A Proposed Framework for Evaluating the Academic-failure Prediction in Distance Learning

Patrícia Takaki1,2 · Moisés Lima Dutra1 · Gustavo de Araújo1 · Eugênio Monteiro da Silva Júnior1

Accepted: 15 November 2021 / Published online: 12 April 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Academic failure is a crucial problem that affects not only students but also institutions and countries. Lack of success in the educational process can lead to health and social disorders and economic losses. Consequently, predicting in advance the occurrence of this event is a good prevention and mitigation strategy. This work proposes a framework to evaluate machine learning-based predictive models of academic failure, to facilitate early pedagogical interventions. We took a Brazilian undergraduate course in the distance learning modality as a case study. We run seven classification models on normalized datasets, which comprised grades for three weeks of classes for a total of six weeks. Since it is an imbalanced-data context, adopting a single metric to identify the best predictive model of student failure would not be efficient. Therefore, the proposed framework considers 11 metrics generated by the classifiers run and the application of exclusion and ordering criteria to produce a list of best predictors. Finally, we discussed and presented some possible applications for minimizing the students’ failure.

Keywords Machine learning · Educational data mining · Failure prediction · Distance learning

1 Introduction
Student failure in undergraduate courses is an old issue for which no definitive solution has yet been found. This event occurs in different ways and depends on several factors; it is an educational issue involving different actors [2]. The failure of a single student can result in this person’s demotivation, loss of his/her expected learning flow, waste of financial resources, negative impacts on the institution’s indicators, decreased funding granted to the institution, besides strongly contributing to increasing the course and institution dropout rate. According to [4, 18], dropout in higher education represents a problem for all nations, causing social, scientific, and economic losses. Sometimes these events represent public resources that do not provide an effective return on investment; other times, they are significant losses of revenues in the private sector. When students abandon their opportunity for training and intellectual growth, they are also excluded from an entire educational process that prepares them for the job market and citizenship.

Educational Data Mining (EDM) represents a multidisciplinary research field dedicated to developing methods to explore data from educational environments [3, 16]. It is possible to understand students more effectively and appropriately through an EDM process, i.e., how they learn, the importance of the context in which the learning occurs, and other factors influencing the learning process [6]. For [14], the main idea behind EDM is developing and using methods to analyze and interpret the ‘big data’ from computer-based learning systems and school, college, or university administration and management systems.

This work applied prediction models to identify undergraduate students at risk of subject failing, with about 50% of the total course time yet to elapse. As a case study to evaluate this proposal, we analyzed the students’ performance in an undergraduate course offered at a distance by a Brazilian public university part of the Open University of Brazil (UAB) system. UAB offers distance learning courses intermittently. That means that there are no annual or semester offers like for traditional courses. Therefore, if a student fails a course, he/she may not have the opportunity to enroll to retake it in the future, as there is no
guarantee that there will be a new offer of this subject until
the end of his/her undergraduate course. Students who fail
in a subject in this context, even if there are pedagogical
strategies to recover their learning and passing, will have
a greater chance of dropping out of their courses. Such a
situation can negatively impact their training, professional
future, institutional promotion, and even the social and
economic development of the location where these stu-
dents are inserted, besides other potential losses. At best,
they will choose to continue their course at another higher
educational institution.

Based on this application scenario, this work sought to
provide those involved in the teaching-learning process at
UAB (students, teachers, and managers) with a predictive
model composed of information about individual risks of
academic failure in specific subjects. Managers (directors,
coordinators of both courses and remote centers), teachers
(in-person and distance tutors), and students can benefit
from qualified information to support the teaching-learning
process. We believe that this initiative has the potential to
minimize this type of event. With the support of subject-fail-
ure predictive-analysis-based alerts, they can act to mitigate
the risk of academic failures.

This work carried out experimental tests with different
classification models to evaluate the best results for predict-
ing students’ failure in a given distance higher education
course. For this purpose, we analyzed and compared eleven
evaluation metrics captured from each of the seven classification
models run. Specifically, we intended:

- To collect and wrangle student-grade reports available in
  the UAB’s Moodle Virtual Learning Environment;
- To perform data pre-processing, including anonymiza-
tions, data insertions and deletions, data normalization,
conversion of categorical data into numeric data, and
splitting training and test datasets;
- To predict the academic failure by applying seven dif-
f erent predictive models, by training the data with and
without balancing options and with a dataset consisting
of three weeks of class, for a total of six weeks;
- To gather and compare eleven different evaluation met-
rics for each model: true positive, true negative, false
positive, false negative, accuracy, precision, recall, speci-
ficity, kappa index, geometric mean (g-means), and har-
monic measure (f-measure);
- To analyze and discuss the results obtained.
- To propose a framework to assess the academic-failure
prediction in distance education.

The remainder of this paper is organized as follows. Sec-
tion 2 describes the materials and methods used. Section 3
presents and discusses the results obtained. Section 4 pre-
sents the proposed framework. Finally, in Section 5, we draw
some conclusions and give the proposal’s limitations and
suggestions for future works.

## 2 Materials and methods

This is an exploratory work organized in three stages,
based on [11]: pre-processing, machine learning, and
post-processing.

1. During pre-processing, we collected, cleansed, and pre-
pared the data gathered in the Moodle environment. This
data corresponds to the students’ grades in 21 subjects
taken by approximately 250 students enrolled in the ana-
lyzed course.

2. In the machine learning stage, we run four classifica-
tion algorithms written in Python and evaluated them:
Logistic Regression, Decision Tree, Random Forest, and
Support Vector Machine. We used the RBF kernel and
polynomial kernel of degrees 2, 3, and 4. We tested the
algorithms with and without data balancing. During the
unbalanced processing, we used the SMOTE and Near
Miss methods.

3. In the post-processing stage, we carried out different
analyses, isolated or combined, of the evaluation metrics
obtained by the classification models previously tested:
true positive, true negative, false positive, false nega-
tive, accuracy, precision, recall, specificity, g-means,
f-measure, and kappa index. Subsequently, we proposed,
implemented and evaluated a framework to identify the
best academic-failure prediction models.

As supporting tools, in addition to the Moodle Learning
Management System, we worked with the WEKA soft-
ware, the GitHub platform, and the Google Collaboratory
environment.

## 3 Related works

EDM techniques to predict students’ academic failures have
been used in several studies reported in the literature [1–3,
5, 7, 8, 10, 12, 13, 15]. Each of them uses different data,
collected at different times in the course offering, applies
different predictive models and compares results with dif-
ferent metrics. Some of these works are worth mentioning.

The work by [5] presents a methodology to automatically
detect students “at-risk” of failing a module of computer
programming courses and support adaptive feedback simul-
taneously. They used six classifiers: K-Neighbors, Decision
tree, Random forest, Logistic regression, Linear SVM, and
Gaussian SVM, without data balancing techniques. The
K-Neighbors (k = 12) was the classifier selected, which
showed high performance for F1, precision, and recall metrics compared with other classifiers. A comparative study on the effectiveness of four EDM techniques (Decision Tree, Support Vector Machine, Neural Network, and Naïve Bayes) for early identification of students likely to fail in introductory programming courses is presented by [8]. They used two data sources from a Brazilian university: one from distance education and the other from face-to-face learning. They performed data pre-processing and fine-tuning tasks, adopting only the f-measure evaluation. The fine-tuned SVM techniques identified at least 92% and 83% of effectiveness the students are likely to fail when they have performed at least 50% and 25% of the distance education and on-campus courses, respectively. The work by [13] proposed a learning analytics approach using data mining and machine learning to predict the grades of four primary assignments in a Hellenic Open University’s annual module. They used Moodle data, ran the classifiers Random Forest, Linear Regression, Neural Network, AdaBoost, SVM, and k-NN, and measured the Mean Absolute Error (MAE) to identify the accuracy of classifiers’ predictions. The Random Forest algorithm provided the best results for the four assignments. Our literature review could not identify studies that used the same methodology we proposed, i.e., that possessed a preliminary stage for evaluating and selecting the best predictors. Consequently, to the best of our knowledge, this fact supports the originality of our work.

4 Results and discussion

4.1 Step 1: Data pre-processing

Data collection and pre-processing produced 4,396 instances of performance data for 250 UAB distance students. We collected student grades for 21 subjects, all with the same amount and score of evaluative activities. The data collected corresponds to students’ grades in subjects completed by March 2020. At that point, the 4th semester of classes was in progress, and the UAB’s methodology for evaluating its students was being modified due to the unexpected coronavirus pandemic outbreak. The selected course possesses the largest number of students among all courses of the distance learning modality at the analyzed university. It is offered in five on-site remote centers (cities) in the Brazilian state of Minas Gerais and provides subjects of 30h, 60h, 90h, and 120h. For this case study, we chose to work with the 60h subjects because they represent 84% (21 out of 25) of the subjects completed by the time of the data collection.

There are 15 points for each of the two Collaborative Activities (CA1 and CA2, whose points are distributed in 7 points for the presentation in class-based meetings, and 8 points for the work delivery in the virtual room); 3 points for each of the four Discussion Forums (DF1, DF2, DF3, and DF4); 9 points for each of the two Individual Activities (IA1 and IA2); and 40 points for the face-to-face assessment (FA), on the last day of class. The 60h courses have a duration of six weeks. We gathered 105 files of grade reports (electronic spreadsheets) from the virtual rooms of Moodle used by the chosen course. We performed several initial data cleaning transformations on the spreadsheets coming from the standard Moodle “Grade Report” to consolidate them in a single format for all spreadsheets. With the spreadsheets assembled, we selected the columns for grades obtained in CA1 and CA2 (split into presentations and deliveries), grades received in DF1 and DF2, and grades received in IA1. Next, we inserted two columns: Status (“Passing/Failing”), based on the final grade, and Remote Center (city number).

The grades from individual activity 2, discussion forums 3 and 4, and the face-to-face assessment have been removed because they occur in the last three weeks of class. That is, grades IA2 and FA are only known at the end of the course. It is important to point out that the objective of this work is to predict subject failure halfway through. The data were anonymized, the numerical values were converted to the American standard (decimal points), and the data file was converted to the CSV format. In WEKA’s Preprocess tab, the numerical data of the grades (from 3 to 9 points) were normalized by using the filter called Normalize, as explained by [19]. The resulting spreadsheet totaled 4396 data instances. It was possible to verify that 3736 students (85%) passed the subjects and 660 students (15%) failed. This dataset was divided so that the scores of the first 16 completed subjects were used to train the models (3348 instances, 76%) and the last five subjects for tests (1048 instances, 24%). An initial exploratory analysis of the collected data is presented in Table 1.

The division between the training (76%) and testing (24%) datasets1 is consistent with the recommendations and practices found in the literature [17]. The data are imbalanced between the Passing (85%) and Failing (15%) categories, which suggests that the evaluation of predictive models

| Table 1 Data Overview: division between training/test and passing/failing |
|-----------------------------|------------------|------------------|
| DATASET                  | Amount | Passing | Failing |
| Training                  | 3348 (76%) | 2814 (84%) | 534 (16%) |
| Test                      | 1048 (24%) | 922 (88%)  | 126 (12%)  |
| TOTAL                     | 4396 (100%) | 3736 (85%) | 660 (15%) |

1 The datasets generated and analysed during the current study are available in the GitHub repository, https://github.com/ptakaki/ML.
should consider using strategies to deal with this imbalanced data and verify its effectiveness.

4.2 Step 2: Machine learning

To develop the predictive models, we used the following Python libraries: numpy, math, matplotlib, pandas, seaborn, scikit-learn, and imblearn. The imblearn library provides the SMOTE (oversampling) and Near Miss (undersampling) APIs for data balancing. After creating the training and testing dataframes, we converted the categorical variables Remote Center (RC) and Status into numeric variables. Next, we separated the dependent and independent variables and created the X_train, y_train, X_test and y_test dataframes. A training data correlation matrix for viewing selected features is shown in Fig. 1.

As expected, the highest correlations between student features and the predictive variable Status_Failing refer to students’ performances in different activities performed up to the 3rd week of class. Individual Activity 1 (IA1) presented the highest correlation with the predictive variable, while remote educational centers had the lowest correlations, slightly different for the City4 remote center. It is important to emphasize that these and other variable analyses are limited to correlations, and under no circumstances can they be interpreted as causes of the event being studied.

Subsequently, we compared the values obtained after running the classification algorithms with the test dataset’s known outputs.

Due to the unbalance of the data in the predictive class, we compared the results of the classifiers with the training data with and without balancing options. The balanced options were executed via oversampling (SMOTE method, imblearn over_sampling API) and undersampling (Near Miss method, imblearn under_sampling API). While the SMOTE method creates synthetic data in the minority class to have the number of instances of the majority class, the Near Miss method removes data from the majority class so that it has the number of instances of the minority class. Thus, we carried out 21 prediction tests and collected 11 evaluation metrics for each classifier run. Table 2 shows the analyzed metrics.

Figure 2 presents the confusion matrices for the predictive models tested, and Fig. 3 shows the seven metrics generated from the confusion matrices. When considering each metric in isolation, the best results are highlighted in green and the worst in red.

4.3 Step 3: Discussion

With the set of 11 metrics applied for each predictive model run, the analysis and discussion of these results sought to identify the best results to predict a students’ failure in advance. We aimed to identify a predictive model that makes it possible to generate alerts at the beginning of the 2nd half
Fig. 2 Confusion matrix metrics for predictive models

| Classifier                          | tp    | tn    | fp    | fn    |
|-------------------------------------|-------|-------|-------|-------|
| Decision Tree                       | 80    | 875   | 47    | 46    |
| Logistic Regression                 | 67    | 918   | 4     | 59    |
| Random Forest                       | 80    | 902   | 20    | 46    |
| SVM with a 2nd-deg kernel           | 84    | 904   | 18    | 42    |
| SVM with a 3rd-deg kernel           | 77    | 900   | 22    | 49    |
| SVM with a 4th-deg kernel           | 83    | 870   | 52    | 43    |
| SVM with RBF kernel                 | 76    | 913   | 9     | 50    |
| NearMiss: Decision Tree             | 98    | 549   | 373   | 28    |
| NearMiss: Logistic Regression       | 81    | 827   | 95    | 45    |
| NearMiss: Random Forest             | 106   | 485   | 437   | 20    |
| NearMiss: SVM with a 2nd-deg kernel | 97    | 698   | 224   | 29    |
| NearMiss: SVM with a 3rd-degree     | 99    | 668   | 254   | 27    |
| NearMiss: SVM with a 4th-deg kernel | 99    | 658   | 264   | 27    |
| NearMiss: SVM with RBF kernel       | 107   | 642   | 280   | 19    |
| SMOTE: Decision Tree                | 92    | 546   | 376   | 34    |
| SMOTE: Logistic Regression          | 87    | 892   | 30    | 39    |
| SMOTE: Random Forest                | 101   | 615   | 307   | 25    |
| SMOTE: SVM with a 2nd-deg kernel    | 93    | 875   | 47    | 33    |
| SMOTE: SVM with a 3rd-deg kernel    | 94    | 881   | 41    | 32    |
| SMOTE: SVM with a 4th-deg kernel    | 89    | 840   | 82    | 37    |
| SMOTE: SVM with RBF kernel          | 102   | 875   | 47    | 24    |

Fig. 3 Results of the other 7 metrics of the predictions

| Classifier                          | accu  | prec  | reca  | spec  | g_means | f_measu | kappa   |
|-------------------------------------|-------|-------|-------|-------|---------|---------|---------|
| Decision Tree                       | 0.91126 | 0.62992 | 0.63492 | 0.94902 | 0.77624 | 0.63241 | 0.58195 |
| Logistic Regression                 | 0.93988 | 0.94366 | 0.53175 | 0.99566 | 0.72763 | 0.68020 | 0.64985 |
| Random Forest                       | 0.93702 | 0.80000 | 0.63492 | 0.97831 | 0.78813 | 0.70796 | 0.67319 |
| SVM with a 2nd-deg kernel           | 0.94275 | 0.82353 | 0.66667 | 0.98048 | 0.80849 | 0.73684 | 0.70512 |
| SVM with a 3rd-deg kernel           | 0.93225 | 0.77778 | 0.61111 | 0.97614 | 0.77235 | 0.68444 | 0.64711 |
| SVM with a 4th-deg kernel           | 0.90935 | 0.61481 | 0.65873 | 0.94360 | 0.78840 | 0.63601 | 0.58431 |
| SVM with RBF kernel                 | 0.94370 | 0.89412 | 0.60317 | 0.99024 | 0.77284 | 0.72038 | 0.69039 |
| NearMiss: Decision Tree             | 0.61737 | 0.20807 | 0.77778 | 0.59544 | 0.68053 | 0.32831 | 0.17105 |
| NearMiss: Logistic Regression       | 0.86641 | 0.46023 | 0.64266 | 0.89696 | 0.75935 | 0.53642 | 0.46087 |
| NearMiss: Random Forest             | 0.56393 | 0.19521 | 0.84127 | 0.52603 | 0.66523 | 0.31689 | 0.15124 |
| NearMiss: SVM with a 2nd-deg kernel | 0.75859 | 0.30218 | 0.76984 | 0.75705 | 0.76342 | 0.43400 | 0.31587 |
| NearMiss: SVM with a 3rd-deg kernel | 0.73187 | 0.28045 | 0.78571 | 0.72451 | 0.75449 | 0.41336 | 0.28702 |
| NearMiss: SVM with a 4th-deg kernel | 0.72233 | 0.27273 | 0.78571 | 0.71367 | 0.74882 | 0.40491 | 0.27560 |
| NearMiss: SVM with RBF kernel       | 0.71469 | 0.27648 | 0.84921 | 0.69631 | 0.76897 | 0.47175 | 0.28800 |
| SMOTE: Decision Tree                | 0.60878 | 0.19658 | 0.73016 | 0.59219 | 0.65757 | 0.30976 | 0.14843 |
| SMOTE: Logistic Regression          | 0.93416 | 0.74359 | 0.69048 | 0.96746 | 0.81732 | 0.71605 | 0.67887 |
| SMOTE: Random Forest                | 0.68321 | 0.24755 | 0.80159 | 0.66703 | 0.73122 | 0.37828 | 0.23834 |
| SMOTE: SVM with a 2nd-deg kernel    | 0.92366 | 0.66429 | 0.73809 | 0.94902 | 0.83694 | 0.69925 | 0.65567 |
| SMOTE: SVM with a 3rd-deg kernel    | 0.93034 | 0.69630 | 0.74603 | 0.95553 | 0.84431 | 0.72031 | 0.68058 |
| SMOTE: SVM with a 4th-deg kernel    | 0.88645 | 0.52047 | 0.70635 | 0.91106 | 0.80220 | 0.59333 | 0.53494 |
| SMOTE: SVM with RBF kernel          | 0.93225 | 0.68456 | 0.80952 | 0.94902 | 0.87650 | 0.74182 | 0.70314 |
of each subject so that everyone (students, tutors, teachers, and coordinators) receives information to carry out pedagogical interventions to reverse this possible failure.

### 4.3.1 Accuracy and recall metrics analysis

This work initially sought to consider the mathematical-computational context for comparing the performance of the classifiers, where the **accuracy** metric has a prominent position in identifying the best prediction made. The following formula gives its calculation: 

\[
\frac{tp + tn}{tp + tn + fp + fn}
\]

indicating how many classifications were made correctly. The results ordered in decreasing order of accuracy are shown in Fig. 4(a).

The predictive model with the SVM-RBF-based kernel obtained the best accuracy, 94.37%. Most models that did not use the data balancing methods SMOTE and Near Miss achieved the best results (above 90%), except for Decision Tree, which underperformed other models that used the SMOTE method. However, the isolated analysis of the accuracy metric can limit the context *in situ*, where the data is unbalanced. It is necessary to analyze the classifiers’ metrics combined so that the performance of these models from other and more specific viewpoints is taken into account. For this reason, the present work used other metrics to evaluate the performance of the implemented models.

The **recall** metric is an indispensable tool to indicate how well the model is getting right in predicting the minority class. The following formula gives its calculation: 

\[
\frac{tp}{tp + fn}
\]

showing how many of those who failed were classified as failing. This metric allows knowing if the model is classified as failing the most significant possible number of students that will really fail. The results ordered in decreasing order of recall are shown in Fig. 4(b). Searching for the best results for recall meets the objective of identifying the model that ideally does not classify as passing a student who will fail. This error will cause this student to be out of reach for the scope of actions and initiatives to reverse or minimize such a negative outcome. The NearMiss with SVM-RBF-based kernel model obtained the best recall (84.92%), but this result should also not be used in isolation. When analyzing the accuracy metric, now combined with the recall, it is possible to notice that the model with the best accuracy (SVM-RBF-based kernel, 94.36%) obtained the 2nd worst recall (60.31%). That is, it misses a lot in the minority class (failing). This situation means that 50 out of the 126 failed students were classified as passing, demonstrating the

| Classifier | accu  | Classifier | reca  |
|------------|-------|------------|-------|
| SVM with RBF kernel | 0.94370 | NearMiss: SVM with RBF kernel | 0.84921 |
| SVM with a 2nd-deg kernel | 0.94275 | NearMiss: Random Forest | 0.84127 |
| Logistic Regression | 0.93989 | SMOTE: SVM with RBF kernel | 0.80952 |
| Random Forest | 0.93702 | SMOTE: Random Forest | 0.80159 |
| SMOTE: Logistic Regression | 0.93416 | NearMiss: SVM with a 3rd-deg kernel | 0.78571 |
| SMOTE: SVM with RBF kernel | 0.93225 | NearMiss: SVM with a 4th-deg kernel | 0.78571 |
| SVM with a 3rd-deg kernel | 0.93225 | NearMiss: Decision Tree | 0.77778 |
| SMOTE: SVM with a 3rd-deg kernel | 0.93034 | NearMiss: SVM with a 2nd-deg kernel | 0.76984 |
| SMOTE: SVM with a 2nd-deg kernel | 0.92366 | SMOTE: SVM with a 3rd-deg kernel | 0.74603 |
| Decision Tree | 0.91126 | SMOTE: SVM with a 2nd-deg kernel | 0.73810 |
| SVM with a 4th-deg kernel | 0.90935 | SMOTE: SVM with a 4th-deg kernel | 0.73016 |
| SMOTE: SVM with a 4th-deg kernel | 0.88645 | SMOTE: Logistic Regression | 0.70635 |
| NearMiss: Logistic Regression | 0.86641 | SMOTE: Logistic Regression | 0.69048 |
| NearMiss: SVM with a 2nd-deg kernel | 0.75859 | SVM with a 2nd-deg kernel | 0.66667 |
| NearMiss: SVM with a 3rd-deg kernel | 0.73187 | SVM with a 4th-deg kernel | 0.65873 |
| NearMiss: SVM with a 4th-deg kernel | 0.72233 | NearMiss: Logistic Regression | 0.64286 |
| NearMiss: SVM with a 3rd-deg kernel | 0.71470 | Random Forest | 0.63492 |
| SMOTE: Random Forest | 0.68321 | Decision Tree | 0.63492 |
| NearMiss: Decision Tree | 0.61737 | SVM with a 3rd-deg kernel | 0.61111 |
| SMOTE: Decision Tree | 0.60878 | SVM with RBF kernel | 0.60318 |
| NearMiss: Random Forest | 0.56393 | Logistic Regression | 0.53175 |

*Fig. 4: Accuracy (a) and Recall (b) of classifiers*
significant damage to mitigating the failure if one had only adopted the accuracy metric.

4.3.2 Precision and specificity metrics analysis

**Precision** measures how well the model is getting it right when classifying a student as failing. The following formula gives its calculation: \( \frac{tp}{tp + fp} \), indicating how many of those classified as failing were actually failed. Analyzing the recall metric, now combined with the precision (Fig. 5(a)), we observe that the model with the best recall (NearMiss with SVM-RBF-based kernel) presented a very low precision (27.65%), the 6th worst. In comparison, Logistic Regression obtained 94.37% of precision.

Figure 5(a) shows the results ordered by precision. These results demonstrate that the NearMiss with SVM-RBF-based kernel model obtained the best recall (84.92%) and the 6th worst precision (27.65%). It incorrectly classified 280 passing students as failing, accumulating 387 failing alerts that needed only 126. In this hypothetical scenario, tutors, teachers, and coordinators are being unnecessarily mobilized by alerts and pedagogical guidelines to contact students who should not be reached at this time. Results like this could generate an overload of work for the teachers and a lack of motivation for students, which clearly can harm the teaching-learning process in several ways. It is also worth mentioning that the Logistic Regression model, without balancing options, obtained the best precision (94.36%) while presenting the worst result for recall (53.17%).

The **specificity** metric assesses the accuracy of the prediction in the majority class (passing). The following formula gives its calculation: \( \frac{tn}{tn + fp} \), indicating how many of those who passed were classified as passing. Figure 5(b) shows the results ordered by specificity. At first, an incipient analysis could conclude that low specificities would not have much relevance in predicting failure. If a model makes a mistake classifying a successful student as a failing one, the consequence of this error for the student could be to study and try harder, which seems to be quite acceptable. However, as the previous analysis demonstrated with the precision metric, classifying many students who passed as failing can cause different types of loss, from financial through motivational, to logistical, among others. As the specificity metric analyzes the majority class (passing), any error is proportionally impactful in the general context. Regarding specificity, the Logistic Regression model achieved the highest score (99.566%).

4.3.3 Analysis of the other metrics

In addition to the metrics analyzed above (accuracy, precision, recall, and specificity), **g-means**, **f-measure**, and **kappa index** metrics add more information that helps to assess the classifiers’ performance. They have the power to combine the previous ones in different ways and can be analyzed in an isolated or combined way, preferably the latter. When taking these three metrics into account allows attributing greater or lesser relevance to the results obtained by each classifier.

The g-means metric corresponds to the square root of the product between specificity and recall \( \sqrt{spec \times reca} \). This metric varies from 0 to 1 and considers the model’s correctness rates in the majority and minority classes. The f-measure metric corresponds to the value of twice the precision times recall divided by the sum of the precision with the recall \( \frac{2 \times prec \times reca}{prec + reca} \). F-measure varies from 0 to 1 and considers the precision and recall, combining two crucial metrics related to the minority class (failing). Both g-means and f-measure are metrics whose values vary between 0 and 1, with the highest values being the most relevant. The Cohen’s kappa index corresponds to a statistical coefficient that compares the accuracy expected by a random classification with the general accuracy of the evaluated model. The following formula gives its calculation: \( kappa = \frac{ana - prec}{ana - prec} \), where

\[
\begin{align*}
pra &= \frac{tp+in}{tp+in+fp+fn} \\
pre &= \frac{tp+in+fp+fn}{tp+in+fp+fn} \\
\end{align*}
\]

and

\[
\begin{align*}
ana &= \frac{tp+in+fp+fn}{tp+in+fp+fn} \\
rec = \frac{tp+in+fp+fn}{tp+in+fp+fn} \\
\text{Values below 0.3 suggest that the model has a low capacity to make bold predictions, which indicates a possibly lower quality of its results in comparison to a random prediction. All the analyses presented here proved essential for understanding the need to interpret the metrics for predicting students’}
\]

### Table: Precision and Specificity of classifiers

| Classifier                  | prec  | spec  |
|-----------------------------|-------|-------|
| Logistic Regression         | 0.9436 | 0.9956 |
| SVM with RBF kernel         | 0.89412 | 0.99024 |
| SVM with a 2nd-deg kernel   | 0.82353 | 0.98048 |
| Random Forest               | 0.80000 | 0.97831 |
| SVM with a 3rd-deg kernel   | 0.77778 | 0.97164 |
| SMOTE: Logistic Regression  | 0.74359 | 0.96746 |
| SMOTE: SVM with a 3rd-deg kernel | 0.69630 | 0.95553 |
| SMOTE: SVM with RBF kernel  | 0.68456 | 0.94902 |
| SMOTE: SVM with a 2nd-deg kernel | 0.66429 | 0.94902 |
| Decision Tree               | 0.62992 | 0.9402 |
| SVM with a 4th-deg kernel   | 0.61481 | 0.94360 |
| SMOTE: SVM with a 4th-deg kernel | 0.52047 | 0.91106 |
| NearMiss: Logistic Regression | 0.46023 | 0.89669 |
| NearMiss: SVM with a 2nd-deg kernel | 0.30218 | 0.75705 |
| NearMiss: SVM with a 3rd-deg kernel | 0.28045 | 0.72451 |
| NearMiss: SVM with RBF kernel | 0.27649 | 0.71367 |
| NearMiss: SVM with a 4th-deg kernel | 0.27273 | 0.69631 |
| SMOTE: Random Forest        | 0.24755 | 0.66703 |
| NearMiss: Decision Tree     | 0.20807 | 0.59544 |
| SMOTE: Decision Tree        | 0.19658 | 0.59219 |
| NearMiss: Random Forest     | 0.19521 | 0.52603 |

Fig. 5 Precision and Specificity of classifiers
failure. The complexity inherent to teaching-learning process goes beyond the objectivity proposed by the performance metrics and must also be taken into consideration.

5 Proposal

Assessing predictive models for students’ failure requires in-depth studies to consider the specific institutional and methodological context in which those students are inserted. The proposed framework consists of (i) applying exclusion criteria on the evaluation metrics obtained to produce a smaller set of viable models; and (ii) applying an ordering criterion to the resulting models.

At first, we must exclude models whose accuracy, precision, recall, specificity, g-means, and f-measure are less than 50%, and models whose kappa indexes are less than 0.3. These criteria eliminate models whose applications in real contexts are not feasible, given the low ability to correctly classify students. The application of the kappa index eliminates models whose metrics are of low quality, considering a random classification. The application of the exclusion criteria removed nine prediction models, leaving 12 for the subsequent stage. Next, we must sort these remaining models in descending recall order to generate a list with the best results of the models filtered in the previous step. As a result, models with the highest recalls will have the lowest false negatives, prioritizing models that less frequently classify students who fail as having passed.

Figure 6 shows the result obtained after applying the proposed framework. The rows in the table highlighted in yellow represent the top three student-failure prediction models. As we can see, the SVM classifier with RBF kernel using the SMOTE method is the best option among the 21 analyzed models. This method minimized the number of failing students incorrectly classified as passing (only 24). Furthermore, it obtained the highest values of g-means and f-measure among all tested models. The metrics presented in Fig. 6 provide a more detailed inspection of the performance of the filtered and ordered models. For example, we can observe that all models based on the undersampling balancing method were excluded after applying the proposed exclusion criterion, suggesting that this balancing strategy did not facilitate the failure prediction in the analyzed scenario. Moreover, it is possible to notice that the 12 remaining models have 8 of the 11 best metrics evaluated (highlighted in green). These observations corroborate the quality of the proposed framework.

6 Closing remarks and future work

The contribution of this work is the proposition, application, and analysis of a framework for evaluating the prediction of school failure in distance learning, which can be used as a strategy to facilitate early pedagogical interventions. Predicting students’ failure in advance can be an effective strategy to reverse this negative outcome and favor the teaching-learning process in different ways. This work implemented and tested seven different predictive models, with and without balancing options, and collected 11 evaluation metrics. The results show the feasibility of achieving good performances in predicting the student failure with 50% of the total course time elapsed (accuracy: 94.37%; precision: 94.37%; recall: 94.37%; specificity: 99.57%; g-means: 87.65%; f-measure: 74.18%; and kappa index: 0.705).

The challenge to be overcome in this work was to choose the best predictive model to be adopted in a student failure risk warning system. Therefore, we proposed a framework to incorporate both the subjectivity inherent to the educational context and the objectivity of the classification metrics. The results of this framework proved to be suitable to be adopted in decision-making processes that aim to mitigate and eventually eliminate the risk of students failing. Promoting data-driven education that takes advantage of the growing availability of digital data and the spread of machine learning...
techniques is an increasingly clear educational agenda. Many educational contexts lack new specific theoretical and practical contributions to face their shortcomings and strengthen their virtues. Therefore, different knowledge areas need to interact to propose updated technological solutions and produce practical actions that are, above all, ethical.

It is worth mentioning that this work possesses some limitations. First, since the analyzed case study represents a specific scenario of a Brazilian university, the results obtained cannot be generalized. Furthermore, although we have compared 11 different evaluation metrics, one could evaluate other metrics, such as the ROC-area metric, or even combine weights from other metrics, as pointed out in [9] and [15]. Finally, all algorithms we implemented used the standard parameterization, so potential adjustments in these parameters could significantly impact the results.

As future studies, we intend to collect and evaluate the opinion of educational experts (notably distance-learning coordinators, teachers, and tutors) about the proposed framework. Based on this data collected, we intend to propose a new evaluation metric that assigns penalties to the false positives and false negatives of tested models, to improve the identification of the best models. We also intend to evaluate the predictive results by using demographic data, together with data collected from Moodle and specific questionnaires. Finally, we hope to implement strategies to incorporate automatic failure prediction alerts to the LMS used by university, considering the selected prediction model. The authors declare that they do not have any conflict of interest.

References

1. Adejo OW, Connolly T (2018) Predicting student academic performance using multi-model heterogeneous ensemble approach. J Appl Res High Educ 10(1):61–75. https://doi.org/10.1108/JARHE-09-2017-0113
2. Aguiar E, Ambrose GAA, Chawla NV, Goodrich V, Brockman J (2014) Engagement vs performance: Using electronic portfolios to predict first semester engineering student persistence. J Learn Anal 1 (3):7–33. https://doi.org/10.18608/jla.2014.13.3
3. Angeli C, Howard SK, Ma J, Yang J, Kirschner PA (2017) Data mining in educational technology classroom research. Comput Educ 113:226–242. https://doi.org/10.1016/j.compedu.2017.05.021
4. Araque F, Roldán C., Salguero A (2009) Factors influencing university drop out rates. Comput Educ 53(3):563–574. https://doi.org/10.1016/j.compedu.2009.03.013
5. Azcona D, Hsiao IH, Smeaton AF (2019) Detecting students-at-risk in computer programming classes with learning analytics from students’ digital footprints. User Model User-Adap Inter 29(4):759–788. https://doi.org/10.1007/s11257-019-09234-7
6. Baker R, Isotani S, Carvalho A (2011) Mineração de dados educacionais: Oportunidades para o Brasil. Revista Brasileira de Informá,tica na Educação 19(02):03–13. https://doi.org/10.5753/ribe.2011.19.02.03
7. Casalino G, Castellano G, Vessio G (2021) Exploiting time in adaptive learning from educational data. In: Bridges and Mediation in Higher Distance Education. https://doi.org/10.1007/978-3-030-67435-9_1. Springer International Publishing, Cham, pp 3–16
8. Costa EB, Fonseca B, Santana MA, De araújo F.E., Rego J (2017) Evaluating the effectiveness of educational data mining techniques for early prediction of students’ academic failure in introductory programming courses. Comput Human Behav 73:247–256. https://doi.org/10.1016/j.chb.2017.01.047
9. Dekker GW, Pechenizkiy M, Vleeshouwers JM (2009) Predicting students drop out: A case study. In: Proceedings of the 2nd international conference on educational data mining. Cordoba, Spain. https://eric.ed.gov/?id=ED539082
10. Detoni D, Cechinel C, Araújo R. M. (2015) Modelagem e predição de reprovação de acadêmicos de cursos de educação a distância a partir da contagem de interações. Revista Brasileira de Informática na Educação 23(3):1–11
11. Fayyad U, Piattetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery in databases. AI Mag 17(13):37–54. https://doi.org/10.1609/aimag.v17i3.1230
12. GKontzis AF, Kotsiantis S, Panagiotakopoulos CT, Verykios VS (2019) A predictive analytics framework as a countermeasure for attrition of students. Interact Learn Environ, 1–15. https://doi.org/10.1080/10494820.2019.1674884
13. GKontzis AF, Kotsiantis S, Tsoni R, Verykios VS (2018) An effective LA approach to predict student achievement. In: Proceedings of the 22nd Pan-Hellenic Conference on Informatics, PCl ’18, p. 76–81. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3291533.3291551
14. Luckin R, Holmes W, Griffiths M, Forcier LB (2016) Intelligence Unleashed. An argument for AI in Education. Pearson, London. http://discovery.ucl.ac.uk/1475756/
15. Marquez-Vera C., Cano A, Romero C, Noaman AYM, Fardoun HM, Ventura S (2016) Early dropout prediction using data mining: a case study with high school students. Expert Syst 33(1):107–124. https://doi.org/10.1111/exsy.12135
16. Romero C, Ventura S (2013) Data mining in education. WIREs Data Mining and Knowledge Discovery 3(1):12–27. https://doi.org/10.1002/widm.1075
17. Romero C, Ventura S (2020) Educational data mining and learning analytics: an updated survey. WIREs Data Mining and Knowledge Discovery 10(3):e1355. https://doi.org/10.1002/widm.1355
18. Silva Filho RLL, Motejunas PR, Hipólito O., Lobo MBDCM (2007) A evasão no ensino superior brasileiro. Cad Pesqui 37(132):641–659. https://doi.org/10.1590/S0100-1574200700300007
19. Witten IH, Frank E, Halle MA (2011) Data Mining. 3edn. Morgan Kaufman, San Francisco, CA

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.