variable contributes to improve the model, so the values are bounded between 0 and 1.

Prior to the calculation of the nonparametric tests, the results matrix of the group of algorithms and the datasets was organized, using the area under the curve (AUC, see Appendix D Table A6) as the metric. The significance threshold was set at 0.05 for the Friedman and Wilcoxon tests to determine if there were significant differences between more than two dependent groups. To perform the empirical tests, we used the null hypothesis H0: there are no significant differences between the groups of algorithms, and the alternative hypothesis, Ha: there is at least one significant difference between the groups of algorithms. The results of the Friedman test yielded a chi-square ($\chi^2$) of 52.305 with 8 degrees of freedom and a $p$-value of $1.47 \times 10^{-8}$ (See Appendix D, Table A4). Since the $p$-value was below the threshold, the null hypothesis was rejected, and the alternative hypothesis was accepted, confirming the existence of significant differences. Next, a pairwise comparison of algorithms will be performed using the Wilcoxon test to assess the significance of these differences.

The above analysis established that there were significant differences, so a test was performed for each pair of algorithms using the Wilcoxon test, which is a Friedman post hoc test and is presented in Table 7, where the $p$-values obtained are shown.

According to the results, significant differences were found in RF vs. Gbm (0.063); C4.5 vs. NaiveBayes (0.612); SVMRadial vs. SVMPoly (0.398); SVMRadial vs. MLP (0.091); and SVMPoly vs. MLP (0.128) (See Appendix D, Table A5 for detailed results). In [78–80], opinions on statistics and significance tests have been discussed, because they are often misused, either by misinterpretation or by overemphasizing their results. It should be stated that statistical tests provide some assurance of the validity and non-randomness of the results [77].

**Table 7.** Wilcoxon signed rank test.

| | XGBoost | RF | Gbm | Bagging | C4.5 | NaiveBayes | SvmRadial | SvmPoly |
|---|---|---|---|---|---|---|---|---|
| RF | 0.018 | - | | | | | | |
| Gbm | 0.018 | 0.063 * | - | | | | | |
| Bagging | 0.018 | 0.018 | 0.018 | - | | | | |
| C4.5 | 0.018 | 0.018 | 0.018 | 0.018 | - | | | |
| NaiveBayes | 0.018 | 0.018 | 0.018 | 0.018 | 0.612 * | - | | |
| SvmRadial | 0.018 | 0.018 | 0.018 | 0.018 | 0.028 | 0.018 | - | |
| SvmPoly | 0.018 | 0.018 | 0.018 | 0.018 | 0.028 | 0.018 | 0.398 * | - |
| MLP | 0.018 | 0.018 | 0.018 | 0.018 | 0.043 | 0.018 | 0.091 * | 0.128 * |

* Reject the null hypothesis.

## 5. Discussion

This paper explores and discusses three research questions related to machine learning techniques that are applied to achieve a predictive model with greater accuracy and readability, in addition to the study of factors that lead to the academic success of university students when they finish the first course. The answers to the questions posed are detailed.

RQ1: Which balancing and feature selection technique is relevant for supervised classification algorithms? In general, it is evident that with the increase in variables, the accuracy of the model increases, and so does its complexity, since the classification algorithms improve performance, although the readability of the model decreases. Against this in the work of Alwarthan [24], they apply recursive feature elimination (RFE) with Pearson correlation coefficient, RFE with mutual information and GA to find relevant features, in addition to class balancing using SMOTE-TomekLink to build the final prediction model. The relevant variables were related to English courses and GPA, as well as students' social variables. Alwarthan [24] used 68 features and achieved 93% accuracy with the initial results, while feature filtering detected 44 relevant variables and 90% accuracy. On the other hand, they analyzed eight relevant characteristics that achieved 77% accuracy; the variables were directly related to the academic performance of the student body.

In [6], the filtering of characteristics using the Gini index was proposed, from which seven characteristics were selected, achieving 79% accuracy using the random-forest algorithm. These results were very similar to ours, but far from being explainable, due to the bias derived from the imbalance of the data. In the proposal made in this study, different data processing techniques were used to obtain an expeditious dataset. On the one hand, the instance filtering method was considered to reduce duplicate or noisy observations by 5%. On the other hand, for feature group filtering, six methods were used, and five filters were applied, with which an accuracy between 58% and 78% was achieved. On the other hand, when applying the "ReliefF" method, 10 features were obtained with an accuracy of 79% (algorithm C4.5). In contrast, with the literature presented, the analyzed datasets had accuracy values below 84% and 32 features on average. The difference with what is proposed in this work is greater than 5% in accuracy, initially attractive. However, the handling of 22 additional features generates a robust and poorly explainable model for decision support.

Consequently, data balancing as part of data preprocessing was crucial to achieve a robust predictive model. The literature reviewed generically posits data balancing as a step prior to feature filtering. The approach taken so far is to obtain a filtered dataset (instances and features) and then apply data balancing. Among the best classification accuracies achieved by the data balancing methods, a range between 73% and 79% was obtained. The "EasyEnsemble" method obtained the best accuracy, AUC and logloss. The latter was far from the original data, as the imbalance rate was high. For example, the imbalance rates (IR) of the original data (7.35 IR) for undergraduate academic statuses (dropout, change and pass) were 57%, 7% and 36%, while for the balanced data (1.75 IR), they were 23%, 40% and 37% with synthetic observations. The accuracy of the XGBoost model with balanced

data was approximately 80%. In summary, the proposed data preprocessing made the dataset unbiased and the predictive model simple and explainable.

RQ2: Which predictive model best discriminates students' academic success? Currently, there are several supervised algorithms used in higher education to predict different educational contexts in higher education. Specifically, the best discrimination was performed by the XGBoost algorithm. This criterion was based first on the values collected with the predictive model, where the accuracy value was 79.49% and the AUC was 87.75%. Sensitivity = 84.25%, which indicated the rate of positive examples that the algorithm was able to classify, while specificity = 87.53% for negative examples. Next, the logloss metric measuring computational cost had 0.3736 and an imbalance rate of 7.18 with the original dataset. However, the logloss value went to 6.34 with the preprocessed dataset and an imbalance rate of 1.775, i.e., lower computational cost and a higher data imbalance rate were inversely proportional to the performance of the predictive model. Although the predictive model obtained using XGBoost is poorly explainable due to its high complexity, it performed better by classifying examples from the test set. Explainability of the predictive model was obtained when the decision tree was applied to the training set to obtain a predictive model based on rules (If, Then) and readable for decision makers.

Similarly, [6,16–19,24,75] converge in their predictions on higher education data using classifiers such as Random Forest (RF), SVM, Neural Networks and decision trees. Likewise, linear regression or logistic regression was used to obtain predictive models that detect failure, success, or academic performance early enough [1,81], or in turn, semi-supervised learning to obtain patterns in students who managed to pass the courses for a university degree [22]. Being the main objective to achieve very attractive and reliable accuracies, undoubtedly, accuracy always comes hand in hand with the quantity and quality of the data. For example, Gil [38] obtained accuracy rates with "random forest" of 77%, 91% and 94% with features of 30, 44 and 68, respectively, where the positive correlation between number of features and accuracy was evidenced. That said, in our results, accuracies very close to 80% were achieved with only 10 features and a completely readable model (10 rules).

RQ3: Which factors are determinants of students' academic success? As part of the development of this study, variables that play a significant role in the academic success of students were found. Specifically, the variables ChangeDegree, RateApproval, Average, and Degree were determinants for the prediction model obtained. These findings are close to the results obtained by Alturki [34], where individual results from the third and fourth semester were examined, both with accuracies of 63.33% (six variables) and 92.6% (nine variables), respectively. The influential variables were grade point average, number of credits taken and academic assessment performance, applying the selection of characteristics for each academic semester. Similarly, Alyahyan [23] identified variables related to GPA and key subjects that detect student performance early enough. As detailed by Beaulac [39] in their study, they identified variables associated with undergraduate degree completion as a first group of variables, whereas the second group of variables was related to the type of major. In summary, the first-year students opt for computer and English related subjects to reach their academic achievement, i.e., characteristics related to academic performance.

Specifically, data preprocessing provides as input an expedited dataset for classification algorithms to achieve an adequate predictive model. Although the results in the reviewed literature resemble ours, and these can be improved by inducing endogenous or exogenous variables for the model to achieve more optimal results, the results can also be improved by over-fitting parameters in the algorithms. It is also worth mentioning that, for example, Ismanto [82] obtained an RF prediction model with an accuracy higher than 90% without preprocessing the data, which resulted in a complex predictive model due to its explainability. Therefore, even if the model obtains the highest accuracy, the prediction bias can also be extended if the parameters are over-fitted or the data preprocessing phase is omitted.

Kaushik [83] has defined feature selection as increasing the quality in the data to facilitate better results, all according to the proposed method set of techniques for feature selection in educational data. What is applied in this paper fits with Kaushik's perspective.

It is important to anticipate early enough and with general quality characteristics to take effective countermeasures, providing timely warnings to students to achieve academic success. In this way, the percentage of underachieving students can be reduced, and appropriate counseling and intervention can be provided to them by the college.

The results provide conclusive support for the anticipation of college completion [84–86], which is essential to assist students in the learning process and ensure their academic success. Thus, taking advantage of the fact that predictions made early enough by machine learning manage to reveal possible difficulties or improvements from students' historical data, its effective use requires building specific strategies [84]. Consequently, the application of the knowledge obtained from the data is leveraged, for example, in constant monitoring or continuous tracking that acts as a tool to assess progress in academic performance, class attendance, extracurricular activities and other key indicators [87]. Other strategies include personalized tutorial support or intervention plans, remediation and other resources for students who have demonstrated compelling needs [88,89]. Machine learning, along with other data analysis techniques, offers valuable suggestions for targeted interventions for the benefit of students, with the goal of helping them achieve academic success in the shortest possible time. The results presented support the authenticity of the analyses performed, as the information is not based on mere coincidences, but on real data. In this context, significant tests were performed using statistical methods such as the nonparametric Friedman and Wilcoxon test, which are widely recognized for comparing the performance of machine learning algorithms [77,90,91]. Although these tests are not recommended for a comprehensive study, due to the need to conform to other assumptions, some authors have deepened their analysis and proposed alternatives to the tests [92,93]. In summary, significant tests are essential for a solid and objective interpretation of the results obtained.

## 6. Conclusions

In response to the research questions, the effectiveness of the prediction model lies in the good practice conducted in the data preprocessing phase. Hence, the importance of obtaining an expeditious dataset is crucial. Unlike the methodologies reviewed in the literature, our applied methodology avoided bias in the accuracy rates of the predictive model, as well as in the academic status (class). In fact, both the robust predictive model achieved by means of XGBoost as well as the simplified decision tree model proved to be effective. The simplified predictive model was able to detect students with high potential for academic success in seven out of ten cases, while the robust model detected them in eight out of ten cases. The simplification and explainability of the model were based on a set of rules obtained from the decision tree used, to make them understandable and provide them to academic experts as suggestions for decision making. Overall, this study provides valuable information on the factors underlying college students' academic success expectations and highlights the importance of effective data preprocessing and model simplification techniques for making accurate, meaningful, and understandable predictions about college students' academic success.

## 7. Limitations

The main limitation of this work was the absence of variables that help to have consistent measurements in the classification algorithms in terms of gender, scholarships, and financial aid, since it is important to analyze the evaluation of equity and discrimination aspects in the decisions made by the algorithms to build the predictive model.

## 8. Future Work

Looking ahead, we intend to explore how the knowledge extracted in this work and the university practices applied with this knowledge can influence classroom management, with the aim of improving students' academic outcomes and reducing the disparity in educational opportunities. To this end, we propose studies related to (i) examining how the personalization of predictive models can be adapted to the phenotype (charac-

teristics) of the student body, where the objective is to examine the use of fuzzy logic to make uncertainty flexible and how the fuzzy model can manage the university context; (ii) designing early warning systems to intervene early and prevent failure or dropout; and (iii) other approaches, such as longitudinal studies, that aid evaluation and effectiveness over time to adjust the models as needed.

**Author Contributions:** Individual contributions: J.H.G.-F. and J.C.: conceptualization, methodology, software, validation, formal analysis, research, writing—writing of the original draft, writing—revising and editing, visualization, supervision; J.G.-M.: resources, writing—revising and editing, visualization, support, project administration. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** The work is supported as part of the UTEQ-FOCICYT IX-2023/29 project "Factors that affect the completion of time to degree and affect the retention of UTEQ students" approved by the twentieth resolution made by the Honorable University Council dated 14 February 2023. The study keeps the respective confidentiality of the information stipulated in the Organic Law for the Protection of Personal Data of the Republic of Ecuador, in addition to the application of the respective "Code of Ethics for Officials and Servants Designated or Contracted by the Universidad Tecnica Estatal de Quevedo" approved by the Honorable University Council on 6 September 2011. Therefore, the research group has declared this research approved for publication in any journal with the document CERT-ETHIC-001-2023.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The dataset is not available but can be obtained from the corresponding author upon reasonable request.

**Acknowledgments:** We would like to express our deep appreciation to the authorities of the IES for authorizing and allowing access, exploration, and analysis of the information, especially for the support provided by the project "Factors that influence the completion of the time to degree and affect the retention of students at UTEQ", headed by Javier Guaña-Moya, and Efraín Díaz Macías. The work is supported as part of the UTEQ-FOCICYT IX-2023/29 project.

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

## Appendix A

This section presents the information used in the work. The dataset used consists of data such as career, class attendance, students' academic performance and socioeconomic information. Numerical and categorical data are according to each variable.

**Table A1.** Description of the dataset used for the study.

| Variable Names | Values | Description | Type |
|---|---|---|---|
| Faculty | 1–5 | Names of the faculties. | Categorical |
| Degree | 1–27 | Names of the university degrees. | Categorical |
| Sex | 1. Male,<br>2. Female | Sex of students. | Categorical |
| Age Entrance | 16–50 | Age at entrance to university. | Numeric |
| Support | 1. Public<br>2. Private | Type of financial support from the high school where the student completed high school. | Categorical |
| Localization | 1. Local,<br>2. Outside of Quevedo,<br>3. Other Province | The geographical area of the school where the student finished high school. | Categorical |
| AveragePre | 0–10 | Average of the grades of the university leveling program (Pre-university/Admission/Selectivity). | Numeric |

**Table A1.** *Cont.*

| Variable Names | Values | Description | Type |
|---|---|---|---|
| Housing | 1. Own housing,<br>2. Rental,<br>3. Mortgaged,<br>4. Borrowed. | This variable is related to the usufruct of the housing where the student and his family live. | Categorical |
| ChangeDegree | 1. Yes,<br>2. No. | This variable describes whether the student has changed degrees when repeating the first year. | Categorical |
| Class | 1. Dropout,<br>2. Change,<br>3. Pass. | Variable with the student's academic status at the end of the university degree. | Categorical |
| AttendancePre | 0–100 | Pre-university attendance percentage. | Numeric |
| Average | 0–10 | Average of the subjects taught in the first year. | Numeric |
| Attendance | 0–100 | Average of the student's attendance percentage in all subjects enrolled. Must meet the minimum attendance percentage of 70%. | Numeric |
| TimeApproval | 1–3 | Number of enrollments used by the student to pass the first course. | Numeric |
| RateApproval | 0–3 | Weighting of the effort in the exams to pass the subjects; the first exam (recovery) has a value of 0.25, while the second one has a value of 0.75. | Numeric |
| CounterDegree | 0–2 | The number of college courses in which the student was enrolled. | Numeric |
| StructureFamily | 1. I am independent<br>2. Only with mom,<br>3. Only with dad,<br>4. Both parents,<br>5. Couple,<br>6. Other relative. | Variable associated with the student's family structure. | Categorical |
| Job | 1. Does not work,<br>2. Full time,<br>3. Part-time,<br>4. Part-time by the hour,<br>5. Occasionally. | This variable is linked to the student's work or occupational situation. | Categorical |
| Financing | 1. Family support, with 1 or 2 children studying.<br>2. Self-employed (own account).<br>3. Family support, with more than three children studying.<br>4. Loan, scholarship, or current credit. | This variable is related to the student's economic disposition to pay for the academic year. | Categorical |
| Zone | 1. Outside of Quevedo,<br>2. Urban,<br>3. Slum,<br>4. Rural. | Describes the geographic district where the student lives. | Categorical |
| Income | 1. More than \$400,<br>2. Between \$399 and \$200,<br>3. Between \$199 and \$100,<br>4. Less than or equal to \$99. | Monthly cash income (approximate) of the family nucleus. | Categorical |

## Appendix B

Table A2 presents various results from the calculation of the metrics applied to the group of classification algorithms. The results presented in this appendix are complementary trainings, as six different balancing techniques were used to generate new datasets that were contributed to train and achieve effective predictive models. Each technique applied balancing methods related to oversampling, undersampling and combined balancing based on the SMOTE algorithm. The "EasyEnsemble" data balancing was the best performing of the algorithms and has been presented in the Results section as part of the data input supply for the group of classification algorithms to obtain the predictive model.

**Table A2.** Performance results of the classification algorithms that trained and tested the predictive models using new datasets constructed using the data balancing algorithms.

| Bal. | Algorithms | Acc. | Kappa | Sensi. | Speci. | Preci. | Recall | F1 | AUC | LogLoss |
|---|---|---|---|---|---|---|---|---|---|---|
| SMOTE | XGBoost | 0.7878 | 0.6214 | 0.8472 | 0.8740 | 0.8237 | 0.8472 | 0.8318 | 0.8743 | 6.7890 |
| | RF | 0.7812 | 0.6118 | 0.8418 | 0.8720 | 0.8143 | 0.8418 | 0.8229 | 0.8671 | |
| | Gbm | 0.7723 | 0.5984 | 0.8446 | 0.8679 | 0.8105 | 0.8446 | 0.8205 | 0.8575 | 5.0183 |
| | Bagging | 0.7687 | 0.5887 | 0.8315 | 0.8633 | 0.8045 | 0.8315 | 0.8135 | 0.8546 | |
| | C4.5 | 0.7639 | 0.5771 | 0.8248 | 0.8577 | 0.8008 | 0.8248 | 0.8101 | 0.7936 | |
| | SvmPoly | 0.6999 | 0.4740 | 0.7819 | 0.8242 | 0.7618 | 0.7819 | 0.7631 | 0.7640 | 5.8172 |
| | SvmRadial | 0.6970 | 0.4681 | 0.7835 | 0.8215 | 0.7587 | 0.7835 | 0.7630 | 0.7649 | |
| | MLP | 0.6545 | 0.4190 | 0.7779 | 0.8087 | 0.7552 | 0.7779 | 0.7350 | 0.7512 | 5.0928 |
| | NaiveBayes | 0.6198 | 0.3802 | 0.7478 | 0.7990 | 0.7817 | 0.7478 | 0.6957 | 0.8040 | 5.0538 |
| KMeans.SMOTE | XGBoost | 0.7956 | 0.6366 | 0.8565 | 0.8802 | 0.8262 | 0.8565 | 0.8367 | 0.8702 | 6.3079 |
| | RF | 0.7794 | 0.6080 | 0.8420 | 0.8702 | 0.8125 | 0.8420 | 0.8226 | 0.8600 | |
| | Gbm | 0.7693 | 0.5929 | 0.8396 | 0.8660 | 0.8060 | 0.8396 | 0.8160 | 0.8515 | 5.1901 |
| | Bagging | 0.7681 | 0.5860 | 0.8228 | 0.8620 | 0.8049 | 0.8228 | 0.8103 | 0.8467 | |
| | C4.5 | 0.7663 | 0.5828 | 0.8259 | 0.8605 | 0.8036 | 0.8259 | 0.8113 | 0.7979 | |
| | SvmPoly | 0.6946 | 0.4613 | 0.7751 | 0.8187 | 0.7548 | 0.7751 | 0.7584 | 0.7616 | 5.8277 |
| | SvmRadial | 0.6892 | 0.4499 | 0.7717 | 0.8139 | 0.7492 | 0.7717 | 0.7551 | 0.7591 | |
| | MLP | 0.6712 | 0.4229 | 0.7703 | 0.8045 | 0.7353 | 0.7703 | 0.7452 | 0.7424 | 4.9865 |
| | NaiveBayes | 0.6067 | 0.3644 | 0.7505 | 0.7933 | 0.7804 | 0.7505 | 0.6862 | 0.7970 | 5.0905 |
| SMOTE.Tomek | XGBoost | 0.7914 | 0.6278 | 0.8474 | 0.8766 | 0.8241 | 0.8474 | 0.8320 | 0.8665 | 6.5445 |
| | Bagging | 0.7747 | 0.5970 | 0.8269 | 0.8656 | 0.8090 | 0.8269 | 0.8148 | 0.8468 | |
| | Gbm | 0.7741 | 0.6029 | 0.8430 | 0.8705 | 0.8137 | 0.8430 | 0.8199 | 0.8577 | 4.9046 |
| | RF | 0.7717 | 0.5922 | 0.8295 | 0.8639 | 0.8050 | 0.8295 | 0.8139 | 0.8562 | |
| | C4.5 | 0.7579 | 0.5639 | 0.8088 | 0.8526 | 0.7947 | 0.8088 | 0.8001 | 0.7623 | |
| | SvmPoly | 0.6975 | 0.4722 | 0.7822 | 0.8242 | 0.7627 | 0.7822 | 0.7619 | 0.7634 | 5.7663 |
| | SvmRadial | 0.6910 | 0.4579 | 0.7749 | 0.8182 | 0.7550 | 0.7749 | 0.7565 | 0.7633 | |
| | MLP | 0.6724 | 0.4459 | 0.7885 | 0.8182 | 0.7631 | 0.7885 | 0.7483 | 0.7592 | 4.8305 |
| | NaiveBayes | 0.6372 | 0.4053 | 0.7622 | 0.8079 | 0.7805 | 0.7622 | 0.7104 | 0.7959 | |

**Table A2.** *Cont.*

| Bal. | Algorithms | Acc. | Kappa | Sensi. | Speci. | Preci. | Recall | F1 | AUC | LogLoss |
|---|---|---|---|---|---|---|---|---|---|---|
| SMOTE.ENN | XGBoost | 0.7478 | 0.5573 | 0.8216 | 0.8542 | 0.7933 | 0.8216 | 0.7988 | 0.8335 | 5.9690 |
| | Gbm | 0.7406 | 0.5481 | 0.8239 | 0.8519 | 0.7923 | 0.8239 | 0.7965 | 0.8230 | 5.2251 |
| | RF | 0.7394 | 0.5492 | 0.8285 | 0.8534 | 0.7962 | 0.8285 | 0.7972 | 0.8192 | |
| | Bagging | 0.7352 | 0.5387 | 0.8179 | 0.8485 | 0.7908 | 0.8179 | 0.7926 | 0.8165 | |
| | C4.5 | 0.7310 | 0.5274 | 0.8109 | 0.8429 | 0.7828 | 0.8109 | 0.7888 | 0.7548 | |
| | SvmRadial | 0.6880 | 0.4590 | 0.7846 | 0.8198 | 0.7594 | 0.7846 | 0.7589 | 0.7511 | |
| | SvmPoly | 0.6880 | 0.4598 | 0.7809 | 0.8206 | 0.7608 | 0.7809 | 0.7568 | 0.7490 | 5.5081 |
| | MLP | 0.6665 | 0.4398 | 0.7875 | 0.8169 | 0.7661 | 0.7875 | 0.7436 | 0.7650 | 4.7577 |
| | NaiveBayes | 0.6186 | 0.3810 | 0.7615 | 0.7989 | 0.7428 | 0.7615 | 0.6858 | 0.7764 | 4.2434 |
| RUS | Gbm | 0.7346 | 0.5346 | 0.8188 | 0.8455 | 0.7861 | 0.8188 | 0.7938 | 0.8102 | 5.1165 |
| | XGBoost | 0.7328 | 0.5344 | 0.8205 | 0.8466 | 0.7886 | 0.8205 | 0.7931 | 0.8173 | 4.4343 |
| | RF | 0.7304 | 0.5303 | 0.8187 | 0.8450 | 0.7868 | 0.8187 | 0.7914 | 0.8153 | |
| | Bagging | 0.7197 | 0.5062 | 0.8034 | 0.8346 | 0.7731 | 0.8034 | 0.7813 | 0.7962 | |
| | C4.5 | 0.6987 | 0.4954 | 0.8131 | 0.8387 | 0.7957 | 0.8131 | 0.7667 | 0.7764 | |
| | SvmRadial | 0.6629 | 0.4081 | 0.7634 | 0.7992 | 0.7249 | 0.7634 | 0.7368 | 0.7360 | |
| | SvmPoly | 0.6605 | 0.4040 | 0.7622 | 0.7976 | 0.7275 | 0.7622 | 0.7374 | 0.7348 | 4.3219 |
| | MLP | 0.6402 | 0.3871 | 0.7610 | 0.7950 | 0.7320 | 0.7610 | 0.7251 | 0.7304 | 2.9982 |
| | NaiveBayes | 0.6031 | 0.3575 | 0.7428 | 0.7907 | 0.7773 | 0.7428 | 0.6827 | 0.7841 | |

## Appendix C

This table presents the results of the filtering of characteristics using the different methods proposed in the study. Each method, according to its nature, filtered the group of variables that best represented the data. Then, the group of variables was evaluated with the C4.5 classification algorithm.

**Table A3.** Selection of characteristics used for the evaluation of the best group of variables. The best group of variables was selected by the RelefFbestK algorithm.

| Filter | Var. | Method | Value | Acc. | Kappa | Sensi. | Speci. | Preci. | Recall | F1 |
|---|---|---|---|---|---|---|---|---|---|---|
| Roughset consistency | 11 | Las Vegas | 1.00 | 0.68 | 0.39 | 0.53 | 0.79 | 0.65 | 0.53 | 0.56 |
| | 9 | SelectKBest | 0.02 | 0.67 | 0.36 | 0.47 | 0.78 | | 0.47 | |
| | 8 | HillClimbing | 1.00 | 0.62 | 0.21 | 0.41 | 0.73 | | 0.41 | |
| | 9 | Sequential Backward | 1.00 | 0.67 | 0.36 | 0.47 | 0.78 | | 0.47 | |
| | 9 | Sequential Floating Forward | 1.00 | 0.67 | 0.36 | 0.47 | 0.78 | | 0.47 | |
| | 10 | Genetic Algorithm | 1.00 | 0.67 | 0.34 | 0.47 | 0.78 | | 0.47 | |
| | 20 | AntColony | 1.00 | 0.78 | 0.60 | 0.83 | 0.86 | 0.81 | 0.83 | 0.82 |
| Determination coefficient | 13 | Las Vegas | 0.48 | 0.72 | 0.46 | 0.60 | 0.82 | 0.70 | 0.60 | 0.62 |
| | 9 | SelectKBest | 0.06 | 0.67 | 0.36 | 0.47 | 0.78 | | 0.47 | |
| | 20 | HillClimbing | 0.48 | 0.78 | 0.60 | 0.83 | 0.86 | 0.81 | 0.83 | 0.82 |
| | 20 | Sequential Backward | 0.48 | 0.78 | 0.60 | 0.83 | 0.86 | 0.81 | 0.83 | 0.82 |
| | 20 | Sequential Floating Forward | 0.48 | 0.78 | 0.60 | 0.83 | 0.86 | 0.81 | 0.83 | 0.82 |
| | 20 | Genetic Algorithm | 0.48 | 0.78 | 0.60 | 0.83 | 0.86 | 0.81 | 0.83 | 0.82 |
| | 20 | AntColony | 0.48 | 0.78 | 0.60 | 0.83 | 0.86 | 0.81 | 0.83 | 0.82 |

**Table A3.** *Cont.*

| Filter | Var. | Method | Value | Acc. | Kappa | Sensi. | Speci. | Preci. | Recall | F1 |
|---|---|---|---|---|---|---|---|---|---|---|
| Gini index | 11 | Las Vegas | 1.00 | 0.68 | 0.39 | 0.53 | 0.79 | 0.65 | 0.53 | 0.56 |
| | 9 | SelectKBest | 0.51 | 0.67 | 0.36 | 0.47 | 0.78 | | 0.47 | |
| | 8 | HillClimbing | 1.00 | 0.62 | 0.21 | 0.41 | 0.73 | | 0.41 | |
| | 9 | sequentialBackward | 1.00 | 0.67 | 0.36 | 0.47 | 0.78 | | 0.47 | |
| | 9 | sequential Floating Forward | 1.00 | 0.67 | 0.36 | 0.47 | 0.78 | | 0.47 | |
| | 11 | Genetic Algorithm | 1.00 | 0.62 | 0.21 | 0.41 | 0.73 | | 0.41 | |
| | 20 | AntColony | 1.00 | 0.78 | 0.60 | 0.83 | 0.86 | 0.81 | 0.83 | 0.82 |
| Mutual information | 12 | Las Vegas | 1.27 | 0.72 | 0.46 | 0.56 | 0.82 | 0.69 | 0.56 | 0.58 |
| | 9 | SelectKBest | 0.16 | 0.67 | 0.36 | 0.47 | 0.78 | | 0.47 | |
| | 6 | HillClimbing | 1.27 | 0.58 | 0.09 | 0.37 | 0.69 | | 0.37 | |
| | 8 | Sequential Backward | 1.27 | 0.62 | 0.21 | 0.41 | 0.73 | | 0.41 | |
| | 8 | Sequential Floating Forward | 1.27 | 0.62 | 0.21 | 0.41 | 0.73 | 0.41 | | |
| | 4 | GeneticAlgorithm | 1.27 | 0.67 | 0.34 | 0.47 | 0.78 | | 0.47 | |
| | 20 | AntColony | 1.27 | 0.78 | 0.60 | 0.83 | 0.86 | 0.81 | 0.83 | 0.82 |
| Gain ratio | 7 | Las Vegas | 0.10 | 0.59 | 0.15 | 0.39 | 0.71 | | 0.39 | |
| | 9 | SelectKBest | 0.13 | 0.67 | 0.36 | 0.47 | 0.78 | | 0.47 | |
| | 7 | HillClimbing | 0.10 | 0.59 | 0.15 | 0.39 | 0.71 | | 0.39 | |
| | 11 | SequentialBackward | 0.10 | 0.68 | 0.39 | 0.53 | 0.79 | 0.65 | 0.53 | 0.56 |
| | 11 | Sequential Floating Forward | 0.10 | 0.68 | 0.39 | 0.53 | 0.79 | 0.65 | 0.53 | 0.56 |
| | 1 | GeneticAlgorithm | 0.10 | 0.59 | 0.15 | 0.39 | 0.71 | | 0.39 | |
| | 19 | AntColony | 0.10 | 0.72 | 0.48 | 0.60 | 0.82 | 0.71 | 0.60 | 0.62 |

## Appendix D

This section presents the results of the nonparametric Friedman and Wilcoxon tests performed. For this purpose, the value of the AUC metric was used. The calculation was performed using the R statistical program. Table A4 presents the values obtained from the calculation of the Friedman test. Table A5 presents the matrix of the Wilcoxon test results, both the Z-value on the left and the *p*-value on the right. Table A6 is the matrix used for the calculation of the tests.

**Table A4.** Average Rankings of the algorithms.

| Algorithm | Ranking |
|---|---|
| XGBoost | 0.9999999999999998 |
| RandomForest | 2.2857142857142856 |
| Gbm | 2.714285714285714 |
| Bagging | 3.999999999999999 |
| C4.5 | 5.999999999999999 |
| NaiveBayes | 5.428571428571429 |
| SvmRadial | 7.428571428571429 |
| SvmPoly | 7.571428571428571 |
| MLP | 8.571428571428571 |

Friedman statistic considering reduction performance (distributed according to chi-square with 8 degrees of freedom: 52.3047619047619 *p*-value computed by Friedman test: $1.474479383034577 \times 10^{-8}$.

**Table A5.** Z Score and significance on Wilcoxon test (Z/*p*-value, within table).

| Algorithms | XGBoost | RF [c] | Gbm | Bagging | C4.5 | NaiveBayes | SvmRadial | SvmPoly |
|---|---|---|---|---|---|---|---|---|
| RF | −2.366 [a]/0.018 | - | | | | | | |
| Gbm | −2.366 [a]/0.018 | −1.859 [a]/0.063 * | - | | | | | |
| Bagging | −2.371 [a]/0.018 | −2.366 [a]/0.018 | −2.366 [a]/0.018 | - | | | | |
| C4.5 | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.366 [a]/0.018 | - | | | |
| NaiveBayes | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −0.507 [b]/0.612 * | - | | |
| SvmRadial | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.197 [a]/0.028 | −2.366 [a]/0.018 | - | |
| SvmPoly | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.197 [a]/0.028 | −2.366 [a]/0.018 | −0.845 [a]/0.398 * | - |
| MLP | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.366 [a]/0.018 | −2.028 [a]/0.043 | −2.366 [a]/0.018 | −1.690 [a]/0.091 * | −1.521 [a]/0.128 * |

[a] Based on positive rankings. [b] Based on negative rankings. [c] Random Forest. * Reject the null hypothesis.

**Table A6.** AUC value metrics with different classifiers and dataset.

| DataSet | Algorithms | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | XGBoost | RF | Gbm | Bagging | C45 | NaiveBayes | SvmRadial | SvmPoly | MLP |
| RawData | 0.8997 | 0.8978 | 0.8930 | 0.8781 | 0.8308 | 0.8168 | 0.7973 | 0.7754 | 0.7621 |
| EasyEnsemble | 0.8775 | 0.8744 | 0.8606 | 0.8591 | 0.8249 | 0.8018 | 0.7676 | 0.7679 | 0.7446 |
| SMOTE | 0.8743 | 0.8671 | 0.8575 | 0.8546 | 0.7936 | 0.8040 | 0.7649 | 0.7640 | 0.7512 |
| KmeansSMOTE | 0.8702 | 0.8600 | 0.8515 | 0.8467 | 0.7979 | 0.7970 | 0.7591 | 0.7616 | 0.7424 |
| SMOTETomek | 0.8665 | 0.8562 | 0.8577 | 0.8468 | 0.7623 | 0.7959 | 0.7633 | 0.7634 | 0.7592 |
| SMOTEENN | 0.8335 | 0.8192 | 0.8230 | 0.8165 | 0.7548 | 0.7764 | 0.7511 | 0.7490 | 0.7650 |
| RUS | 0.8173 | 0.8153 | 0.8102 | 0.7962 | 0.7764 | 0.7841 | 0.7360 | 0.7348 | 0.7304 |

## References


1. Realinho, V.; Machado, J.; Baptista, L.; Martins, M.V. Predicting Student Dropout and Academic Success. *Data* **2022**, *7*, 146. [CrossRef]
2. Ortiz-Lozano, J.M.; Rua-Vieites, A.; Bilbao-Calabuig, P.; Casadesús-Fa, M. University student retention: Best time and data to identify undergraduate students at risk of dropout. *Innov. Educ. Teach. Int.* **2018**, *57*, 74–85. [CrossRef]
3. Urbina-Nájera, A.B.; Téllez-Velázquez, A.; Barbosa, R.C. Patterns to Identify Dropout University Students with Educational Data Mining. *Rev. Electron. De Investig. Educ.* **2021**, *23*, e1507. [CrossRef]
4. Lopes Filho JA, B.; Silveira, I.F. Early detection of students at dropout risk using administrative data and machine learning. *RISTI—Rev. Iber. De Sist. E Tecnol. De Inf.* **2021**, *40*, 480–495.
5. Guanin-Fajardo, J.H.; Barranquero, J.C. Contexto universitario, profesores y estudiantes: Vínculos y éxito académico. *Rev. Iberoam. De Educ.* **2022**, *88*, 127–146. [CrossRef]
6. Zeineddine, H.; Braendle, U.; Farah, A. Enhancing prediction of student success: Automated machine learning approach. *Comput. Electr. Eng.* **2020**, *89*, 106903. [CrossRef]
7. Guerrero-Higueras, M.; Llamas, C.F.; González, L.S.; Fernández, A.G.; Costales, G.E.; González, M.C. Academic Success Assessment through Version Control Systems. *Appl. Sci.* **2020**, *10*, 1492. [CrossRef]
8. Rafik, M. Artificial Intelligence and the Changing Roles in the Field of Higher Education and Scientific Research. In *Artificial Intelligence in Higher Education and Scientific Research. Bridging Human and Machine: Future Education with Intelligence*; Springer: Singapore, 2023; pp. 35–46. [CrossRef]
9. BOE. BOE-A-2023-7500 Ley Orgánica 2/2023, de 22 de marzo, del Sistema Universitario. 2023. Available online: https://www.boe.es/buscar/act.php?id=BOE-A-2023-7500 (accessed on 23 March 2024).
10. Guney, Y. Exogenous and endogenous factors influencing students' performance in undergraduate accounting modules. *Account. Educ.* **2009**, *18*, 51–73. [CrossRef]
11. Tamada, M.M.; Giusti, R.; Netto, J.F.d.M. Predicting Students at Risk of Dropout in Technical Course Using LMS Logs. *Electronics* **2022**, *11*, 468. [CrossRef]
12. Contini, D.; Cugnata, F.; Scagni, A. Social selection in higher education. Enrolment, dropout and timely degree attainment in Italy. *High. Educ.* **2017**, *75*, 785–808. [CrossRef]
13. Costa, E.B.; Fonseca, B.; Santana, M.A.; De Araújo, F.F.; Rego, J. Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Comput. Hum. Behav.* **2017**, *73*, 247–256. [CrossRef]
14. Márquez-Vera, C.; Cano, A.; Romero, C.; Noaman, A.Y.M.; Fardoun, H.M.; Ventura, S. Early dropout prediction using data mining: A case study with high school students. *Expert Syst.* **2015**, *33*, 107–124. [CrossRef]
15. Fernández, A.; del Río, S.; Chawla, N.V.; Herrera, F. An insight into imbalanced Big Data classification: Outcomes and challenges. *Complex Intell. Syst.* **2017**, *3*, 105–120. [CrossRef]
16. Rodríguez-Hernández, C.F.; Musso, M.; Kyndt, E.; Cascallar, E. Artificial neural networks in academic performance prediction: Systematic implementation and predictor evaluation. *Comput. Educ. Artif. Intell.* **2021**, *2*, 100018. [CrossRef]

17. Contreras, L.E.; Fuentes, H.J.; Rodríguez, J.I. Academic performance prediction by machine learning as a success/failure indicator for engineering students. *Form. Univ.* **2020**, *13*, 233–246. [CrossRef]
18. Hassan, H.; Anuar, S.; Ahmad, N.B.; Selamat, A. Improve student performance prediction using ensemble model for higher education. In *Frontiers in Artificial Intelligence and Applications*; IOS Press: Amsterdam, The Netherlands, 2019; Volume 318, pp. 217–230.
19. Bolón-Canedo, V.; Alonso-Betanzos, A. Ensembles for feature selection: A review and future trends. *Inf. Fusion* **2018**, *52*, 1–12. [CrossRef]
20. Meghji, A.F.; Mahoto, N.A.; Unar, M.A.; Shaikh, M.A. The role of knowledge management and data mining in improving educational practices and the learning infrastructure. *Mehran Univ. Res. J. Eng. Technol.* **2020**, *39*, 310–323. [CrossRef]
21. Crivei, L.; Czibula, G.; Ciubotariu, G.; Dindelegan, M. Unsupervised learning based mining of academic data sets for students' performance analysis. In Proceedings of the SACI 2020—IEEE 14th International Symposium on Applied Computational In-telligence and Informatics, Proceedings, Timisoara, Romania, 21–23 May 2020; Volume 17, pp. 11–16.
22. Guanin-Fajardo, J.; Casillas, J.; Chiriboga-Casanova, W. Semisupervised learning to discover the average scale of graduation of university students. *Rev. Conrado* **2019**, *15*, 291–299.
23. Alyahyan, E.; Düşteargör, D. Decision trees for very early prediction of student's achievement. In Proceedings of the 2020 2nd International Conference on Computer and Information Sciences (ICCIS), Sakaka, Saudi Arabia, 3–15 October 2020; pp. 1–7.
24. Alwarthan, S.; Aslam, N.; Khan, I.U. An Explainable Model for Identifying At-Risk Student at Higher Education. *IEEE Access* **2022**, *10*, 107649–107668. [CrossRef]
25. Adekitan, A.I.; Noma-Osaghae, E. Data mining approach to predicting the performance of first year student in a university using the admission requirements. *Educ. Inf. Technol.* **2018**, *24*, 1527–1543. [CrossRef]
26. Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P. Knowledge Discovery and Data Mining: Towards a Unifying Framework. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, 2–4 August 1996; pp. 82–88.
27. Haixiang, G.; Yijing, L.; Shang, J.; Mingyun, G.; Yuanyue, H.; Bing, G. Learning from class-imbalanced data: Review of methods and applications. *Expert Syst. Appl.* **2017**, *73*, 220–239. [CrossRef]
28. Chawla, N.; Bowyer, K. SMOTE: Synthetic Minority Over-sampling Technique Nitesh. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
29. Bertolini, R.; Finch, S.J.; Nehm, R.H. Enhancing data pipelines for forecasting student performance: Integrating feature selection with crossvalidation. *Int. J. Educ. Technol. High. Educ.* **2021**, *18*, 44. [CrossRef] [PubMed]
30. Febro, J.D. Utilizing Feature Selection in Identifying Predicting Factors of Student Retention. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 269–274. [CrossRef]
31. Ghaemi, M.; Feizi-Derakhshi, M.-R. Feature selection using Forest Optimization Algorithm. *Pattern Recognit.* **2016**, *60*, 121–129. [CrossRef]
32. R Development Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2020.
33. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
34. Alturki, S.; Alturki, N.; Stuckenschmidt, H. Using Educational Data Mining to Predict Students' Academic Performance for Applying Early Interventions. *J. Inf. Technol. Educ. JITE. Innov. Pract. IIP* **2021**, *20*, 121–137. [CrossRef]
35. Fernández-García, A.J.; Rodríguez-Echeverría, R.; Preciado, J.C.; Manzano, J.M.C.; Sánchez-Figueroa, F. Creating a recommender system to support higher education students in the subject enrollment decisión. *IEEE Access* **2020**, *8*, 189069–189088. [CrossRef]
36. Helal, S.; Li, J.; Liu, L.; Ebrahimie, E.; Dawson, S.; Murray, D.J.; Long, Q. Predicting academic performance by considering student heterogeneity. *Knowl.-Based Syst.* **2018**, *161*, 134–146. [CrossRef]
37. Yağci, M. Educational data mining: Prediction of students' academic performance using machine learning algorithms. *Smart Learn. Environ.* **2022**, *9*, 11. [CrossRef]
38. Gil, P.D.; Martins, S.d.C.; Moro, S.; Costa, J.M. A data-driven approach to predict first-year students' academic success in higher education institutions. *Educ. Inf. Technol.* **2020**, *26*, 2165–2190. [CrossRef]
39. Beaulac, C.; Rosenthal, J.S. Predicting University Students' Academic Success and Major Using Random Forests. *Res. High. Educ.* **2019**, *60*, 1048–1064. [CrossRef]
40. Fernandes, E.R.; de Carvalho, A.C. Evolutionary inversion of class distribution in overlapping areas for multiclass imbalanced learning. *Inf. Sci.* **2019**, *494*, 141–154. [CrossRef]
41. Morales, P.; Luengo, J.; García, L.P.F.; Lorena, A.C.; de Carvalho, A.C.P.L.F.; Herrera, F.; Ciencias, I.D.; Paulo, U.D.S.; Av, T.S.-C.; Carlos, S.; et al. Noisefiltersr the noise-filtersr package. *R J.* **2017**, *9*, 219–228. [CrossRef]
42. Zeng, X.; Martinez, T. A noise filtering method using neural networks. In Proceedings of the IEEE International Workshop on Soft Computing Techniques in Instrumentation and Measurement and Related Applications (SCIMA2003), Provo, UT, USA, 17 May 2003; pp. 26–31.
43. Verbaeten, S.; Assche, A. Ensemble methods for noise elimination in classification problems. In *Multiple Classifier Systems. MCS 2003*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2003; pp. 317–325.
44. Ali, A.; Jayaraman, R.; Azar, E.; Maalouf, M. A comparative analysis of machine learning and statistical methods for evaluating building performance: A systematic review and future benchmarking framework. *J. Affect. Disord.* **2024**, *252*, 111268. [CrossRef]

45. Rajula, H.S.R.; Verlato, G.; Manchia, M.; Antonucci, N.; Fanos, V. Comparison of Conventional Statistical Methods with Machine Learning in Medicine: Diagnosis, Drug Development, and Treatment. *Medicina* **2020**, *56*, 455. [CrossRef] [PubMed]
46. García, S.; Luengo, J.; Herrera, F. Tutorial on practical tips of the most influential data preprocessing algo-rithms in data mining. *Knowl.-Based Syst.* **2016**, *98*, 1–29. [CrossRef]
47. Cruz RM, O.; Sabourin, R.; Cavalcanti GD, C. Dynamic classifier selection: Recent advances and perspectives. *Inf. Fusion* **2018**, *41*, 195–216. [CrossRef]
48. Yadav, S.K.; Pal, S. Data Mining: A Prediction for Performance Improvement of Engineering Students using Classification. *arXiv* **2012**, arXiv:1203.3832. [CrossRef]
49. Bradley, A.P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* **1997**, *30*, 1145–1159. [CrossRef]
50. Nájera, A.B.U.; de la Calleja, J.; Medina, M.A. Associating students and teachers for tutoring in higher education using clustering and data mining. *Comput. Appl. Eng. Educ.* **2017**, *25*, 823–832. [CrossRef]
51. Kononenko, I. Estimating Attributes: Analysis and Extensions of RELIEF. In *European Conference on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 1994; pp. 171–182.
52. Liu, H.; Setiono, R. Feature selection and classification: A probabilistic wrapper approach. In Proceedings of the 9th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEAAIE´96), Fukuoka, Japan, 4–7 June 1996; pp. 419–424.
53. Zhu, Z.; Ong, Y.-S.; Dash, M. Wrapper–Filter Feature Selection Algorithm Using a Memetic Framework. *IEEE Trans. Syst. Man Cybern. Part B* **2007**, *37*, 70–76. [CrossRef]
54. Liu, H.; Yu, L. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 491–502. [CrossRef]
55. Batista, G.E.A.P.A.; Monard, M.C. An analysis of four missing data treatment methods for supervised learning. *Appl. Artif. Intell.* **2003**, *17*, 519–533. [CrossRef]
56. Kira, K.; Rendell, L. The feature selection problem: Traditional methods and a new algorithm. In Proceedings of the AAAI'92: Proceedings of the Tenth National Conference on Artificial Intelligence, San Jose, CA, USA, 12–16 July 1992; pp. 129–134.
57. Qian, W.; Shu, W. Mutual information criterion for feature selection from incomplete data. *Neurocomputing* **2015**, *168*, 210–220. [CrossRef]
58. Sheinvald, J.; Dom, B.; Niblack, W. A modeling approach to feature selection. In Proceedings of the 10th International Conference on Pattern Recognition, Atlantic City, NJ, USA, 16–21 June 1990; Volume I, pp. 535–539.
59. Coefficient of Determination. In *The Concise Encyclopedia of Statistics*; Springer: New York, NY, USA, 2008; pp. 88–91. [CrossRef]
60. Quinlan, J. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
61. Ceriani, L.; Verme, P. The origins of the Gini index: Extracts from Variabilità e Mutabilità (1912) by Corrado Gini. *J. Econ. Inequal.* **2011**, *10*, 421–443. [CrossRef]
62. Pawlak, Z. *Imprecise Categories, Approximations and Rough Sets*; Springer: Dordrecht, The Netherlands, 1991; Volume 19, pp. 9–32.
63. Wang, D.; Zhang, Z.; Bai, R.; Mao, Y. A hybrid system with filter approach and multiple population genetic algorithm for feature selection in credit scoring. *J. Comput. Appl. Math.* **2018**, *329*, 307–321. [CrossRef]
64. Douzas, G.; Bacao, F.; Last, F. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Inf. Sci.* **2018**, *465*, 1–20. [CrossRef]
65. Batista, G.E.; Bazzan, A.L.; Monard, M.C. Balancing training data for automated annotation of keywords: A case study. *WOB* **2003**, *3*, 10–18.
66. Ivan, T. Two modifications of cnn. *IEEE Trans. Syst. Man Commun. SMC* **1976**, *6*, 769–772.
67. Liu, X.-Y.; Wu, J.; Zhou, Z.-H. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Trans. Syst. Man Cybern. Part B* **2008**, *39*, 539–550. [CrossRef]
68. Hearst, M.A. Support vector machines. *IEEE Intell. Syst.* **1998**, *13*, 18–28. [CrossRef]
69. Almeida, L.B. C1. 2 multilayer perceptrons. In *Handbook of Neural Computation*; Oxford University Press: New York, NY, USA, 1997; pp. 1–30.
70. Breiman, L. Random forests. *Ensemble Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
71. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
72. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
73. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
74. Webb, G.I. Naïve Bayes. *Encycl. Mach. Learn.* **2010**, *15*, 713–714.
75. Shetu, S.F.; Saifuzzaman, M.; Moon, N.N.; Sultana, S.; Yousuf, R. Student's performance prediction using data mining technique depending on overall academic status and environmental attributes. In *Advances in Intelligent Systems and Computing*; Springer: Berlin/Heidelberg, Germany, 2021; Volume 1166, pp. 757–769.
76. Fisher, R.A. *The Design of Experiments*; Oliver & Boyd: Thomas Oliver, NY, USA, 1935.
77. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30. Available online: http://jmlr.org/papers/v7/demsar06a.html (accessed on 9 April 2024).
78. Cohen, J. The eart is round ($p < 0.05$). *Am. Psychol.* **1994**, *49*, 997–1003. [CrossRef]

79. Schmidt, F.L. Statistical significance testing and cumulative knowledge in psychology: Implications for training of researchers. *Psychol. Methods* **1996**, *1*, 115–129. [CrossRef]
80. Harlow, L.L.; Mulaik, S.A.; Steiger, J.H. (Eds.) Multivariate Applications Book Series. In *What If There Were No Significance Tests?* Lawrence Erlbaum Associates Publishers: Mahwah, NJ, USA, 1997.
81. Al-Fairouz, E.I.; Al-Hagery, M.A. Students Performance: From Detection of Failures and Anomaly Cases to the Solutions-Based Mining Algorithms. *Int. J. Eng. Res. Technol.* **2020**, *13*, 2895–2908. [CrossRef]
82. Ismanto, E.; Ghani, H.A.; Saleh, N.I.B.M. A comparative study of machine learning algorithms for virtual learning environment performance prediction. *IAES Int. J. Artif. Intell.* **2023**, *12*, 1677–1686. [CrossRef]
83. Kaushik, Y.; Dixit, M.; Sharma, N.; Garg, M. Feature Selection Using Ensemble Techniques. In *Futuristic Trends in Network and Communication Technologies*; FTNCT 2020. Communications in Computer and Information Science; Springer: Singapore, 2021; Volume 1395, pp. 288–298. [CrossRef]
84. Mayer, A.-K.; Krampen, G. Information literacy as a key to academic success: Results from a longitudinal study. *Commun. Comput. Inf. Sci.* **2016**, *676*, 598–607. [CrossRef]
85. Harackiewicz, J.M.; Barron, K.E.; Tauer, J.M.; Elliot, A.J. Predicting success in college: A longitudinal study of achievement goals and ability measures as predictors of interest and performance from freshman year through graduation. *J. Educ. Psychol.* **2002**, *94*, 562–575. [CrossRef]
86. Meier, Y.; Xu, J.; Atan, O.; van der Schaar, M. Predicting Grades. *IEEE Trans. Signal Process.* **2015**, *64*, 959–972. [CrossRef]
87. Lord, S.M.; Ohland, M.W.; Orr, M.K.; Layton, R.A.; Long, R.A.; Brawner, C.E.; Ebrahiminejad, H.; Martin, B.A.; Ricco, G.D.; Zahedi, L. MIDFIELD: A Resource for Longitudinal Student Record Research. *IEEE Trans. Educ.* **2022**, *65*, 245–256. [CrossRef]
88. Tompsett, J.; Knoester, C. Family socioeconomic status and college attendance: A consideration of individual-level and school-level pathways. *PLoS ONE* **2023**, *18*, e0284188. [CrossRef]
89. Ma, Y.; Cui, C.; Nie, X.; Yang, G.; Shaheed, K.; Yin, Y. Pre-course student performance prediction with multi-instance multi-label learning. *Sci. China Inf. Sci.* **2018**, *62*, 29101. [CrossRef]
90. Berrar, D. Confidence curves: An alternative to null hypothesis significance testing for the comparison of classifiers. *Mach. Learn.* **2017**, *106*, 911–949. [CrossRef]
91. Berrar, D.; Lozano, J.A. Significance tests or confidence intervals: Which are preferable for the comparison of classifiers? *J. Exp. Theor. Artif. Intell.* **2013**, *25*, 189–206. [CrossRef]
92. García, S.; Herrera, F. An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. *J. Mach. Learn. Res.* **2008**, *9*, 2677–2694.
93. Biju, V.G.; Prashanth, C. Friedman and Wilcoxon Evaluations Comparing SVM, Bagging, Boosting, K-NN and Decision Tree Classifiers. *J. Appl. Comput. Sci. Methods* **2017**, *9*, 23–47. [CrossRef]